

PHP4

Deutschsprachige Befehls- und Funktionsreferenz

Autor: Jörg Krause

www.joerg.krause.net

Stand: 28.09.2000

Basiert auf Version: PHP 4.0.1

Veröffentlicht unter GNU GPL Version 2

Einige Teile sind © PHP Documentation Group

This manual can be redistributed under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

A Referenz

A.1 Informationen zu PHP4 und dieser Referenz

Diese Referenz wird auch in gedruckter Form zur Verfügung gestellt. Sie ist Bestandteil des Buches »PHP4 – Grundlagen und Profiwissen«, das Oktober 2000 bei Carl Hanser Verlag, München, erschienen ist. Dieses Buch ist die umfassendste und beste am deutschsprachigen Markt verfügbare Literatur zum Thema »PHP«. Behandelt werden alle Aspekte der Programmierung, von den Grundlagen der Webserverprogrammierung für Anfänger bis hin zu komplexen Themen wie XML, WDDX und WML für Profis. Mit einem Umfang von fast 1 200 Seiten ist es schon in der letzten Auflage zu PHP3 das Standardwerk zu PHP gewesen.

Wenn Sie diese Referenz bequem auf Papier nutzen und zugleich unzählige wertvolle Informationen haben möchten, sollten Sie das Buch unbedingt erwerben. Es ersetzt praktisch jede weitere Sekundärliteratur. Sie finden das Buch im gut sortierten Buchhandel und in allen Online-Buchhandlungen unter der ISBN-Nummer 3-446-21546-8. Fragen Sie auch nach den anderen Büchern des Autors zu den Themen Windows 2000, MS SQL Server 7, Electronic Commerce und Online Marketing, FrontPage 2000, Microsoft Active Server Pages u.v.m. Hinweise zum Autor finden Sie auf der letzten Seite. Vielen Dank!

A.2 Copyright und Lizenzhinweise

Entwicklung

Diese Referenz entstand anfangs durch Übersetzung und Erweiterung der mit PHP ausgelieferten Dokumentation der PHP Documentation Group. Als Basis diente die englische Fassung der Referenz der PHP Documentation Group vom Oktober 1999 für die Version PHP 3.0.12. Sie wurde bis September 2000 auf den häufiger verwendeten Funktionskern gekürzt und anschließend um die Funktionen der Version PHP 4 erweitert und vollständig ins deutsche übersetzt. Diese Übersetzung ist unabhängig von der von der PHP Documentation Group herausgegebenen Übersetzung und ohne Rückgriff auf diese entstanden. Die hier vorliegende Fassung wurde und wird laufend erweitert und ergänzt.

Die offizielle Dokumentation wird unter der GPL – General Public License veröffentlicht. Die hier abgedruckte Fassung wird ebenso unter den Bestimmungen der GPL Version 2 veröffentlicht. Die entsprechenden Lizenzbestimmungen der PHP Documentation Group werden uneingeschränkt anerkannt. Mehr Informationen finden Sie unter <http://www.php.net/manual>.

Entsprechend den Bestimmungen dieser Lizenz gilt für die Referenz folgendes:

- Sie dürfen auf beliebigen Medien unveränderte Kopien dieser Referenz anfertigen und verbreiten.
- Sie müssen diese Kopien mit der (in englischer Fassung) auf CD beiliegenden GPL verbreiten. Beachten Sie die Hinweise zur deutschen Fassung im nächsten Abschnitt.
- Sie können jederzeit eine aktuelle, kostenlose Kopie der Referenz in elektronischer Form über die Website www.phpwelt.de oder www.php.comzept.de/php4 erhalten.

Allgemeine Hinweise

Copyright © 2000, Jörg Krause

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Es folgt die deutsche Version (rechtlich verbindlich ist die englische Fassung):

Copyright © 2000, Jörg Krause

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG – sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK.

Details finden Sie in der GNU General Public License. Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Urheberrechtshinweis

Carl Hanser Verlag, München, hereby disclaims all copyright interest in the program »PHP Referenz« (function reference for the php scripting language) written by Joerg Krause.

1. September 2000

Carl Hanser Verlag, München

Auf Deutsch:

Der Carl Hanser Verlag, München, erhebt keinerlei urheberrechtlichen Anspruch auf das Programm »PHP Referenz« (Funktionsreferenz für die Skriptsprache PHP), geschrieben von Jörg Krause.

1. September 2000

Carl Hanser Verlag, München

Deutsche Fassung der GPL (Auszug)

Übersetzt von Katja Lachmann Übersetzungen (na194@fim.uni-erlangen.de), modifiziert von Peter Gerwinski (peter.gerwinski@uni-essen.de) und Jörg Krause (joerg@krause.net). Es handelt sich nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Die Free Software Foundation (FSF) ist nicht der Herausgeber dieser Übersetzung, und sie hat diese Übersetzung auch nicht als rechtskräftigen Ersatz für die Original-GNU-GPL anerkannt. Da die Übersetzung nicht sorgfältig von Anwälten überprüft wurde, können die Übersetzer nicht garantieren, dass die Übersetzung die rechtlichen Aussagen der GNU-GPL exakt wiedergibt. Wenn Sie sichergehen wollen, daß von Ihnen geplante Aktivitäten im Sinne der GNU-GPL gestattet sind, halten Sie sich bitte an die englischsprachige Originalversion.

Die Free Software Foundation möchte Sie darum bitten, diese Übersetzung nicht als offizielle Lizenzbedingungen für von Ihnen geschriebene Programme zu verwenden. Bitte benutzen Sie hierfür stattdessen die von der Free Software Foundation herausgegebene englischsprachige Originalversion.

Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung

§0 Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, dass das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als »das Programm« bezeichnet; die Formulierung »auf dem Programm basierendes Werk« bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch

auszugsweise, sei es unverändert oder verändert oder in eine andere Sprache übersetzt, enthält.

Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in ihren Anwendungsbe- reich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, dass die Ausgabe durch die Ausführung des Programmes erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

- §1** Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, dass Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluss veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und des Weiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen. Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.
- §2** Sie dürfen Ihre Kopien des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, dass zusätzlich alle folgenden Bedingungen erfüllt werden:
- (a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
 - (b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.
 - (c) Wenn das veränderte Programm normalerweise bei der Ausführung interaktiv Kommandos einliest, müssen Sie dafür sorgen, dass es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdrückt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, dass es keine Gewährleistung gibt (oder anderenfalls, dass Sie Garantie leisten), und dass die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muss der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muss Ihr

auf dem Programm basierendes Werk auch keine solche Meldung ausgeben.)

Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Abschnitte des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich diese Lizenz und ihre Bedingungen nicht auf diese Abschnitte, wenn sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muss die Verbreitung des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf die Gesamtheit ausgedehnt werden – und damit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor.

Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder zu beschneiden, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben.

Ferner bringt ein einfaches Zusammenstellen eines anderen Werkes, das nicht auf dem Programm basiert, zusammen mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Vertriebsmedium das andere Werk nicht in den Anwendungsbereich dieser Lizenz.

- §3 Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Paragraph 1 und 2 vervielfältigen und verbreiten – vorausgesetzt, dass Sie außerdem eine der folgenden Leistungen erbringen:
- (a) Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muss. Oder:
 - (b) Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen – zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen –, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weitergegeben wird. Oder:
 - (c) Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben.

(Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.)

Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet »der komplette Quelltext«: Der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien sowie der zur Compilation und Installation verwendeten Skripte. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft – es sei denn, diese Komponente selbst gehört zum ausführbaren Programm.

Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes dadurch erfolgt, dass der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

- §4 Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.
- §5 Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unterzeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.
- §6 Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

- §7 Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Bedingungen (durch Gerichtsbeschluss, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten.

Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sinne nach angewandt werden; im Übrigen soll dieser Paragraph als Ganzes gelten.

Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Verbreitungssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluss.

Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

- §8 Wenn die Verbreitung oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, sodass die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.
- §9 Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, kön-

nen aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden.

Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, dass es dieser Lizenz in einer bestimmten Versionsnummer oder »jeder späteren Version« unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

- §10** Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet werden, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im Allgemeinen zu fördern.
- §11** Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, »wie es ist«, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich – aber nicht begrenzt auf – Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.
- §12** In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich – aber nicht beschränkt auf – Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

Hinweise des Autors zur Referenz

Die hier abgedruckte und online verfügbare Referenz weicht in weiten Teilen von der offiziellen Dokumentation zu PHP ab. Diese Referenz ist ein eigenständiges Projekt des Autors, das unabhängig von der offiziellen Dokumentation ist und andere Ziele verfolgt, als die PHP Documentation Group. Es sei dem Leser überlassen, sich für das eine oder andere Projekt zu entscheiden. Für Kritik und Anregungen schreiben Sie an:

- *joerg@krause.net*

A.3 Inhaltsverzeichnis der Referenz

A	Referenz.....	3
A.1	Informationen zu PHP4 und dieser Referenz	3
A.2	Copyright und Lizenzhinweise.....	3
Entwicklung.....		3
Allgemeine Hinweise		4
Deutsche Fassung der GPL (Auszug)		5
Hinweise des Autors zur Referenz		11
A.3	Inhaltsverzeichnis der Referenz	12
A.4	Aufbau und Schreibstil	52
A.5	Alphabetische Übersicht	53
Schreibweisen		53
Was nicht aufgeführt wurde.....		54
A.6	Befehlsreferenz	80
Kontrollstrukturen		80
if...else...elseif.....		80
while		81
do ... while.....		81
for		81
foreach.....		82
break		82
continue.....		83
switch		83
require		84
include.....		84
function		84
return.....		84
class.....		84

new.....	85
extends.....	86
Umgang mit Klassen.....	86
call_user_method.....	86
get_class.....	86
get_parent_class.....	86
get_class_methods.....	86
get_class_vars.....	87
get_object_vars.....	87
is_subclass_of.....	87
class_exists.....	87
method_exists.....	87
get_declared_classes.....	87
com.....	88
com_load.....	88
com_invoke.....	88
com_propget.....	88
com_propput.....	89
Variable, Konstanten und Parameter.....	89
define.....	89
defined.....	89
func_get_arg.....	89
func_get_args.....	89
func_num_args.....	89
global.....	90
static.....	90
\$.....	90

Typkonvertierungen	90
(double)	90
(integer)	91
(string)	91
(array)	91
(object)	91
A.7 Funktionsreferenz.....	92
URL und HTML	92
parse_url	92
urldecode	92
urlencode	92
base64_encode.....	92
base64_decode.....	92
header	93
headers_sent.....	93
setcookie.....	93
Variablenfunktionen.....	94
call_user_func.....	94
gettype.....	94
intval.....	95
doubleval	95
empty.....	95
is_array.....	95
is_double.....	96
is_float	96
is_int	96
is_integer.....	96
is_long	96

is_object.....	97
is_real.....	97
is_string.....	97
isset.....	97
settype.....	97
strval.....	98
unset.....	98
Array-Funktionen.....	98
array.....	98
array_walk.....	99
arsort.....	99
asort.....	100
count.....	100
current.....	100
each.....	101
end.....	102
key.....	102
ksort.....	102
krsort.....	103
list.....	103
next.....	103
pos.....	104
prev.....	104
range.....	104
reset.....	104
rsort.....	104
shuffle.....	105
sizeof.....	105

sort	105
uasort.....	106
uksort.....	106
usort.....	106
Neue Array-Funktionen in PHP4	107
array_count_values	107
array_diff	107
array_flip.....	108
array_keys.....	108
array_merge.....	108
array_merge_recursive	109
array_multisort	109
array_pad.....	109
array_pop.....	110
array_push.....	110
array_reverse.....	110
array_shift.....	111
array_unique	111
array_unshift	111
array_slice	112
array_splice.....	112
array_values	113
compact	113
extract	114
in_array	115
Zeichenkettenfunktionen	115
addslashes.....	115
addcslashes	115

bin2hex	116
chop.....	116
chr.....	116
chunk_split.....	116
convert_cyr_string	117
count_chars	117
crc32	118
crypt.....	118
echo	118
explode	119
get_html_translation_table	119
get_meta_tags.....	119
htmlspecialchars.....	120
htmlentities	120
implode.....	120
join.....	121
ltrim.....	121
md5	121
metaphone.....	121
nl2br	121
ord	122
parse_str	122
print.....	122
printf	122
quoted_printable_decode	123
quotemeta.....	123
rawurldecode.....	123
rawurlencode.....	123

setlocale.....	124
similar_text.....	126
soundex.....	126
sprintf.....	127
strchr.....	128
strcmp.....	128
strcspn.....	128
strip_tags.....	128
stripslashes.....	129
strlen.....	129
strrpos.....	129
strpos.....	129
strrchr.....	129
strev.....	130
strspn.....	130
strstr.....	130
strtok.....	130
strtolower.....	131
strtoupper.....	131
str_replace.....	131
strtr.....	131
substr.....	132
trim.....	132
ucfirst.....	132
ucwords.....	132
Funktionen für Rechtschreibprüfung.....	133
pspell_new.....	133
pspell_check.....	134

pspell_suggest	134
GNU Gettext Sprachunterstützung	134
bindtextdomain	134
dcgettext	135
dgettext	135
gettext	135
textdomain	135
Funktionen für reguläre Ausdrücke	135
ereg	136
ereg_replace	136
eregi	137
eregi_replace	137
split	137
sql_regcase	137
preg_match	138
preg_match_all	138
preg_replace	139
preg_split	140
preg_quote	140
preg_grep	140
Die Musterschalter der regulären Ausdrücke	141
Datums-, Kalender- und Zeitfunktionen	142
checkdate	142
date	142
strftime	144
gmstrftime	145
getdate	145
gettimeofday	146

gmdate.....	146
localtime.....	146
strptime.....	147
mktime.....	147
gmmktime.....	148
time.....	148
microtime.....	148
Kalenderfunktionen.....	148
jdtogregorian.....	148
gregoriantojd.....	149
jdtojulian.....	149
juliantojd.....	149
jdtojewish.....	149
jewishtojd.....	149
jdtofrench.....	150
frenchtojd.....	150
jdmonthname.....	150
jddayofweek.....	150
easter_date.....	151
easter_days.....	151
Mathematische Funktionen.....	151
Konstanten.....	151
abs.....	152
acos.....	152
asin.....	152
atan.....	152
atan2.....	153
base_convert.....	153

bindec.....	153
ceil	153
cos.....	153
decbin.....	153
dechex.....	154
decoct.....	154
deg2rad.....	154
exp.....	154
floor.....	154
getrandmax.....	154
hexdec.....	154
log.....	155
log10.....	155
max.....	155
min	155
mt_rand.....	155
mt_srand	156
mt_getrandmax.....	156
number_format.....	156
octdec.....	157
pi.....	157
pow	157
rad2deg.....	157
rand.....	157
round	157
sin.....	158
sqrt	158
srand	158

tan	158
Funktionen beliebiger Genauigkeit	158
bcadd	158
bccomp	159
bcdiv	159
bcmod	159
bcmul	159
bcpow	160
bcscale.....	160
bcsqrt	160
bcsub.....	160
Funktionen für das Dateisystem	160
dir	161
chdir.....	161
closedir	162
getcwd	162
opendir	162
readdir.....	162
rewinddir	162
basename.....	163
chgrp.....	163
chmod.....	163
chown	163
clearstatcache.....	164
copy	164
dirname	164
diskfreespace	164
fclose.....	165

feof.....	165
fgetc.....	165
fgetcsv.....	165
fgets.....	166
fgetss.....	166
file.....	166
file_exists.....	166
fileatime.....	166
filectime.....	167
filegroup.....	167
fileinode.....	167
filemtime.....	167
fileowner.....	168
fileperms.....	168
filesize.....	168
filetype.....	168
flock.....	168
fopen.....	169
fpassthru.....	170
fputs.....	170
fread.....	170
fseek.....	171
ftell.....	171
ftruncate.....	171
fwrite.....	171
set_file_buffer.....	172
is_dir.....	172
is_executable.....	172

is_file.....	172
is_link	173
is_readable	173
is_writeable.....	173
link	174
linkinfo	174
mkdir	174
pclose.....	174
popen.....	174
readfile.....	175
readlink	175
rename.....	175
rewind	175
rmdir.....	175
stat.....	176
lstat.....	176
symlink.....	177
tempnam	177
touch	177
umask	177
unlink	178
Funktionen zur Sitzungssteuerung	178
Konfiguration für Sessions	178
session_decode.....	179
session_destroy	179
session_encode.....	179
session_id.....	180
session_is_registered.....	180

session_module_name	180
session_name	180
session_register	180
session_save_path	181
session_start	181
session_unregister	181
session_set_save_handler	181
Funktionen für den Netzwerkzugriff	182
fsockopen	182
ip2long	183
long2ip	183
pfsockopen	183
set_socket_blocking	184
gethostbyaddr	184
gethostbyname	184
gethostbyname1	184
checkdnsrr	184
getmxrr	185
FTP-Funktionen	185
ftp_connect	185
ftp_login	185
ftp_pwd	185
ftp_chdir	185
ftp_cdup	186
ftp_mkdir	186
ftp_rmdir	186
ftp_nlist	186
ftp_rawlist	186

ftp_systype.....	186
ftp_pasv.....	187
ftp_get.....	187
ftp_fget	187
ftp_put.....	187
ftp_fput.....	188
ftp_size	188
ftp_mdtm	188
ftp_rename.....	188
ftp_delete	189
ftp_quit.....	189
Verschlüsselungs-Funktionen	189
mcrypt_get_cipher_name	190
mcrypt_get_block_size.....	190
mcrypt_get_key_size.....	190
mcrypt_create_iv	191
mcrypt_cbc.....	191
mcrypt_cfb.....	191
mcrypt_ecb	192
mcrypt_ofb.....	192
Hashfunktionen.....	192
mhash_get_hash_name.....	193
mhash_get_block_size.....	193
mhash_count	193
mhash	194
WDDX-Funktionen	194
wddx_serialize_value.....	195
wddx_serialize_vars.....	195

wddx_packet_start.....	196
wddx_packet_end.....	196
wddx_add_vars.....	196
wddx_deserialize.....	196
XML-Funktionen.....	196
Ereignisbehandlungsroutinen.....	196
XML-Fehlercodes.....	197
xml_parser_create.....	198
xml_set_element_handler.....	198
xml_set_character_data_handler.....	199
xml_set_processing_instruction_handler.....	200
xml_set_default_handler.....	200
xml_set_unparsed_entity_decl_handler.....	201
xml_set_notation_decl_handler.....	202
xml_set_external_entity_ref_handler.....	202
xml_parse.....	203
xml_get_error_code.....	204
xml_error_string.....	204
xml_get_current_line_number.....	204
xml_get_current_column_number.....	204
xml_get_current_byte_index.....	204
xml_parser_free.....	204
xml_parser_set_option.....	205
xml_parser_get_option.....	205
utf8_decode.....	205
utf8_encode.....	205
PDF-Funktionen.....	206
pdf_get_info.....	206

pdf_set_info_creator.....	206
pdf_set_info_title	206
pdf_set_info_subject.....	207
pdf_set_info_keywords.....	207
pdf_set_info_author	207
pdf_open.....	207
pdf_close	208
pdf_begin_page.....	208
pdf_end_page.....	208
pdf_show	208
pdf_show_xy	208
pdf_set_font.....	209
pdf_set_leading.....	209
pdf_set_text_rendering.....	209
pdf_set_horiz_scaling	210
pdf_set_text_rise	210
pdf_set_text_matrix.....	210
pdf_set_text_pos	210
pdf_set_char_spacing.....	210
pdf_set_word_spacing.....	211
pdf_continue_text.....	211
pdf_stringwidth.....	211
pdf_save	211
pdf_restore.....	211
pdf_translate.....	211
pdf_scale	212
pdf_rotate.....	212
pdf_setflat	212

pdf_setlinejoin	212
pdf_setlinecap.....	213
pdf_setmiterlimit.....	213
pdf_setlinewidth	213
pdf_setdash.....	213
pdf_moveto.....	213
pdf_curveto.....	213
pdf_lineto	213
pdf_circle.....	214
pdf_arc.....	214
pdf_rect.....	214
pdf_closepath	214
pdf_stroke	214
pdf_closepath_stroke.....	215
pdf_fill	215
pdf_fill_stroke	215
pdf_closepath_fill_stroke.....	215
pdf_endpath.....	215
pdf_clip.....	216
pdf_setgray_fill	216
pdf_setgray_stroke	216
pdf_setgray	216
pdf_setrgbcolor_fill.....	216
pdf_setrgbcolor_stroke.....	216
pdf_setrgbcolor	217
pdf_add_outline.....	217
pdf_set_transition	217
pdf_set_duration.....	218

pdf_open_gif	218
pdf_open_memory_image	218
pdf_open_jpeg.....	218
pdf_close_image	218
pdf_place_image.....	218
pdf_put_image.....	219
pdf_execute_image.....	219
Shockwave-Flash-Funktionen	219
swf_openfile	220
swf_closefile	220
swf_labelframe.....	220
swf_showframe.....	220
swf_setframe	220
swf_getframe.....	221
swf_mulcolor.....	221
swf_addcolor.....	221
swf_placeobject	221
swf_modifyobject.....	221
swf_removeobject.....	221
swf_nextid.....	222
swf_startdoaction.....	222
swf_enddoaction.....	222
swf_actiongotoframe.....	222
swf_actiongeturl	222
swf_actionnextframe	222
swf_actionprevframe.....	222
swf_actionplay	223
swf_actionstop	223

swf_actiontogglequality	223
swf_actionwaitforframe	223
swf_actionsettarget	223
swf_actiongotolabel	223
swf_defineline	223
swf_definerect	224
swf_startshape	224
swf_definepoly	224
swf_shapelinesolid	224
swf_shapefilloff	224
swf_shapefillsolid	224
swf_shapefillbitmapclip	225
swf_shapefillbitmaptile	225
swf_shapemoveto	225
swf_shapelineto	225
swf_shapecurveto	225
swf_shapecurveto3	225
swf_shapearc	226
swf_endshape	226
swf_definefont	226
swf_setfont	226
swf_fontsize	226
swf_fontslant	226
swf_fontracking	226
swf_getfontinfo	227
swf_definetext	227
swf_textwidth	227
swf_definebitmap	227

swf_getbitmapinfo.....	227
swf_startsymbol.....	228
swf_endsymbol.....	228
swf_startbutton.....	228
swf_addbuttonrecord.....	228
swf_oncondition.....	229
swf_endbutton.....	230
swf_viewport.....	230
swf_ortho.....	230
swf_ortho2.....	230
swf_perspective.....	230
swf_polarview.....	231
swf_lookat.....	231
swf_pushmatrix.....	231
swf_popmatrix.....	231
swf_scale.....	231
swf_translate.....	231
swf_rotate.....	232
swf_posround.....	232
Mail-Funktionen.....	232
mail.....	232
imap_append.....	233
imap_base64.....	233
imap_body.....	233
imap_check.....	233
imap_close.....	234
imap_createmailbox.....	234
imap_delete.....	234

imap_deletemailbox	234
imap_expunge	234
imap_fetchbody	234
imap_fetchstructure	235
imap_header	235
imap_headers	237
imap_listmailbox	238
imap_listsubscribed	238
imap_mail_copy	238
imap_mail_move	238
imap_num_msg	238
imap_num_recent	239
imap_open	239
imap_ping	240
imap_renamemailbox	240
imap_reopen	240
imap_subscribe	240
imap_undelete	240
imap_unsubscribe	241
imap_qprint	241
imap_8bit	241
imap_binary	241
imap_scanmailbox	241
imap_mailboxmsginfo	241
imap_rfc822_write_address	242
imap_rfc822_parse_adrlist	242
imap_setflag_full	242
imap_clearflag_full	243

imap_sort	243
imap_fetchheader	243
imap_uid	244
LDAP-Funktionen	244
ldap_add	244
ldap_mod_add	245
ldap_mod_del	245
ldap_mod_replace	245
ldap_bind	245
ldap_close	245
ldap_connect	246
ldap_count_entries	246
ldap_delete	246
ldap_dn2ufn	246
ldap_explode_dn	246
ldap_first_attribute	247
ldap_first_entry	247
ldap_free_result	247
ldap_get_attributes	247
ldap_get_dn	248
ldap_get_entries	248
ldap_get_values	249
ldap_list	249
ldap_modify	250
ldap_next_attribute	250
ldap_next_entry	250
ldap_read	250
ldap_search	251

ldap_unbind	251
Bildfunktionen (GD)	252
getimagesize	252
imagearc	253
imagechar	253
imagecharup	253
imagecolorallocate	253
imagecolortransparent	254
imagecopyresized	254
imagecreate	254
imagecreatefromgif	255
imagecreatefromjpeg	255
imagecreatefrompng	255
imagedashedline	255
imagedestroy	256
imagefill	256
imagefilledpolygon	256
imagefilledrectangle	256
imagefilltoborder	256
imagefontheight	257
imagefontwidth	257
imagegif	257
imagejpeg	257
imagepng	257
imageinterlace	258
imageline	258
imageloadfont	258
imagepolygon	259

imagerectangle	259
imagesetpixel.....	259
imagestring.....	259
imagestringup	259
imagesx	260
imagesy	260
imagettfbbox.....	260
imagettftext.....	261
imagecolorat	262
imagecolorclosest.....	262
imagecolorexact	262
imagecolorresolve.....	262
imagecolorset.....	262
imagecolorsforindex.....	263
imagecolorstotal.....	263
imagepsloadfont	263
imagepsfreefont	263
imagepsencodefont.....	264
imagepstext.....	264
imagepsbbox.....	265
iptcparse.....	265
Hyperwave-Funktionen (hw).....	266
hw_array2objrec.....	266
hw_children.....	266
hw_childrenobj	266
hw_close.....	266
hw_connect.....	267
hw_cp	267

hw_deleteobject.....	267
hw_docbyanchor.....	267
hw_docbyanchorobj	267
hw_documentattributes	267
hw_documentbodytag	268
hw_documentcontent.....	268
hw_documentsetcontent.....	268
hw_documentsize	268
hw_errormsg	268
hw_edittest	268
hw_error.....	269
hw_free_document.....	269
hw_getparents	269
hw_getparentsobj.....	269
hw_getchildcoll	269
hw_getchildcollobj.....	269
hw_getremote	270
hw_getremotechildren	270
hw_getsrcbydestobj	270
hw_getobject.....	270
hw_getadlock.....	270
hw_gettext.....	270
hw_getobjectbyquery	271
hw_getobjectbyqueryobj.....	271
hw_getobjectbyquerycoll	271
hw_getobjectbyquerycollobj.....	272
hw_getchilddoccoll.....	272
hw_getchilddoccollobj.....	272

hw_getanchors	272
hw_getanchorsobj.....	272
hw_mv.....	272
hw_identify.....	273
hw_incollections	273
hw_info	273
hw_inscoll.....	273
hw_insdock.....	273
hw_insertdocument.....	273
hw_insertobject	274
hw_modifyobject	274
hw_new_document	275
hw_objrec2array.....	275
hw_outputdocument.....	275
hw_pconnect.....	275
hw_pipedocument.....	275
hw_root	276
hw_unlock	276
hw_who.....	276
hw_username	276
A.8 Datenbankfunktionen	277
DBM-Funktionen.....	277
dbmopen	277
dbmclose	277
dbmexists	277
dbmfetch	277
dbminsert.....	278
dbmreplace	278

dbmdelete.....	278
dbmfirstkey.....	278
dbmnextkey	278
dblist	279
dBase-Funktionen.....	279
dbase_create.....	279
dbase_open	280
dbase_close	280
dbase_pack.....	280
dbase_add_record.....	280
dbase_replace_record.....	281
dbase_delete_record.....	281
dbase_get_record	281
dbase_numfields	281
dbase_numrecords.....	282
filePro-Funktionen	282
filepro.....	282
filepro_fieldname.....	282
filepro_fieldtype.....	282
filepro_fieldwidth	283
filepro_retrieve	283
filepro_fieldcount.....	283
filepro_rowcount.....	283
Informix-Funktionen.....	283
ifx_connect	283
ifx_pconnect.....	284
ifx_close.....	285
ifx_query	285

ifx_prepare.....	286
ifx_do.....	286
ifx_error.....	287
ifx_errormsg.....	287
ifx_affected_rows.....	287
ifx_getsqlca.....	288
ifx_fetch_row.....	288
ifx_htmltbl_result.....	289
ifx_fieldtypes.....	290
ifx_fieldproperties.....	290
ifx_num_fields.....	291
ifx_num_rows.....	291
ifx_free_result.....	291
ifx_create_char.....	291
ifx_free_char.....	291
ifx_update_char.....	292
ifx_get_char.....	292
ifx_create_blob.....	292
ifx_copy_blob.....	292
ifx_free_blob.....	292
ifx_get_blob.....	293
ifx_update_blob.....	293
ifx_blobinfile_mode.....	293
ifx_textasvarchar.....	293
ifx_byteasvarchar.....	293
ifx_nullformat.....	294
ifxus_create_slob.....	294
ifx_free_slob.....	294

ifxus_close_slob.....	294
ifxus_open_slob.....	294
ifxus_tell_slob.....	295
ifxus_seek_slob.....	295
ifxus_read_slob.....	295
ifxus_write_slob.....	295
MS-SQL-Funktionen	296
mssql_affected_rows	296
mssql_close	296
mssql_connect	296
mssql_data_seek.....	297
mssql_fetch_array	297
mssql_fetch_field	297
mssql_fetch_object.....	298
mssql_fetch_row	298
mssql_field_seek	298
mssql_free_result	299
mssql_num_fields	299
mssql_num_rows	299
mssql_pconnect.....	299
mssql_query.....	300
mssql_result.....	300
mssql_select_db.....	300
mSQL-Funktionen.....	301
mysql.....	301
mysql_affected_rows.....	301
mysql_close.....	301
mysql_connect.....	301

mysql_create_db	302
mysql_createdb	302
mysql_data_seek	302
mysql_dbname	302
mysql_drop_db	302
mysql_dropdb	303
mysql_error.....	303
mysql_fetch_array	303
mysql_fetch_field.....	303
mysql_fetch_object.....	304
mysql_fetch_row.....	304
mysql_fieldname.....	304
mysql_field_seek.....	304
mysql_fieldtable.....	305
mysql_fieldtype.....	305
mysql_fieldflags	305
mysql_fieldlen	305
mysql_free_result.....	305
mysql_freeresult.....	305
mysql_list_fields	306
mysql_listfields	306
mysql_list_dbs.....	306
mysql_listdbs.....	306
mysql_list_tables.....	306
mysql_listtables.....	306
mysql_num_fields.....	307
mysql_num_rows	307
mysql_numfields.....	307

mysql_numrows.....	307
mysql_pconnect.....	307
mysql_query	307
mysql_regcase.....	308
mysql_result.....	308
mysql_select_db	308
mysql_selectdb	308
mysql_tablename	308
MySQL-Funktionen.....	309
mysql_affected_rows.....	309
mysql_close.....	309
mysql_connect.....	309
mysql_create_db	310
mysql_data_seek	310
mysql_db_query.....	310
mysql_drop_db	310
mysql_errno.....	310
mysql_error.....	311
mysql_fetch_array.....	311
mysql_fetch_field.....	311
mysql_fetch_lengths.....	312
mysql_fetch_object.....	312
mysql_fetch_row	313
mysql_field_name.....	313
mysql_field_seek.....	313
mysql_field_table.....	313
mysql_field_type.....	313
mysql_field_flags	314

mysql_field_len.....	315
mysql_free_result	315
mysql_insert_id.....	315
mysql_list_fields	315
mysql_list_dbs.....	316
mysql_list_tables.....	316
mysql_num_fields	316
mysql_num_rows	316
mysql_pconnect	316
mysql_query	317
mysql_result	317
mysql_select_db.....	317
mysql_tablename.....	317
ODBC-Funktionen.....	318
odbc_autocommit	318
odbc_binmode.....	318
odbc_close.....	319
odbc_close_all	319
odbc_commit.....	319
odbc_connect.....	319
odbc_cursor	320
odbc_do.....	320
odbc_exec.....	320
odbc_execute	320
odbc_fetch_into.....	321
odbc_fetch_row	321
odbc_field_name.....	321
odbc_field_type.....	321

odbc_field_len	322
odbc_free_result.....	322
odbc_longreadlen.....	322
odbc_num_fields.....	322
odbc_pconnect.....	323
odbc_prepare.....	323
odbc_num_rows.....	323
odbc_result.....	323
odbc_result_all	324
odbc_rollback	324
odbc_setoption	324
Oracle 7- und Oracle 8-Funktionen.....	324
ocidefinebyname	325
ocibindbyname	325
ocilogon.....	326
ocilogoff.....	327
ociexecute	327
ocicommit.....	327
ocirollback.....	327
ocinumrows	327
ocireult	328
ocifetch.....	328
ocifetchinto.....	328
ocicolumnisnull	328
ocicolumnsize	329
ora_bind	329
ora_close.....	329
ora_columnname.....	330

ora_columntype	330
ora_commit	330
ora_commitoff	330
ora_commiton	330
ora_error	331
ora_errorcode	331
ora_exec	331
ora_fetch	331
ora_getcolumn	331
ora_logoff	331
ora_logon	331
ora_open	332
ora_parse	332
ora_rollback	332
PostgreSQL-Funktionen	332
pg_close	333
pg_cmdtuples	333
pg_connect	333
pg_dbname	334
pg_errormessage	334
pg_exec	334
pg_fetch_array	334
pg_fetch_object	335
pg_fetch_row	336
pg_fieldisnull	337
pg_fieldname	337
pg_fieldnum	337
pg_fieldprtlens	337

pg_fieldsize.....	337
pg_fieldtype.....	338
pg_freeresult.....	338
pg_getlastoid.....	338
pg_host.....	338
pg_loclose.....	338
pg_locreate.....	338
pg_loopen.....	339
pg_loread.....	339
pg_loreadall.....	339
pg_lounlink.....	339
pg_lowrite.....	340
pg_numfields.....	340
pg_numrows.....	340
pg_options.....	340
pg_pconnect.....	340
pg_port.....	341
pg_result.....	341
pg_tty.....	341
Sybase-Funktionen.....	341
sybase_affected_rows.....	341
sybase_close.....	342
sybase_connect.....	342
sybase_data_seek.....	342
sybase_fetch_array.....	343
sybase_fetch_field.....	343
sybase_fetch_object.....	343
sybase_fetch_row.....	344

sybase_field_seek.....	344
sybase_free_result.....	344
sybase_num_fields	344
sybase_num_rows	345
sybase_pconnect	345
sybase_query	345
sybase_result	345
sybase_select_db.....	346
A.9 Systemnahe Funktionen.....	347
Funktionen zur Ausgabesteuerung	347
flush.....	347
ob_end_clean.....	347
ob_end_flush.....	347
ob_get_contents.....	347
ob_implicit_flush.....	348
ob_start.....	348
Verbindungsfunktionen	348
connection_aborted	348
connection_status.....	348
connection_timeout	348
ignore_user_abort.....	348
Datenverarbeitungsfunktionen	349
pack.....	349
serialize	350
unpack.....	350
unserialize.....	351
Apache-Funktionen	351
apache_lookup_uri.....	351

apache_note	352
getallheaders.....	352
virtual	352
Systemfunktionen.....	353
dl.....	353
leak.....	353
uniqid.....	353
Programmausführung	353
escapeshellcmd.....	353
exec.....	354
system.....	354
passthru.....	355
eval.....	355
die.....	355
exit.....	356
function_exists.....	356
openlog.....	356
syslog.....	356
closelog.....	356
debugger_on.....	356
debugger_off.....	356
register_shutdown_function	357
sleep	357
usleep.....	357
PHP-Informationen	357
error_log.....	357
error_reporting.....	358
getenv	358

get_cfg_var	359
get_current_user	359
get_magic_quotes_gpc.....	359
get_magic_quotes_runtime	359
getlastmod	359
getmyinode.....	360
getmypid.....	360
getmyuid.....	360
getrusage.....	360
highlight_file	361
highlight_string.....	361
ini_get.....	361
ini_restore	361
ini_set	361
phpinfo.....	362
phpversion.....	362
extension_loaded	362
putenv	362
set_magic_quotes_runtime.....	362
set_time_limit.....	363
var_dump	363
print_r.....	363
B Cross-Referenz	365
B.1 Cross-Referenz VBScript.....	365
B.2 Cross-Referenz Perl.....	375
C Server-Variablen und Statuscodes.....	381
C.1 Server-Variablen.....	381
C.2 HTTP-Statuscodes	383

Übersicht der Codes.....	383
C.3 Die Codes im Detail	385
Informationscodes - Informational Codes 1xx.....	385
Erfolgsmeldungen - Successful 2xx.....	385
Umleitungsmeldungen - Redirection 3xx.....	387
Client-Fehler - Client Error 4xx.....	388
Server-Fehler - Server Error 5xx	390
D An den Autor	393

A.4 Aufbau und Schreibstil

Die Referenz stützt sich auf die übliche Schreibweise bei der Syntaxangabe. Da PHP überwiegend aus Funktionen besteht, wird die Funktionsschreibweise primär zu finden sein. Der folgende Ausdruck ist typisch (das Dreieck kennzeichnet Syntaxbeschreibungen zusätzlich):

◆ `int function(string param1, 0|1, int [param2]);`

Wie ist diese Schreibweise zu lesen?

- **function()**. Der Funktionsname ist **fett** hervorgehoben. Wenn die Klammern zwingend erforderlich sind, gehören sie zum Funktionskopf und sind ebenso hervorgehoben.
- `int`. Vor der Funktion steht der Datentyp, den die Funktion zurückgibt. Im Beispiel gibt die Funktion also ganzzahlige Werte zurück.
- `string param1`. Dieser Parameter ist nicht optional. Erwartet wird eine Zeichenkette. In der Erklärung zur Funktion wird auf `param1` verwiesen.
- `0|1`. Diese Parameter sind ebenso nicht optional, werden aber nicht durch einen Wert ersetzt, sondern so wie angegeben eingesetzt. Das `|`-Zeichen steht für eine Alternative, Sie können also entweder 0 oder 1 schreiben.
- `int [param2]`. Dieser Parameter wird als Ganzzahl (Integer) erwartet, ist aber optional – dies geben die eckigen Klammern an.

Der Aufruf der Funktion könnte im Skript so aussehen:

```
$intvar = function("test", 1, 45);
```

void

An einigen Stellen finden Sie das Wort `void`. Dies steht für »nichts«. Manchmal sind die Parameter auch untereinander (zeilenweise) angeordnet. Dies hat keine Bedeutung für die Funktion selbst. Es dient nur der besseren Lesbarkeit. Schreiben Sie beim Programmieren immer alles in eine Zeile.

©-Hinweis

Die Referenz entstand aus der Übersetzung der PHP-Dokumentation und wird unter den Bedingungen der GPL zur Verfügung gestellt.

A.5 Alphabetische Übersicht

Die folgende Übersicht listet alle Funktionsnamen in PHP in alphabetischer Reihenfolge auf. Die Liste bezieht sich auf Version 4.0. Alle Funktionsnamen sind zugleich reservierte Namen. Vermeiden Sie die Verwendung reservierter Namen für Variablen und Konstanten. In dieser Übersicht ist jede jemals in PHP definierte Funktion zu finden, einschließlich der ersten Version, in der diese Funktion verfügbar war. Da nicht alle Funktionen dokumentiert sind und einige ausgesprochen selten benötigt werden, sind nicht immer Querverweise in die Referenz zu finden.

Schreibweisen

Achten Sie auf folgende Schreibweisen in der Liste:

- `functionsname.....865`, [3.15]. Die Funktion wird auf Seite 865 in der Referenz beschrieben, sie ist seit PHP 3.0.15 verfügbar. Entsprechend steht 3.00 für 3.0.0, 3.01 für 3.0.1 und 3.10 für 3.0.10. Es gab bis Mitte 2000, dem endgültigen Erscheinungstermin von PHP 4, insgesamt 18 Releases von PHP 3.0: 3.0.0 bis 3.0.17. **Ziel**
- `functions_name ([4b2])`. Die Funktion wird nicht beschrieben, sie ist noch nicht dokumentiert. Sie ist mit PHP 4, 2. Beta, eingeführt worden (Es gab drei öffentliche Betaversionen: 1, 2 und 4). 4RC1 steht für Release Candidate 1, RC 2 entsprechend für Release Candidate 2 (es gab zwei RCs). 4.00, 4.01, 4.02 steht für die offiziellen Final-Releases. **Version**
- `functions_name ([3.13][4RC1])`. Die Funktion ist seit 3.0.13 verfügbar, war jedoch in den ersten Betaversionen von PHP4 nicht mehr vertreten und tauchte im Release Candidate 1 wieder auf. Sie wäre also beispielsweise in PHP 3.0.16 verfügbar, in PHP 4 Beta2 jedoch nicht.
- `functions_name (aliasname)`. Der Funktionsname ist unter dem Alias *aliasname* gebräuchlicher, sehen Sie dort nach. **Aliase**

Einige »Funktionsnamen« sind fett geschrieben, beispielsweise **array** oder **die**. Dies sind keine Funktionen, sondern Sprachkonstrukte oder Anweisungen, die wie Funktionen aufgerufen werden. Da viele Leser solche Konstrukte vermutlich in der Liste suchen, habe ich sie mit aufgenommen. Um korrekt zu bleiben (und den Profis Genüge zu tun), sind sie aber gekennzeichnet. **Sprachkonstrukte**

Was nicht aufgeführt wurde

CVS-Versionen wurden in dieser Übersicht nicht beachtet, da diese Entwicklerstadien den meisten Lesern nicht zugänglich sein dürften.

Keine Aufnahme fanden auch Funktionen, die nur einige Versionen lang verfügbar waren und dann wieder verschwanden. Insbesondere sind solche nicht aufgeführt, die den Sprung von PHP3 nach PHP4 nicht überlebt haben. Sie sollten diese Funktionen nicht verwenden, auch wenn sie in anderen Quellen oder in der Dokumentation stehen.

Einige Funktionen sind auch in der offiziellen PHP-Dokumentation nicht aufgeführt und werden außerdem nur sehr selten verwendet. Alle Funktionen sind jedoch in der alphabetischen Liste zu finden. Steht keine Seitenangabe dahinter, erscheint sie nicht in der Referenz.

A

- abs..... 152, [3.00]
- acos..... 152, [3.00]
- ada_afetch ([3.00])
- ada_autocommit ([3.00])
- ada_close ([3.00])
- ada_commit ([3.00])
- ada_connect ([3.00])
- ada_exec ([3.00])
- ada_fetchrow ([3.00])
- ada_fieldname ([3.00])
- ada_fieldnum ([3.00])
- ada_fielddtype ([3.00])
- ada_freeresult ([3.00])
- ada_numfields ([3.00])
- ada_numrows ([3.00])
- ada_result ([3.00])
- ada_resultall ([3.00])
- ada_rollback ([3.00])
- addslashes..... 115, [4b4]
- addslashes..... 115, [3.00]
- add_root (*domxml_add_root*)
- apache_lookup_uri..... 351, [3.04]
- apache_note..... 352, [3.02]
- array**..... 98, [3.00]
- array_count_values..... 107, [4b4]
- array_diff..... 107, [4.01]
- array_flip..... 108, [4b4]
- array_keys..... 107, [4b1]
- array_merge..... 108, [4b1]
- array_merge_recursive..... 109, [4.01]
- array_multisort..... 109, [4b4]
- array_pad..... 109, [4b4]
- array_pop..... 110, [4b1]
- array_push..... 110, [4b1]
- array_reverse..... 110, [4b4]
- array_shift..... 111, [4b1]
- array_slice..... 112, [4b1]
- array_splice..... 112, [4b1]
- array_unique..... 111, [4.01]
- array_unshift..... 112, [4b1]
- array_values..... 113, [4b1]
- array_walk..... 99, [3.03]
- arsort..... 99, [3.00]
- asin..... 152, [3.00]
- asort..... 100, [3.00]
- aspell_check ([3.07] *pspell_check*)
- aspell_check_raw ([3.07])
- aspell_new ([3.07] *pspell_new*)
- aspell_suggest ([3.07] *pspell_suggest*)
- assert ([4b4])
- assert_options ([4b4])
- atan..... 152, [3.00]
- atan2..... 153, [3.05]
- attributes (*domxml_attributes*)

B

- base64_decode..... 92, [3.00]
- base64_encode..... 92, [3.00]
- base_convert..... 153, [3.06]
- basename..... 163, [3.00]
- bcadd..... 158, [3.00]
- bccomp..... 159, [3.00]

<code>bcdiv</code>	159, [3.00]	<code>compact</code>	113, [4b1]
<code>bcmod</code>	159, [3.00]	com	88, [4b2]
<code>bcmul</code>	159, [3.00]	<code>com_get</code> (<i>com_propget</i>)	
<code>bcpow</code>	160, [3.00]	<code>com_invoke</code>	88, [3.03]
<code>bcscale</code>	160, [3.00]	<code>com_load</code>	88, [3.03]
<code>bcsqrt</code>	160, [3.00]	<code>com_propget</code>	88, [3.03]
<code>bcsub</code>	160, [3.00]	<code>com_propput</code>	89, [3.03]
<code>bin2hex</code>	116, [3.10]	<code>com_propset</code> (<i>com_propput</i>)	
<code>bindec</code>	153, [3.00]	<code>com_set</code> (<i>com_propset</i>)	
<code>bindtextdomain</code>	134, [3.07]	<code>connection_aborted</code>	348, [3.07]
break	82, [3.00]	<code>connection_status</code>	348, [3.07]
C		<code>connection_timeout</code>	348, [3.07]
<code>call_user_func</code>	94, [3.03]	continue	83, [3.00]
<code>call_user_method</code>	86, [3.03]	<code>convert_cyr_string</code>	117, [3.06]
<code>ceil</code>	153, [3.00]	<code>copy</code>	164, [3.00]
<code>chdir</code>	161, [3.00]	<code>cos</code>	153, [3.00]
<code>checkdate</code>	142, [3.00]	<code>count</code>	100, [3.00]
<code>checkdnsrr</code>	184, [3.00]	<code>count_chars</code>	117, [4b4]
<code>chgrp</code>	163, [3.00]	<code>cpdf_add_annotation</code> ([3.12])	
<code>children</code> (<i>domxml_children</i>)		<code>cpdf_add_outline</code> ([3.09])	
<code>chmod</code>	163, [3.00]	<code>cpdf_arc</code> ([3.08])	
<code>chop</code>	116, [3.00]	<code>cpdf_begin_text</code> ([3.08])	
<code>chown</code>	163, [3.00]	<code>cpdf_circle</code> ([3.08])	
<code>chr</code>	116, [3.00]	<code>cpdf_clip</code> ([3.08])	
<code>chunk_split</code>	116, [3.06]	<code>cpdf_close</code> ([3.08])	
class	84, [3.00]	<code>cpdf_closepath</code> ([3.08])	
<code>class_exists</code>	87, [4.01]	<code>cpdf_closepath_fill_stroke</code> ([3.08])	
<code>clearstatcache</code>	164, [3.00]	<code>cpdf_closepath_stroke</code> ([3.08])	
<code>close</code> (<i>closedir</i>)		<code>cpdf_continue_text</code> ([3.08])	
<code>closedir</code>	161, [3.00]	<code>cpdf_curveto</code> ([3.08])	
<code>closelog</code>	356, [3.00]	<code>cpdf_end_text</code> ([3.08])	

<code>cpdf_endpath</code> ([3.08])	<code>cpdf_set_text_matrix</code> ([3.08])
<code>cpdf_fill</code> ([3.08])	<code>cpdf_set_text_pos</code> ([3.08])
<code>cpdf_fill_stroke</code> ([3.08])	<code>cpdf_set_text_rendering</code> ([3.08])
<code>cpdf_finalize</code> ([3.08])	<code>cpdf_set_text_rise</code> ([3.08])
<code>cpdf_finalize_page</code> ([3.10])	<code>cpdf_set_title</code> ([3.08])
<code>cpdf_import_jpeg</code> ([3.09])	<code>cpdf_set_word_spacing</code> ([3.08])
<code>cpdf_lineto</code> ([3.08])	<code>cpdf_setdash</code> ([3.08])
<code>cpdf_moveto</code> ([3.08])	<code>cpdf_setflat</code> ([3.08])
<code>cpdf_newpath</code> ([3.09])	<code>cpdf_setgray</code> ([3.08])
<code>cpdf_open</code> ([3.08])	<code>cpdf_setgray_fill</code> ([3.08])
<code>cpdf_output_buffer</code> ([3.09])	<code>cpdf_setgray_stroke</code> ([3.08])
<code>cpdf_page_init</code> ([3.08])	<code>cpdf_setlinecap</code> ([3.08])
<code>cpdf_place_inline_image</code> ([3.09])	<code>cpdf_setlinejoin</code> ([3.08])
<code>cpdf_rect</code> ([3.08])	<code>cpdf_setlinewidth</code> ([3.08])
<code>cpdf_restore</code> ([3.08])	<code>cpdf_setmiterlimit</code> ([3.08])
<code>cpdf_rlineto</code> ([3.09])	<code>cpdf_setrgbcolor</code> ([3.08])
<code>cpdf_rmoveto</code> ([3.09])	<code>cpdf_setrgbcolor_fill</code> ([3.08])
<code>cpdf_rotate</code> ([3.08])	<code>cpdf_setrgbcolor_stroke</code> ([3.08])
<code>cpdf_rotate_text</code> ([3.09])	<code>cpdf_show</code> ([3.08])
<code>cpdf_save</code> ([3.08])	<code>cpdf_show_xy</code> ([3.08])
<code>cpdf_save_to_file</code> ([3.08])	<code>cpdf_stringwidth</code> ([3.08])
<code>cpdf_scale</code> ([3.08])	<code>cpdf_stroke</code> ([3.08])
<code>cpdf_set_char_spacing</code> ([3.08])	<code>cpdf_text</code> ([3.08])
<code>cpdf_set_creator</code> ([3.08])	<code>cpdf_translate</code> ([3.08])
<code>cpdf_set_current_page</code> ([3.09])	<code>crc32</code> 118, [4.01]
<code>cpdf_set_font</code> ([3.08])	<code>crash</code> ([4.01])
<code>cpdf_set_horiz_scaling</code> ([3.08])	<code>crypt</code> 118, [3.08]
<code>cpdf_set_keywords</code> ([3.08])	<code>current</code> 100, [3.08]
<code>cpdf_set_leading</code> ([3.08])	<code>cybercash_base64_decode</code> ([4b4])
<code>cpdf_set_page_animation</code> ([3.08])	<code>cybercash_base64_decode</code> ([4b4])
<code>cpdf_set_subject</code> ([3.08])	<code>cybercash_decr</code> ([4b4])

cybercash_encr ([4b4])		dbmfetch277, [3.00]
D		dbmfirstkey278, [3.00]
date142, [3.00]		dbminsert278, [3.00]
dav_set_mkol_handlers ([3.06])		dbmnextkey278, [3.00]
dba_close ([3.08])		dbmopen277, [3.00]
dba_delete ([3.08])		dbmreplace278, [3.00]
dba_exists ([3.08])		dcgettext135, [3.08]
dba_fetch ([3.08])		debugger_off356, [3.00]
dba_firstkey ([3.08])		debugger_on356, [3.00]
dba_insert ([3.08])		decbin153, [3.00]
dba_nextkey ([3.08])		dechex154, [3.00]
dba_open ([3.08])		decoct154, [3.00]
dba_optimize ([3.08])		decrypt ([3.00])
dba_popen ([3.08])		define89, [3.00]
dba_replace ([3.08])		defined89, [3.00]
dba_sync ([3.08])		define_syslog_variables ([3.00])
dbase_add_record280, [3.00]		deg2rad154, [3.04]
dbase_close280, [3.00]		dgettext135, [3.07]
dbase_create279, [3.00]		delete178, [3.00]
dbase_delete_record281, [3.00]		die355, [3.00]
dbase_get_record281, [3.00]		dir (<i>getdir</i>)161, [3.00]
dbase_get_record_with_names ([3.05])		dirname164, [3.00]
dbase_numfields281, [3.00]		diskfreespace164, [3.07]
dbase_numrecords282, [3.00]		d1353, [3.00]
dbase_open280, [3.00]		domxml_add_root ([4b4])
dbase_pack280, [3.00]		domxml_attributes ([4b4])
dbase_replace_record281, [3.11]		domxml_children ([4b4])
dblist279, [3.00]		domxml_dumpmem ([4b4])
dbmclose277, [3.00]		domxml_getattr ([4b4])
dbmdelete278, [3.00]		domxml_new_child ([4b4])
dbmexists277, [3.00]		domxml_new_xmldoc ([4b4])

domxml_node ([4b4])	extension_loaded,..... 362, [3.10]
domxml_root ([4b4])	extract..... 114, [3.07]
domxml_setattr ([4b4])	F
domxml_test ([4b4])	fclose..... 165, [3.00]
dtd (<i>domxml_intdtd</i>)	fdf_add_template ([3.13])
dumpmem (<i>domxml_dumpmem</i>)	fdf_close ([3.06])
doubleval..... 95, [3.00]	fdf_create ([3.06])
E	fdf_get_file ([3.06])
each..... 101, [3.00]	fdf_get_status ([3.06])
easter_date..... 151, [3.09]	fdf_get_value ([3.06])
easter_days..... 151, [3.09]	fdf_next_field_name ([3.06])
echo 118, [3.00]	fdf_open ([3.06])
empty 95, [3.00]	fdf_save ([3.06])
else 80, [3.00]	fdf_set_ap ([3.06])
elseif 80, [3.00]	fdf_set_file ([3.06])
encrypt ([3.00])	fdf_set_status ([3.06])
end..... 102, [3.00]	fdf_set_value ([3.06])
endif (<i>if</i>)	feof..... 165, [3.00]
ereg..... 136, [3.00]	fgetc..... 165, [3.00]
ereg_replace..... 136, [3.00]	fgetcsv..... 165, [3.08]
eregi..... 137, [3.00]	fgets..... 166, [3.00]
eregi_replace..... 137, [3.00]	fgetss..... 166, [3.00]
error_log..... 357, [3.00]	file..... 166, [3.00]
error_reporting..... 358, [3.00]	file_exists..... 166, [3.00]
escapeshellcmd..... 353, [3.00]	fileatime..... 166, [3.00]
eval 355, [3.00]	filectime..... 167, [3.00]
exec..... 354, [3.00]	filegroup..... 167, [3.00]
exit..... 356, [3.00]	fileinode..... 167, [3.00]
exp..... 154, [3.00]	filemtime..... 167, [3.00]
explode..... 119, [3.00]	fileowner..... 168, [3.00]
extends 86, [3.00]	fileperms..... 168, [3.00]

- filepro282, [3.00]
- filepro_fieldcount283, [3.00]
- filepro_fieldname282, [3.00]
- filepro_fieldtype282, [3.00]
- filepro_fieldwidth283, [3.00]
- filepro_retrieve283, [3.00]
- filepro_rowcount283, [3.00]
- filesize168, [3.00]
- filetype168, [3.00]
- flock168, [3.07]
- floor154, [3.00]
- flush347, [3.00]
- for**81, [3.00]
- foreach82, [4.00]
- fopen169, [3.00]
- fpass thru170, [3.00]
- fputs (*fwrite*)170, [3.00]
- fread170, [3.00]
- free (*oci8_free*, [4b4])
- frenchtojd150, [3.00]
- fseek171, [3.00]
- fsockopen182, [3.00]
- fstat ([4RC1])
- ftell171, [3.00]
- ftp_cdup186, [3.13][4b4]
- ftp_chdir185, [3.13][4b4]
- ftp_connect185, [3.13][4b4]
- ftp_delete189, [3.13][4b4]
- ftp_fget187, [3.13][4b4]
- ftp_fput188, [3.13][4b4]
- ftp_get187, [3.13][4b4]
- ftp_login185, [3.13][4b4]
- ftp_mdtm188, [3.13][4b4]
- ftp_mkdir186, [3.13][4b4]
- ftp_nlist186, [3.13][4b4]
- ftp_pasv187, [3.13][4b4]
- ftp_put187, [3.13][4b4]
- ftp_pwd185, [3.13][4b4]
- ftp_quit189, [3.13][4b4]
- ftp_rawlist186, [3.13][4b4]
- ftp_rename188, [3.13][4b4]
- ftp_rmdir186, [3.13][4b4]
- ftp_site [3.13][4b4]
- ftp_size188, [3.13][4b4]
- ftp_systype186, [3.13][4b4]
- ftruncate171, [4RC1]
- function**84, [3.00]
- function_exists ([3.07])
- func_get_arg89, [4RC1]
- func_get_args89, [4RC1]
- func_num_args89, [4RC1]
- fwrite171, [3.00]
- G**
- gamma_correct_tag ([3.13][+4])
- getallheaders352, [3.00]
- getattr (*domxml_getaddr*)
- getcwd162, [4b4]
- getdate145, [3.00]
- getenv358, [3.00]
- gethostbyaddr184, [3.00]
- gethostbyname184, [3.00]
- gethostbyname1184, [3.00]

- getimagesize..... 252, [3.00]
- getlastaccess ([3.00])
- getlastemail ([3.00])
- getlasthost ([3.00])
- getlastmod..... 359, [3.00]
- getlastref ([3.00])
- getlogdir ([3.00])
- getloghost ([3.00])
- getmxrr..... 185, [3.00]
- getmyiid ([3.00][+4])
- getmyinode..... 360, [3.00]
- getmypid..... 360, [3.00]
- getmyuid..... 360, [3.00]
- getprotobyname ([4b4])
- getprotobynumber ([4b4])
- getrandmax..... 154, [3.00]
- getrusage..... 360, [3.07]
- getservbyname ([4b4])
- getservbyport ([4b4])
- getstartlogging ([3.00])
- gettext..... 135, [3.07]
- gettimeofday..... 146, [3.07]
- gettype..... 94, [3.00]
- getbrowser ([3.00])
- get_cfg_var..... 359, [3.00]
- get_class..... 86, [4.00]
- get_class_methods..... 86, [4.00]
- get_class_vars..... 87, [4.00]
- get_current_user..... 359, [3.00]
- get_declared_classes..... 87, [4.00]
- get_extension_funcs ([4b2])
- get_html_translation_table.... 119, [4b2]
- get_included_files ([4.00])
- get_loaded_extensions ([4b2])
- get_magic_quotes_gpc..... 359, [3.06]
- get_magic_quotes_runtime.... 359, [3.06]
- get_meta_tags..... 119, [3.06]
- get_object_vars..... 87, [4.00]
- get_parent_class..... 86, [4.00]
- get_required_files ([4.00])
- gmdate..... 146, [3.00]
- gmmktime..... 148, [3.00]
- gmstrftime..... 145, [3.12]
- gregoriantojd..... 149, [3.00]
- gzclose ([3.00])
- gzEOF ([3.00])
- gzfile ([3.00])
- gzgetc ([3.00])
- gzgets ([3.00])
- gzgetss ([3.00])
- gzopen ([3.00])
- gzpassthru ([3.00])
- gzputs (*gzwrite*)
- gzread ([3.00])
- gzrewind ([3.00])
- gzseek ([3.00])
- gzte11 ([3.00])
- gzwrite ([3.00])
- ## H
- header..... 93, [3.00]
- headers_sent..... 93, [3.08]
- hebrew ([3.00])

hebrevc ([3.00])	hw_getobject270, [3.03]
hexdec154, [3.00]	hw_getobjectbyquery271, [3.03]
highlight_file361, [4b1]	hw_getobjectbyquerycoll271, [3.03]
highlight_string361, [4b1]	hw_getobjectbyquerycollobj ...272, [3.03]
htmlentities120, [3.00]	hw_getobjectbyqueryobj271, [3.03]
htmlspecialchars120, [3.00]	hw_getparents269, [3.03]
hw_array2objrec266, [3.04]	hw_getparentsobj269, [3.03]
hw_children266, [3.03]	hw_getremote270, [3.03]
hw_childrenobj266, [3.03]	hw_getremotechildren270, [3.03]
hw_close266, [3.03]	hw_getsrcbydestobj270, [3.03]
hw_connect267, [3.03]	hw_gettext270, [3.03]
hw_cp267, [3.03]	hw_identify273, [3.03]
hw_deleteobject267, [3.03]	hw_incollections273, [3.03]
hw_docbyanchor267, [3.03]	hw_info273, [3.03]
hw_docbyanchorobj267, [3.03]	hw_inscoll273, [3.03]
hw_documentattributes267, [3.03]	hw_insdoc273, [3.03]
hw_documentbodytag268, [3.03]	hw_insertdocument273, [3.03]
hw_documentcontent268, [3.03]	hw_insertobject274, [3.03]
hw_documentsetcontent268, [3.03]	hw_modifyobject274, [3.03]
hw_documentsize268, [3.03]	hw_mv272, [3.03]
hw_edittext268, [3.03]	hw_new_document275, [3.03]
hw_error269, [3.03]	hw_objrec2array275, [3.03]
hw_errormsg268, [3.03]	hw_outputdocument275, [3.03]
hw_free_document269, [3.03]	hw_pconnect275, [3.03]
hw_getanchors272, [3.03]	hw_pipedocument275, [3.03]
hw_getanchorsobj272, [3.03]	hw_root276, [3.03]
hw_getandlock270, [3.03]	hw_setlinkroot ([3.03])
hw_getchildcoll269, [3.03]	hw_stat ([3.03])
hw_getchildcollobj269, [3.03]	hw_unlock276, [3.03]
hw_getchilddoccoll272, [3.03]	hw_username276, [3.03]
hw_getchilddoccollobj272, [3.03]	hw_who276, [3.03]

I	
ibase_bind (→ CD, [3.07])	ifx_byteasvarchar..... 293, [3.00]
ibase_close (→ CD, [3.07])	ifx_close..... 285, [3.00]
ibase_connect (→ CD, [3.07])	ifx_connect..... 283, [3.00]
ibase_execute (→ CD, [3.07])	ifx_copy_blob..... 292, [3.02]
ibase_fetch_row (→ CD, [3.07])	ifx_create_blob..... 292, [3.02]
ibase_free_query (→ CD, [3.07])	ifx_create_char..... 291, [3.03]
ibase_free_result (→ CD, [3.07])	ifx_do..... 286, [3.01]
ibase_pconnect (→ CD, [3.07])	ifx_error..... 287, [3.00]
ibase_prepare (→ CD, [3.07])	ifx_errormsg..... 287, [3.01]
ibase_query (→ CD, [3.07])	ifx_fetch_row..... 288, [3.01]
ibase_timefmt (→ CD, [3.07])	ifx_fieldproperties..... 290, [3.01]
icap_close (→ CD, [3.07])	ifx_fielddtypes..... 290, [3.01]
icap_delete_event (→ CD, [3.07])	ifx_free_blob..... 292, [3.01]
icap_fetch_event (→ CD, [3.07])	ifx_free_char..... 291, [3.04]
icap_list_alarms (→ CD, [3.07])	ifx_free_result..... 291, [3.01]
icap_list_events (→ CD, [3.07])	ifx_get_blob..... 293, [3.02]
icap_open (→ CD, [3.07])	ifx_get_char..... 292, [3.04]
icap_snooze (→ CD, [3.07])	ifx_getsqlca..... 288, [3.06]
icap_store_event (→ CD, [3.07])	ifx_htmltbl_result..... 289, [3.00]
if 80, [3.00]	ifx_nullformat..... 294, [3.01]
ifxus_close_slob (→ CD, [3.00])	ifx_num_fields..... 291, [3.00]
ifxus_create_slob (→ CD, [3.00])	ifx_num_rows..... 291, [3.00]
ifxus_free_slob (→ CD, [3.00])	ifx_pconnect..... 284, [3.00]
ifxus_open_slob (→ CD, [3.00])	ifx_prepare..... 286, [3.01]
ifxus_read_slob (→ CD, [3.00])	ifx_query..... 285, [3.03]
ifxus_seek_slob (→ CD, [3.00])	ifx_textasvarchar..... 293, [3.04]
ifxus_tell_slob (→ CD, [3.00])	ifx_update_blob..... 293, [3.04]
ifxus_write_slob (→ CD, [3.00])	ifx_update_char..... 292, [3.06]
ifx_affected_rows..... 287, [3.00]	ifxus_close_slob..... 294, [3.01]
ifx_blobinfile_mode..... 293, [3.00]	ignore_user_abort..... 348, [3.07]
	imagearc..... 253, [3.00]

imagechar	253, [3.00]	imageloadfont	258, [3.00]
imagecharup	253, [3.00]	imagepolygon	259, [3.00]
imagecolorallocate	253, [3.00]	imagepsbbox	265, [3.08]
imagecolordeallocate ([3.06])		imagepsencodefont	264, [3.08]
imagecolorat	262, [3.00]	imagepsfreefont	264, [3.08]
imagecolorclosest	262, [3.00]	imagepsloadfont (→ CD, [3.08])	
imagecolorexact	262, [3.00]	imagepslantfont (→ CD, [3.08])	
imagecolorresolve	262, [3.00]	imagepstext (→ CD, [3.08])	
imagecolorset	262, [3.00]	imagerectangle	259, [3.00]
imagecolorsforindex	263, [3.00]	imagesetpixel	259, [3.00]
imagecolorstotal	263, [3.00]	imagestring	259, [3.00]
imagecolortransparent	254, [3.00]	imagestringup	259, [3.00]
imagecopy ([3.06])		imagesx	260, [3.00]
imagecopyresized	254, [3.00]	imagesy	260, [3.00]
imagecreate	254, [3.00]	imagettfbbox	260, [3.01]
imagecreatefromgif	255, [3.00]	imagettftext	261, [3.00]
imagecreatefromjpeg	255, [3.16]	imagewbmp ([3.14] [+4])	
imagecreatefrompng	255, [3.13]	imap_8bit	241, [3.00]
imagedashedline	255, [3.00]	imap_alerts ([3.12])	
imagedestroy	256, [3.00]	imap_append	233, [3.00]
imagefill	256, [3.00]	imap_base64	233, [3.00]
imagefilledpolygon	256, [3.00]	imap_binary ([3.02])	
imagefilledrectangle	256, [3.00]	imap_body	233, [3.00]
imagefilltoborder	256, [3.00]	imap_bodystruct ([3.00])	
imagefontheight	257, [3.00]	imap_check	233, [3.00]
imagefontwidth	257, [3.00]	imap_clearflag_full ([3.04])	
imagegif	257, [3.00]	imap_close	234, [3.00]
imagejpeg	257, [3.16]	imap_create (<i>imap_createmailbox</i>)	
imagepng	257, [3.16]	imap_createmailbox	234, [3.00]
imageinterlace	258, [3.00]	imap_delete	234, [3.00]
imageline	258, [3.00]	imap_deletemailbox	234, [3.00]

imap_errors (→ CD, [3.12])
imap_expunge..... 234, [3.00]
imap_fetchbody..... 234, [3.00]
imap_fetchheader (→ CD, [3.05])
imap_fetchstructure..... 235, [3.00]
imap_fetchtext (*imap_body*)
imap_fetch_overview (→ CD, [3.07])
imap_getmailboxes (→ CD, [3.12])
imap_getsubscribed (→ CD, [3.12])
imap_header..... 235, [3.00]
imap_headerinfo (→ CD, [3.00])
imap_headers..... 237, [3.00]
imap_last_error (→ CD, [3.12])
imap_list (*imap_listmailbox*)
imap_listmailbox..... 238, [3.00]
imap_listsubscribed..... 238, [3.00]
imap_lsub (*imap_listsubscribed*)
imap_mail (→ CD, [3.14])
imap_mail_compose (→ CD, [3.06])
imap_mail_copy..... 238, [3.00]
imap_mail_move..... 238, [3.00]
imap_mailboxmsginfo..... 241, [3.02]
imap_msgno (→ CD, [3.04])
imap_num_msg..... 238, [3.00]
imap_num_recent..... 239, [3.00]
imap_open..... 239, [3.00]
imap_ping..... 240, [3.00]
imap_popen (→ CD, [3.12])
imap_qprint..... 241, [3.00]
imap_renamemailbox..... 240, [3.00]
imap_reopen..... 240, [3.00]
imap_rfc822_parse_adrlist.... 242, [3.02]
imap_rfc822_parse_headers..... 242, [4b4]
imap_rfc822_write_address.... 242, [3.02]
imap_scan (*imap_scanmailbox*)
imap_scanmailbox..... 241, [3.00]
imap_search (→ CD, [3.12])
imap_setflag_full..... 242, [3.03]
imap_sort..... 243, [3.03]
imap_status (→ CD, [3.04])
imap_subscribe (→ CD, [3.00])
imap_uid..... 244, [3.03]
imap_undelete (→ CD, [3.00])
imap_unsubscribe (→ CD, [3.00])
imap_utf7_decode (→ CD, [3.15])
imap_utf7_encode (→ CD, [3.15])
imap_utf8 (→ CD, [3.13])
implode..... 120, [3.00]
ini_alter (*ini_set*)
ini_get..... 361, [4b1]
ini_restore..... 361, [4b1]
ini_set..... 361, [4b1]
intval..... 95, [3.00]
in_array..... 114, [4b1]
ip2long..... 183, [4b4]
include..... 84, [3.00]
iptcembed (→ CD, [3.07])
iptcparse..... 265, [3.07]
is_array..... 95, [3.00]
is_dir..... 172, [3.00]
is_double..... 96, [3.00]
is_executable..... 172, [3.00]

<code>is_file</code>	172, [3.00]	<code>lcg_value</code> (→ CD, [4b4])
<code>is_float</code> (<i>is_double</i>)		<code>ldap_add</code>
<code>is_int</code> (<i>is_integer</i>)		244, [3.00]
<code>is_integer</code>	96, [3.00]	<code>ldap_bind</code>
<code>is_link</code>	173, [3.00]	245, [3.00]
<code>is_long</code>	96, [3.00]	<code>ldap_close</code>
<code>is_numeric</code> (→ CD, [4b4])		245, [3.00]
<code>is_object</code>	97, [3.00]	<code>ldap_connect</code>
<code>is_readable</code>	173, [3.00]	246, [3.00]
<code>is_real</code> (<i>is_double</i>)		<code>ldap_count_entries</code>
<code>is_string</code>	97, [3.00]	246, [3.00]
<code>is_subclass_of</code> (→ CD, [4.00])		<code>ldap_delete</code>
<code>is_writable</code> (<i>is_writeable</i>)		246, [3.00]
<code>is_writeable</code>	173, [3.00]	<code>ldap_dn2ufn</code>
isset	97, [3.00]	246, [3.00]
J		<code>ldap_err2str</code> (→ CD, [3.13][4.00])
<code>jddayofweek</code>	150, [3.00]	<code>ldap_errno</code> (→ CD, [3.12][4.00])
<code>jdmonthname</code>	150, [3.00]	<code>ldap_error</code> (→ CD, [3.12][4.00])
<code>jdtofrrench</code>	150, [3.00]	<code>ldap_explode_dn</code>
<code>jdtogregorian</code>	148, [3.00]	246, [3.00]
<code>jdtojewish</code>	149, [3.00]	<code>ldap_first_attribute</code>
<code>jdtojulian</code>	149, [3.00]	247, [3.00]
<code>jdtonix</code> ([4.00])		<code>ldap_first_entry</code>
<code>jewishtojd</code>	149, [3.00]	247, [3.00]
<code>join</code>	121, [3.00]	<code>ldap_free_result</code>
<code>juliantojd</code>	149, [3.00]	247, [3.00]
K		<code>ldap_get_attributes</code>
<code>key</code>	102, [3.00]	247, [3.00]
<code>krsort</code>	[3.13]	<code>ldap_get_dn</code>
<code>ksort</code>	102, [3.00]	248, [3.00]
<code>lastchild</code> (<i>domxml_lastchild</i>)		<code>ldap_get_entries</code>
		248, [3.00]
		<code>ldap_get_values</code>
		249, [3.00]
		<code>ldap_get_values_len</code>
		249, [4.00]
		<code>ldap_list</code>
		249, [3.00]
		<code>ldap_mod_add</code> (→ CD, [3.00])
		<code>ldap_mod_del</code> (→ CD, [3.00])
		<code>ldap_mod_replace</code> (→ CD, [3.00])
		<code>ldap_modify</code>
		250, [3.00]
		<code>ldap_next_attribute</code>
		250, [3.00]
		<code>ldap_next_entry</code>
		250, [3.00]
		<code>ldap_read</code>
		250, [3.00]
		<code>ldap_search</code>
		251, [3.00]
		<code>ldap_unbind</code>
		251, [3.00]
		<code>leak</code>
		353, [3.00]

link.....	174, [3.00]	mcal_event_set_end ([3.13][4b4])
linkinfo.....	174, [3.00]	mcal_event_set_end_now ([3.13][4b4])
list	103, [3.00]	mcal_event_set_recur_daily ([3.13][4b4])
load (<i>ociloadlob</i>)		mcal_event_set_recur_monthly_mday ([3.13][4b4])
localtime.....	146, [4.00]	mcal_event_set_recur_monthly_wday ([3.13][4b4])
log.....	155, [3.00]	mcal_event_set_recur_none ([3.13][4b4])
log10.....	155, [3.00]	mcal_event_set_recur_weekly ([3.13][4b4])
logas ([3.00])		mcal_event_set_recur_yearly ([3.13][4b4])
long2ip.....	183, [4b4]	mcal_event_set_start ([3.13][4b4])
lstat.....	176, [3.04]	mcal_event_set_start_now ([3.13][4b4])
ltrim.....	121, [3.00]	mcal_event_set_title ([3.13][4b4])
M		mcal_fetch_current_stream_event ([3.13][4b4])
mail.....	232, [3.00]	mcal_fetch_event ([3.13][4b4])
magic_quotes_runtime (<i>set_magic_quotes_runtime</i>)		mcal_is_leap_year ([3.13][4b4])
max.....	155, [3.00]	mcal_list_alarms ([3.13][4b4])
mcal_append_event ([3.13][4b4])		mcal_list_events ([3.13][4b4])
mcal_close ([3.13][4b4])		mcal_next_recurrence ([3.13][4b4])
mcal_create_calendar ([3.13][4b4])		mcal_open ([3.13][4b4])
mcal_date_compare ([3.13][4b4])		mcal_popen ([3.13][4b4])
mcal_date_valid ([3.13][4b4])		mcal_rename_calendar ([3.13][4b4])
mcal_days_in_month ([3.13][4b4])		mcal_reopen ([3.13][4b4])
mcal_day_of_week ([3.13][4b4])		mcal_snooze ([3.13][4b4])
mcal_day_of_year ([3.13][4b4])		mcal_store_event ([3.13][4b4])
mcal_delete_calendar ([3.13][4b4])		mcal_time_valid ([3.13][4b4])
mcal_delete_event ([3.13][4b4])		mcal_week_of_year ([3.13][4b4])
mcal_event_add_attribute ([3.13][4b4])		mck_base64_decode ([3.07])
mcal_event_init ([3.13][4b4])		
mcal_event_set_alarm ([3.13][4b4])		
mcal_event_set_category ([3.13][4b4])		
mcal_event_set_class ([3.13][4b4])		
mcal_event_set_description ([3.13][4b4])		

mck_base64_encode ([3.07])	msql_drop_db 302, [3.00]
mck_decr ([3.07])	msql_dropdb (<i>msql_drop_db</i>)
mck_encr ([3.07])	msql_error 303, [3.00]
mcrypt_cbc 191, [3.09]	msql_fetch_array 303, [3.00]
mcrypt_cfb 191, [3.09]	msql_fetch_field 303, [3.00]
mcrypt_create_iv 191, [3.09]	msql_fetch_object 304, [3.00]
mcrypt_ecb 192, [3.09]	msql_fetch_row 304, [3.00]
mcrypt_get_block_size 190, [3.09]	msql_field_seek 304, [3.00]
mcrypt_get_cipher_name 190, [3.09]	msql_field_flags 305, [3.07]
mcrypt_get_key_size 190, [3.09]	msql_field_len 305, [3.07]
mcrypt_ofb 192, [3.09]	msql_field_name 304, [3.07]
md5 121, [3.00]	msql_field_table 305, [3.07]
metaphone 121, [4b4]	msql_field_type 305, [3.07]
method_existes 87, [4.00]	msql_fieldflags (<i>msql_field_flags</i>)
mhash 194, [3.09]	msql_fieldlen (<i>msql_field_len</i>)
mhash_count 193, [3.09]	msql_fieldname (<i>msql_field_name</i>)
mhash_get_block_size 193, [3.09]	msql_fielddata (<i>msql_field_table</i>)
mhash_get_hash_name 193, [3.09]	msql_fieldtype (<i>msql_field_type</i>)
microtime 148, [3.00]	msql_free_result 305, [3.00]
min 155, [3.00]	msql_freeresult (<i>msql_free_result</i>)
mkdir 174, [3.00]	msql_list_dbs 306, [3.00]
mktime 147, [3.00]	msql_listdbs (<i>msql_list_dbs</i>)
msql 301, [3.00]	msql_list_fields 306, [3.00]
msql_affected_rows 301, [3.00]	msql_listfields (<i>msql_list_fields</i>)
msql_close 301, [3.00]	msql_list_tables 306, [3.00]
msql_connect 301, [3.00]	msql_listtables (<i>msql_list_tables</i>)
msql_create_db 302, [3.00]	msql_num_fields 307, [3.00]
msql_createdb (<i>msql_create_db</i>)	msql_numfields (<i>msql_num_fields</i>)
msql_data_seek 302, [3.00]	msql_num_rows 307, [3.00]
msql_dbname 302, [3.00]	msql_numrows (<i>msql_num_rows</i>)
msql_dbquery , [3.00]	msql_pconnect 307, [3.00]

<code>mssql_query</code>	307, [3.00]	<code>mysql_db_query</code>	310, [3.00]
<code>mssql_regcase</code>	308, [3.00]	<code>mysql_drop_db</code>	310, [3.00]
<code>mssql_result</code>	308, [3.00]	<code>mysql_errno</code>	310, [3.00]
<code>mssql_select_db</code>	308, [3.00]	<code>mysql_error</code>	311, [3.00]
<code>mssql_selectdb</code> (<i>mssql_select_db</i>)		<code>mysql_fetch_array</code>	311, [3.00]
<code>mssql_tablename</code>	308, [3.00]	<code>mysql_fetch_field</code>	311, [3.00]
<code>mssql_affected_rows</code>	296, [3.06]	<code>mysql_fetch_lengths</code>	312, [3.00]
<code>mssql_close</code>	296, [3.00]	<code>mysql_fetch_object</code>	312, [3.00]
<code>mssql_connect</code>	296, [3.00]	<code>mysql_fetch_row</code>	313, [3.00]
<code>mssql_data_seek</code>	297, [3.00]	<code>mysql_field_flags</code>	314, [3.00]
<code>mssql_fetch_array</code>	297, [3.00]	<code>mysql_field_len</code>	315, [3.00]
<code>mssql_fetch_field</code>	297, [3.00]	<code>mysql_field_name</code>	313, [3.00]
<code>mssql_fetch_object</code>	298, [3.00]	<code>mysql_field_seek</code>	313, [3.00]
<code>mssql_fetch_row</code>	298, [3.00]	<code>mysql_field_table</code>	313, [3.00]
<code>mssql_field_seek</code>	298, [3.00]	<code>mysql_field_type</code>	313, [3.00]
<code>mssql_free_result</code>	299, [3.00]	<code>mysql_free_result</code>	315, [3.00]
<code>mssql_num_fields</code>	299, [3.00]	<code>mysql_insert_id</code>	315, [3.00]
<code>mssql_num_rows</code>	299, [3.00]	<code>mysql_list_dbs</code>	316, [3.00]
<code>mssql_pconnect</code>	299, [3.00]	<code>mysql_list_fields</code>	315, [3.00]
<code>mssql_query</code>	300, [3.00]	<code>mysql_list_tables</code>	316, [3.00]
<code>mssql_result</code>	300, [3.00]	<code>mysql_num_fields</code>	316, [3.00]
<code>mssql_select_db</code>	300, [3.00]	<code>mysql_num_rows</code>	316, [3.00]
<code>mt_getrandmax</code>	156, [3.00]	<code>mysql_pconnect</code>	316, [3.00]
<code>mt_rand</code>	155, [3.00]	<code>mysql_query</code>	317, [3.00]
<code>mt_srand</code>	156, [3.00]	<code>mysql_result</code>	317, [3.00]
<code>mysql_affected_rows</code>	309, [3.00]	<code>mysql_select_db</code>	317, [3.00]
<code>mysql_close</code>	309, [3.00]	<code>mysql_tablename</code>	317, [3.00]
<code>mysql_connect</code>	309, [3.00]		
<code>mysql_change_user</code> ([3.13][†4])		N	
<code>mysql_create_db</code>	310, [3.00]	<code>name</code> (<i>domxml_name</i>)	
<code>mysql_data_seek</code>	310, [3.00]	<code>natcasesort</code> ([4.00])	
		<code>natsort</code> ([4.00])	

<code>newchild</code> (<i>domxml_newchild</i>)	<code>ociloadlob</code> (→ CD, [4b4])
<code>new_xml doc</code> (<i>domxml_newxml doc</i>)	<code>ocilogoff</code> 327, [3.04]
<code>next</code> 103, [3.00]	<code>ocilogon</code> 326, [3.04]
<code>n12br</code> 121, [3.00]	<code>ocinewcursor</code> (→ CD, [3.07])
<code>node</code> (<i>domxml_node</i>)	<code>ocinewdescriptor</code> (→ CD, [3.07])
<code>number_format</code> 156, [3.00]	<code>ocinlogon</code> (→ CD, [3.07])
O	<code>ocinumcols</code> (→ CD, [3.04])
<code>ob_end_clean</code> 347, [4b0]	<code>ocinumrows</code> (→ CD, [3.04])
<code>ob_end_flush</code> 347, [4b0]	<code>ociparse</code> (→ CD, [3.04])
<code>ob_get_contents</code> 347, [4b0]	<code>ociplugon</code> (→ CD, [3.07])
<code>ob_implicit_flush</code> 348, [4b4]	<code>ociresult</code> (→ CD, [3.04])
<code>ob_start</code> 348, [4b0]	<code>ocirollback</code> (→ CD, [3.07])
<code>ocibindbyname</code> 325, [3.04]	<code>ocirowcount</code> (→ CD, [3.07])
<code>ocicolumnisnull</code> 328, [3.04]	<code>ocisavedesc</code> (→ CD, [3.13])
<code>ocicolumnname</code> (→ CD, [3.04])	<code>ocisavelob</code> (→ CD, [4b4])
<code>ocicolumnprecision</code> (→ CD, [4R1])	<code>ocisavelobfile</code> (→ CD, [4b4])
<code>ocicolumnscale</code> (→ CD, [4R1])	<code>ociserverversion</code> (→ CD, [3.04])
<code>ocicolumnsize</code> 329, [3.04]	<code>ocisetprefetch</code> (→ CD, [3.13])
<code>ocicolumntype</code> (→ CD, [3.04])	<code>ocistatementtype</code> (→ CD, [3.04])
<code>ocicolumntyperaw</code> (→ CD, [4R1])	<code>ociwritelobtofile</code> (→ CD, [4b4])
<code>ocicommit</code> 327, [3.07]	<code>octdec</code> 157, [3.00]
<code>ocidefinebyname</code> 325, [3.07]	<code>odbc_autocommit</code> 318, [3.06]
<code>ocierror</code> (→ CD, [3.07])	<code>odbc_binmode</code> 318, [3.06]
<code>ociexecute</code> 327, [3.04]	<code>odbc_close</code> 319, [3.06]
<code>ocifetch</code> 328, [3.04]	<code>odbc_close_all</code> 319, [3.06]
<code>ocifetchinto</code> 328, [3.04]	<code>odbc_columnprivileges</code> ([4b2])
<code>ocifetchstatement</code> (→ CD, [3.07])	<code>odbc_columns</code> ([4b2])
<code>ocifreecursor</code> (<i>ocifreestatement</i>)	<code>odbc_commit</code> 319, [3.06]
<code>ocifreedesc</code> (→ CD, [4R1])	<code>odbc_connect</code> 319, [3.06]
<code>ocifreestatement</code> (→ CD, [3.05])	<code>odbc_cursor</code> 320, [3.06]
<code>ociinternaldebug</code> (→ CD, [3.04])	<code>odbc_do</code> 320, [3.06]

odbc_define ([4b2])	ora_columnname..... 330, [3.00]
odbc_exec..... 320, [3.06]	ora_columntype..... 330, [3.00]
odbc_execute..... 320, [3.06]	ora_commit..... 330, [3.00]
odbc_fetch_into..... 321, [3.06]	ora_commitoff..... 330, [3.00]
odbc_fetch_row..... 321, [3.06]	ora_commiton..... 330, [3.00]
odbc_field_len ([3.06])	ora_error..... 331, [3.00]
odbc_field_name..... 321, [3.06]	ora_errorcode..... 331, [3.00]
odbc_field_type..... 321, [3.06]	ora_exec..... 331, [3.00]
odbc_foreignkeys ([4b4])	ora_fetch..... 331, [3.00]
odbc_free_result..... 322, [3.06]	ora_getcolumn..... 331, [3.00]
odbc_gettypeinfo ([4b4])	ora_logoff..... 331, [3.00]
odbc_longreadlen..... 322, [3.06]	ora_logon..... 331, [3.00]
odbc_num_fields..... 322, [3.06]	ora_open..... 332, [3.00]
odbc_num_rows..... 323, [3.06]	ora_parse..... 332, [3.00]
odbc_pconnect..... 323, [3.06]	ora_rollback..... 332, [3.00]
odbc_prepare..... 323, [3.06]	ord..... 122, [3.00]
odbc_primarykeys ([4b4])	P
odbc_procedurecolumns ([4b4])	pack..... 349, [3.00]
odbc_procedures ([4b4])	parent (<i>domxml_parent</i>)
odbc_result..... 323, [3.06]	parse_ini_file ([4R1])
odbc_result_all..... 324, [3.06]	parse_str..... 122, [3.00]
odbc_rollback..... 324, [3.06]	parse_url..... 92, [3.00]
odbc_setoption..... 324, [3.06]	passthru..... 355, [3.00]
odbc_specialcolumns ([4b4])	pclose..... 174, [3.00]
odbc_statistics ([4b4])	pdf_add_annotation (→ CD, [3.12])
odbc_tableprivileges ([4b4])	pdf_add_outline..... 217, [3.06]
odbc_tables ([4b4])	pdf_add_pdflink ([3.12])
opendir..... 162, [3.00]	pdf_add_weblink ([3.12])
openlog..... 356, [3.00]	pdf_arc..... 214, [3.06]
ora_bind..... 329, [3.00]	pdf_begin_page..... 208, [3.06]
ora_close..... 329, [3.00]	pdf_circle..... 214, [3.06]

pdf_clip216, [3.06]	pdf_scale212, [3.06]
pdf_close208, [3.06]	pdf_set_char_spacing210, [3.06]
pdf_close_image218, [3.07]	pdf_set_duration218, [3.06]
pdf_closepath214, [3.06]	pdf_set_font209, [3.06]
pdf_closepath_fill_stroke215, [3.06]	pdf_set_horiz_scaling210, [3.06]
pdf_closepath_stroke215, [3.06]	pdf_set_info_author207, [3.06]
pdf_continue_text211, [3.06]	pdf_set_info_creator206, [3.06]
pdf_curveto213, [3.06]	pdf_set_info_keywords207, [3.06]
pdf_end_page208, [3.06]	pdf_set_info_subject207, [3.06]
pdf_endpath215, [3.06]	pdf_set_info_title206, [3.06]
pdf_execute_image219, [3.07]	pdf_set_leading209, [3.06]
pdf_fill215, [3.06]	pdf_set_parameter (→ CD, [4RC1])
pdf_fill_stroke215, [3.06]	pdf_set_text_matrix210, [3.06]
pdf_get_font ([3.06])	pdf_set_text_pos210, [3.06]
pdf_get_fontname ([3.06])	pdf_set_text_rendering209, [3.06]
pdf_get_fontsize ([3.06])	pdf_set_text_rise210, [3.06]
pdf_get_info206, [3.06]	pdf_set_transition217, [3.06]
pdf_get_image_height ([3.12])	pdf_set_word_spacing211, [3.06]
pdf_get_image_width ([3.12])	pdf_setdash213, [3.06]
pdf_lineto213, [3.06]	pdf_setflat212, [3.06]
pdf_moveto213, [3.06]	pdf_setgray216, [3.06]
pdf_open207, [3.06]	pdf_setgray_fill216, [3.06]
pdf_open_gif218, [3.07]	pdf_setgray_stroke216, [3.06]
pdf_open_jpeg218, [3.07]	pdf_setlinecap213, [3.06]
pdf_open_memory_image218, [3.10]	pdf_setlinejoin212, [3.06]
pdf_place_image218, [3.07]	pdf_setlinewidth213, [3.06]
pdf_put_image219, [3.07]	pdf_setmiterlimit213, [3.06]
pdf_rect214, [3.06]	pdf_setrgbcolor217, [3.06]
pdf_restore211, [3.06]	pdf_setrgbcolor_fill216, [3.06]
pdf_rotate212, [3.06]	pdf_setrgbcolor_stroke216, [3.06]
pdf_save211, [3.06]	pdf_show208, [3.06]

pdf_show_boxed (→ CD, [4RC1])	pg_numfields..... 340, [3.00]
pdf_show_xy..... 208, [3.06]	pg_numrows..... 340, [3.00]
pdf_stringwidth..... 211, [3.06]	pg_options..... 340, [3.00]
pdf_stroke..... 214, [3.06]	pg_pconnect..... 340, [3.00]
pdf_translate..... 211, [3.06]	pg_port..... 341, [3.00]
pfsockopen..... 183, [3.07]	pg_result..... 341, [3.00]
pg_close..... 333, [3.00]	pg_tty..... 341, [3.00]
pg_cmdtuples..... 333, [3.00]	phpinfo..... 362, [3.00]
pg_connect..... 333, [3.00]	phpversion..... 362, [3.00]
pg_dbname..... 334, [3.00]	pi..... 157, [3.00]
pg_errormessage..... 334, [3.00]	popen..... 174, [3.00]
pg_exec..... 334, [3.00]	pos..... 104, [3.00]
pg_fetch_array..... 334, [3.00]	posix_ctermid ([3.13])
pg_fetch_object..... 335, [3.00]	posix_getcwd ([3.13])
pg_fetch_row..... 336, [3.00]	posix_getegid ([3.10])
pg_fieldisnull..... 337, [3.00]	posix_geteuid ([3.10])
pg_fieldname..... 337, [3.00]	posix_getgrgid ([3.10])
pg_fieldnum..... 337, [3.00]	posix_getgrnam ([3.13])
pg_fieldprtlen..... 337, [3.00]	posix_getgroups ([3.10])
pg_fieldsize..... 337, [3.00]	posix_getlogin ([3.13])
pg_fieldtype..... 338, [3.00]	posix_getpgid ([3.10])
pg_freeresult..... 338, [3.00]	posix_getpgrp ([3.10])
pg_getlastoid..... 338, [3.00]	posix_getpid ([3.10])
pg_host..... 338, [3.00]	posix_getppid ([3.10])
pg_loclose..... 338, [3.00]	posix_getpwnam ([3.13])
pg_locreate..... 338, [3.00]	posix_getpwuid ([3.13])
pg_loopen..... 339, [3.00]	posix_getrlimit ([3.10])
pg_loread..... 339, [3.00]	posix_getsid ([3.10])
pg_loreadall..... 339, [3.00]	posix_getuid ([3.10])
pg_lounlink..... 339, [3.00]	posix_isatty ([3.13])
pg_lowrite..... 340, [3.00]	posix_kill ([3.13])

posix_mkfifo ([3.13])	rawurlencode123, [3.00]
posix_setgid ([3.13])	read (<i>readdir</i>)
posix_setpgid ([3.13])	readdir162, [3.00]
posix_setsid ([3.13])	readfile175, [3.00]
posix_setuid ([3.13])	readgzfile, [3.00]
posix_times ([3.13])	readline ([4b4])
posix_ttyname ([3.13])	readline_add_history ([4b4])
posix_uname ([3.13])	readline_clear_history ([4b4])
pow157, [3.00]	readline_completion_function ([4b4])
preg_grep140, [3.09]	readline_info ([4b4])
preg_match138, [3.09]	readline_list_history ([4b4])
preg_match_all138, [3.09]	readline_read_history ([4b4])
preg_quote140, [3.09]	readline_write_history ([4b4])
preg_replace139, [3.09]	readlink175, [3.00]
preg_split140, [3.09]	realpath ([4b4])
prev104, [3.00]	recode (<i>recode_string</i>)
print122, [3.00]	recode_file ([4.00])
printf122, [3.00]	recode_string ([4.00])
print_r363, [4b1]	register_shutdown_function ...357, [3.04]
pspell_new133, [4.02]	rename175, [3.00]
pspell_check134, [4.02]	require84, [3.00]
pspell_suggest134, [4.02]	reset104, [3.00]
putenv362, [3.00]	return84, [3.00]
Q	rewind175, [3.00]
quoted_printable_decode123, [3.06]	rewinddir162, [3.00]
quotemeta123, [3.00]	rmdir175, [3.00]
R	root (<i>domxml_root</i>)
rad2deg157, [3.04]	round157, [3.00]
rand157, [3.00]	rsort104, [3.00]
range104, [3.08]	rtrim (<i>chop</i>)
rawurldecode123, [3.00]	

S

- save (*ocisavelob*)
- savefile (*ocisavelobfile*)
- sem_acquire ([3.06])
- sem_get ([3.06])
- sem_release ([3.06])
- serialize..... 350, [3.05]
- session_decode..... 179, [4b1]
- session_destroy..... 179, [4b1]
- session_encode..... 179, [4b1]
- session_get_cookie_params ([4b1])
- session_id..... 180, [4b1]
- session_is_registered..... 180, [4b1]
- session_module_name..... 180, [4b1]
- session_name..... 180, [4b1]
- session_register..... 180, [4b1]
- session_save_path..... 181, [4b1]
- session_set_cookie_params ([4b4])
- session_set_save_handler..... 181, [4b4]
- session_start..... 181, [4b4]
- session_unregister..... 181, [4b4]
- session_unset ([4b4])
- set_attr (*domxml_setattr*)
- set_socket_blocking..... 184, [3.00]
- set_file_buffer..... 181, [3.08]
- set_magic_quotes_runtime..... 362, [3.00]
- set_time_limit..... 363, [3.00]
- setcookie..... 93, [3.00]
- setlocale..... 124, [3.00]
- settype..... 97, [3.00]
- shell_exec (→ CD, [4b1])
- shm_attach ([3.06])
- shm_detach ([3.06])
- shm_get_var ([3.06])
- shm_put_var ([3.06])
- shm_remove ([3.06])
- shm_remove_var ([3.06])
- shuffle..... 105, [3.08]
- similar_text..... 126, [3.07]
- sin..... 158, [3.00]
- sizeof..... 105, [3.00]
- sleep..... 357, [3.00]
- snmp_get_quick_print ([3.08])
- snmp_set_quick_print ([3.08])
- snmpget ([3.00])
- snmpset ([3.08])
- snmpwalk ([3.08])
- snmpwalkoid ([3.08])
- socket_get_status ([4b4])
- socket_set_blocking
([4b4],
set_socket_blocking)
- socket_set_timeout ([4b4])
- solid_close ([3.00])
- solid_connect ([3.00])
- solid_exec ([3.00])
- solid_fetchrow ([3.00])
- solid_fieldname ([3.00])
- solid_fieldnum ([3.00])
- solid_freeresult ([3.00])
- solid_numfields ([3.00])
- solid_numrows ([3.00])
- solid_result ([3.00])

sort	105, [3.00]	strtoupper	131, [3.00]
soundex	126, [3.00]	strtr	131, [3.00]
split	137, [3.00]	strval	98, [3.00]
sprintf	127, [3.00]	substr	132, [3.00]
sql_regcase	137, [3.00]	substr_count ([4.00])	
sqrt	158, [3.00]	substr_replace ([4b4])	
srand	158, [3.00]	swf_openfile	220, [4.01]
stat	176, [3.00]	swf_closefile	220, [4.01]
str_repeat ([4b4])		swf_labelframe	220, [4.01]
str_replace	131, [3.00]	swf_showframe	220, [4.01]
strchr	128, [3.00]	swf_setframe	220, [4.01]
strcasecmp ([3.03])		swf_getframe	220, [4.01]
strcmp	128, [3.00]	swf_mulcolor	221, [4.01]
strcspn	128, [3.00]	swf_addcolor	221, [4.01]
strftime	144, [3.00]	swf_placeobject	221, [4.01]
strip_tags	128, [3.00]	swf_modifyobject	221, [4.01]
stripslashes ([4b4])		swf_removeobject	221, [4.01]
stripslashes	129, [3.00]	swf_nextid	222, [4.01]
strlen	129, [3.00]	swf_startdoaction	222, [4.01]
strnatcasecmp ([4.00])		swf_enddoaction	222, [4.01]
strnatcmp ([4.00])		swf_actiongotoframe	222, [4.01]
strncmp ([4.00])		swf_actiongeturl	222, [4.01]
strpos	129, [3.00]	swf_actionnextframe	222, [4.01]
strrchr	129, [3.00]	swf_actionprevframe	222, [4.01]
strrev	130, [3.00]	swf_actionplay	223, [4.01]
strrpos	129, [3.00]	swf_actionstop	223, [4.01]
strspn	130, [3.00]	swf_actiontogglequality	223, [4.01]
strstr	130, [3.00]	swf_actionwaitforframe	223, [4.01]
strtok	130, [3.00]	swf_actionsettarget	223, [4.01]
strtolower	131, [3.00]	swf_actiongotolabel	223, [4.01]
strtotime	147 [3.12] [4b2]	swf_defineline	223, [4.01]

swf_definerect.....	224, [4.01]	swf_perspective.....	230, [4.01]
swf_startshape.....	224, [4.01]	swf_polarview.....	231, [4.01]
swf_shapelinesolid.....	224, [4.01]	swf_lookat.....	231, [4.01]
swf_shapefilloff.....	224, [4.01]	swf_pushmatrix.....	231, [4.01]
swf_shapefillsolid.....	224, [4.01]	swf_popmatrix.....	231, [4.01]
swf_shapefillbitmapclip.....	225, [4.01]	swf_scale.....	231, [4.01]
swf_shapefillbitmaptile.....	225, [4.01]	swf_translate.....	231, [4.01]
swf_shapemoveto.....	225, [4.01]	swf_rotate.....	232, [4.01]
swf_shapelineto.....	225, [4.01]	swf_posround.....	232, [4.01]
swf_shapecurveto.....	225, [4.01]	switch	83, [3.00]
swf_shapecurveto3.....	225, [4.01]	sybase_affected_rows.....	341, [3.00]
swf_shapearc.....	226, [4.01]	sybase_close.....	342, [3.00]
swf_endshape.....	226, [4.01]	sybase_connect.....	342, [3.00]
swf_definefont.....	226, [4.01]	sybase_data_seek.....	342, [3.00]
swf_setfont.....	226, [4.01]	sybase_fetch_array.....	343, [3.00]
swf_fontsize.....	226, [4.01]	sybase_fetch_field.....	343, [3.00]
swf_fontslant.....	226, [4.01]	sybase_fetch_object.....	343, [3.00]
swf_fonttracking.....	226, [4.01]	sybase_fetch_row.....	344, [3.00]
swf_getfontinfo.....	227, [4.01]	sybase_field_seek.....	344, [3.00]
swf_definetext.....	227, [4.01]	sybase_free_result.....	344, [3.00]
swf_textwidth.....	227, [4.01]	sybase_num_fields.....	344, [3.00]
swf_definebitmap.....	227, [4.01]	sybase_num_rows.....	345, [3.00]
swf_getbitmapinfo.....	227, [4.01]	sybase_pconnect.....	345, [3.00]
swf_startsymbol.....	228, [4.01]	sybase_query.....	345, [3.00]
swf_endsymbol.....	228, [4.01]	sybase_result.....	345, [3.00]
swf_startbutton.....	228, [4.01]	sybase_select_db.....	346, [3.00]
swf_addbuttonrecord.....	228, [4.01]	symlink.....	177, [3.00]
swf_oncondition.....	229, [4.01]	syslog.....	356, [3.00]
swf_endbutton.....	230, [4.01]	system.....	354, [3.00]
swf_viewport.....	230, [4.01]	T	
swf_ortho2.....	230, [4.01]	tan.....	158, [3.00]

- tempnam177, [3.00]
- time148, [3.00]
- tmpfile ([3.13][4b4])
- touch177, [3.00]
- trim132, [3.00]
- U**
- uasort106, [3.04]
- ucfirst132, [3.00]
- ucwords132, [3.03]
- uksort106, [3.04]
- umask177, [3.00]
- uniqid353, [3.00]
- unixtojd ([4.00])
- unlink178, [3.00]
- unpack350, [3.00]
- user_error ([4.00])
- unserialize351, [3.05]
- unset98, [3.00]
- urldecode92, [3.00]
- urlencode92, [3.00]
- usleep357, [3.00]
- usort106, [3.04]
- utf8_decode205, [3.06]
- utf8_encode205, [3.06]
- V**
- var_dump363, [3.05]
- virtual352, [3.00]
- vm_addalias ([3.00])
- vm_adduser ([3.00])
- vm_delalias ([3.00])
- vm_deluser ([3.00])
- vm_passwd ([3.00])
- W**
- wddx_add_vars196, [3.07]
- wddx_deserialize196, [3.07]
- wddx_packet_end196, [3.07]
- wddx_packet_start196, [3.07]
- wddx_serialize_value195, [3.07]
- wddx_serialize_vars195, [3.07]
- writetofile (*ociwritelobtofile*)
- X**
- xmlDoc ([4b4])
- xmlDocfile ([4b4])
- xmltree ([4b4])
- xml_error_string204, [3.06]
- xml_get_current_byte_index ...204, [3.06]
- xml_get_current_column_number 204, [3.06]
- xml_get_current_line_number ..204, [3.06]
- xml_get_error_code204, [3.06]
- xml_parse203, [3.06]
- xml_parser_create198, [3.06]
- xml_parser_free204, [3.06]
- xml_parser_get_option205, [3.06]
- xml_parser_set_option205, [3.06]
- xml_set_character_data_handler
.....199, [3.06]
- xml_set_default_handler200, [3.06]
- xml_set_element_handler198, [3.06]
- xml_set_external_entity_ref_handler
.....202, [3.06]
- xml_set_notation_decl_handler 202, [3.06]
- xml_set_processing_instruction_handle
r
.....200, [3.06]

xml_set_unparsed_entity_decl_handler
..... 201, [3.06]

Y

yp_err_string ([3.07])
yp_errno ([3.07])
yp_first ([3.07])
yp_get_default_domain ([3.07])
yp_master ([3.07])
yp_match ([3.07])
yp_next ([3.07])
yp_order ([3.07])

Z

zend_logo_guid ([4b4])
zend_version ([4.01])

A.6 Befehlsreferenz

Als Befehle werden die fest in PHP eingebauten Kommandos und Kontrollstrukturen bezeichnet.

Kontrollstrukturen

if...else...elseif

**if
else
elseif**

Mit `if` wird eine Programmverzweigung eingeleitet, `else` leitet optional den alternativen Zweig (Nein-Zweig) ein und `elseif` setzt die Struktur mit einer Unterabfrage fort.

- ◆ `if (bedingung) JA-Befehl;`
- ◆ `if (bedingung) { JA-Befehle; JA-Befehle }`
- ◆ `if (bedingung) {
 JA-Befehle;
} else {
 NEIN-Befehle;
}`
- ◆ `if (bedingung) {
 JA-Befehle;
} else {
 NEIN-Befehle;
} elseif (weitere_bedingung) {
 JA-Bedingung im NEIN-Zweig;
} else
 NEIN-Bedingung im NEIN-Zweig;
}`

Alternative Syntax

`if` erlaubt eine alternative Syntax, die das Umschließen von HTML-Sequenzen erleichtert:

- ◆ `if (bedingung):
 JA-Befehle;
elseif (nein_bedingung):
 JA-im-NEIN-Zweig;
else:
 NEIN-Befehle;
endif;`

Bedingungen müssen einen logischen Wert (TRUE oder FALSE) zurückgeben. Alle Ausdrücke geben einen Wert zurück, wobei die Zahl 0, die Zeichenketten "0" und "" als FALSE gewertet werden, alles andere als TRUE. Zuweisungen geben den Wert der Zuweisung zurück, `$var=0` ist also FALSE, während `$var=1` TRUE ist.

while

Mit `while` werden Schleifen erzeugt. Die Schleife wird durchlaufen, solange die Bedingung `TRUE` ist.

while

```
while (bedingung) {
    Befehle, solange Bedingung Wahr ist;
}
```



PHP erlaubt eine alternative Syntax, die das Umschließen von HTML-Sequenzen erleichtert:

Alternative Syntax

```
while (bedingung):
    Befehle, solange Bedingung Wahr ist;
endwhile;
```



Ein vorzeitiges Verlassen der Schleife ist mit dem Befehl `break` möglich, der notwendigerweise von einem `if`-Befehl ausgelöst wird.

do ... while

Bei der `do...while`-Schleife wird die Bedingung am Ende geprüft. Unabhängig von dem Ergebnis wird die Schleife also mindestens einmal durchlaufen.

do ... while

```
do {
    Befehle-solange-Bedingung-Wahr;
} while (bedingung);
```

**for**

Mit `for`-Schleifen werden abzählbare Durchläufe gesteuert. Die Anwendung ist langsamer als `do`- oder `while`-Schleifen. Sie sollten `for` deshalb nur einsetzen, wenn die gewünschte Konstruktion mit anderen Mitteln nicht darstellbar ist.

for

```
for (startanweisung; bedingung; prüfanweisung) {
    Befehle-solange-Bedingung-wahr;
}
```



Beim Eintritt in die Schleife wird die *startanweisung* ausgeführt. Hier können auch mehrere, durch Kommata getrennte Anweisungen stehen. Normalerweise wird hier eine Schleifenvariable initialisiert. Bei jedem Durchlauf wird zu Beginn die *bedingung* geprüft. Am Ende der Schleife wird die *prüfanweisung* ausgeführt. Auch hier können mehrere durch Kommata getrennte Befehle stehen. Wenn die Bedingung `TRUE` zurückgibt, wird die Schleife erneut

durchlaufen. Die Prüfanweisung wird oft genutzt, um eine Schrittweite vorzugeben.



```
for ($i = 0; $i <= 100; $i++) {
    echo "Durchlauf: $i<BR>";
}
```

foreach

foreach

Diese Schleifenanweisung durchläuft ein Array oder ein assoziatives Array. Bei jedem Durchlauf wird den Laufvariablen der Wert des aktuellen Elements zugewiesen.

Im ersten Fall wird der Variablen *\$value* mit jedem Durchlauf jeweils ein Element des Arrays *\$array* zugewiesen:

```
◆ foreach($array as $value) {
    Anweisungsblock
}
```

Im zweiten Fall wird ein assoziatives Array erwartet. Die Schlüssel werden der Variablen *\$key*, die Werte *\$value* zugewiesen.

```
◆ foreach($array as $key => $value) {
    Anweisungsblock
}
```

Die Operatoren `as` und `=>` sind Bestandteil dieses Befehls.

Siehe auch die Funktionen `list`, `each` und `array`.

break

break

Mit `break` werden andere Kontrollstrukturen unmittelbar verlassen. Die Anwendung ist nur in `if`-Anweisungen sinnvoll.

```
◆ while ($i < 25) {
    ...
    if ($i == $x) { break; }
    ...
}
```

In folgenden Anweisungen kann `break` eingesetzt werden:

- while
- do
- for
- switch

continue

Mit diesem Befehl wird die Programmausführung am Anfang der umgebenden Schleife fortgesetzt. Die Anwendung ist nur in if-Anweisungen sinnvoll.

```
while ($i < 25) {  
    ...  
    if ($i == $x) { continue; }  
    ...  
}
```

continue



In folgenden Anweisungen kann `continue` eingesetzt werden:

- `while`
- `do`
- `for`
- `switch`

switch

Längere Vergleiche mit `if...elseif` können unübersichtlich und fehleranfällig sein. In solchen Fällen hilft `switch`. Der Befehl prüft eine Variable auf Gleichheit mit einem Kontrollwert.

```
switch (variable) {  
    case kontrollwert1:  
        Befehle-wenn-Wert-gleich;  
        break;  
    case kontrollwert1:  
        Befehle-wenn-Wert-gleich;  
        break;  
    default:  
        Befehle-wenn-kein-Wert-gleich;  
}
```

switch



Jeder Zweig, der angesprochen wird, sollte mit einem `break`-Befehl beendet werden, ansonsten wird die Ausführung im nächsten `case`-Zweig fortgesetzt und dabei auch der `default`-Zweig ausgeführt. Ansonsten wird `default` nur ausgeführt, wenn keine der Bedingungen übereinstimmt.

require

require Mit diesem Befehl wird der Inhalt einer Datei geladen. Die eingeschlossene Datei verhält sich so, als wäre der enthaltene Text direkt geschrieben worden.

- ◆ `require ("pfad/dateiname");`
- ◆ `require ($datei);`

Sie können diesen Befehl nicht innerhalb von Schleifen anwenden.

include

include `include` lädt eine Datei und führt diese separat aus. Die Anwendung in Schleifen ist deshalb zulässig.

- ◆ `include ("pfad/dateiname");`
- ◆ `include ($datei);`

Sie müssen hinter Bedingungen (`if`, `for` usw.) Blockkennzeichen setzen, damit der gesamte Inhalt der eingeschlossenen Datei umschlossen wird.

function

function Mit dieser Anweisung wird eine benutzerdefinierte Funktion eingeleitet.

- ◆

```
function ([$variable [ = "Standardwert"], ... weitere]) {
    Befehle der Funktion;
}
```

Die Parameter sind optional.

return

return Diese Anweisung legt den Rückgabewert einer benutzerdefinierten Funktion fest.

- ◆

```
function ([$variable [ = "Standardwert"], ... weitere]) {
    Befehle der Funktion;
    return $rueckgabewert;
}
```

class

class Mit `class` wird eine Klassendefinition eingeleitet. Die Klasse umschließt Variablen und Funktionen. Variablen werden als Eigenschaften, Funktionen als Methoden der Klasse abgebildet.

```

class NamederKlasse {
    var $eigenschaft;

    function Methode([Parameter])
    {
        Definition der Methode;
    }
}

```



Instanzen der Klasse werden mit `new` erzeugt (siehe nächsten Eintrag). Um innerhalb der Definition auf Methoden und Eigenschaften der Klasse zugreifen zu können, wird die Pseudovariablen `$this` benutzt:

`$this->var`

```
$this->methode(Parameter);
```

Eigenschaften werden als Variable mit dem Schlüsselwort `var` gebildet:

`var`

```
var $eigenschaft
```

Im Gegensatz zu PHP3 darf diese Konstruktion in PHP4 nicht zugleich mit einer Zuweisung kombiniert werden. Nehmen Sie diese Zuweisung im Konstruktor vor.

PHP3 kennt Konstruktoren. Ein Konstruktor entsteht, wenn eine Methode denselben Namen wie die Klasse hat. Im Gegensatz dazu gibt es keine Destruktoren, da Objekte Ihre Gültigkeit am Ende des Skripts verlieren und damit eine Destruktion nicht zur Laufzeit des Skripts sinnvoll ist.

Konstruktoren

PHP4 kennt als Operator für den Aufruf der Methoden außerdem `::` anstatt `->`. Weitere Erläuterungen zu Klassen und Objekten finden Sie im Buch in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.** ab Seite **Fehler! Textmarke nicht definiert..**

new

`new` erzeugt eine Instanz einer Klasse, also ein neues Objekt. Auf dieses Objekt können dann die Eigenschaften und Methoden der Klasse angewendet werden.

`new`

```
$mein_objekt = new Klassenname;
```



extends

extends Um Klassen zu vererben, fehlt PHP die vollständige Funktionalität. Um einfache Vererbungen innerhalb der Definition zu ermöglichen, wird die Anweisung `extends` verwendet.

◆

```
class TochterKlasse extends MutterKlasse {
    Definitionen;
}
```

Umgang mit Klassen

Die folgenden Funktionen unterstützen den Umgang mit Klassen.

call_user_method

call_user_method Diese Funktion ruft die Methode einer benutzerdefinierten Klasse auf.

◆

```
mixed call_user_method(string method_name, object obj
    [, mixed parameter [, mixed ...]])
```

Als Parameter wird der Name der Methode in einer Zeichenkette übergeben. Das folgende Beispiel zeigt den konventionellen Aufruf und anschließend die Nutzung der Funktion `call_user_method`:

```
$centry->print_info();
call_user_method ("print_info", $centry, "\t");
```

get_class

get_class Diese Funktion ermittelt den Namen der Klasse, in der das Objekt *obj* definiert wurde.

◆

```
string get_class(object obj)
```

get_parent_class

get_parent_class Diese Funktion gibt den Namen der Elternklasse zurück, von der die Klasse abgeleitet wurde, zu der das Objekt *obj* gehört.

◆

```
string get_parent_class(object obj)
```

get_class_methods

get_class_methods Diese Funktion gibt ein Array mit den Namen der Methoden zurück, die in der angegebenen Klasse definiert wurden.

◆

```
array get_class_methods(string class_name)
```

get_class_vars

Diese Funktion gibt ein Array mit den Namen der Eigenschaften zurück, die in der angegebenen Klasse definiert wurden.

array `get_class_vars`(string `class_name`)

get_class_vars**get_object_vars**

Die Funktion gibt ein assoziatives Array zurück, das die Objekt-Eigenschaften eines spezifischen Objekts enthält.

array `get_object_vars`(object `obj`)

get_object_vars**is_subclass_of**

Die Funktion ermittelt, ob ein Objekt einer Klasse entstammt, die selbst Tochter der Klasse *superclass* ist.

bool `is_subclass_of`(object `obj`, string `superclass`)

is_subclass_of**class_exists**

Mit `class_exists` wird überprüft, ob eine bestimmte Klasse definiert ist.

bool `class_exists`(string `class_name`)

class_exists**method_exists**

Die Funktion gibt TRUE zurück, wenn die angegebene Methode in dem Objekt *obj* definiert ist.

bool `method_exists`(object `object`, string `method_name`)

method_exists**get_declared_classes**

Diese Funktion gibt ein Array zurück, das die Namen aller deklarierten Klassen enthält.

array `get_declared_classes`(void)

get_declared_classes

Ab PHP 4.0.1pl2, werden drei eingebaute Klassen am Anfang des Arrays zurückgegeben: `stdClass`, `OverloadedTestClass` und `Directory`.

com**com**

com ist eine Pseudoklasse, mit der eine Instanz eines COM-Objekts gebildet werden kann. COM ist nur unter Windows verfügbar. Es gibt mehrere Funktionen, die verwendet werden können. Das folgende Beispiel zeigt die Anwendung der Funktionen, indem das ADO verwendet wird, um direkt auf eine MS Access-Datenbank zuzugreifen:



```
<?php
$adoConn = new COM("ADODB.Connection");
echo "ADO Version: " . $adoConn->Version;
define(CSTR,
    "Driver={Microsoft Access Driver (*.mdb)};DBQ=Auction.mdb");
$intConn = $adoConn->Open(CSTR, 2);
if ($adoConn->State == 1) {
    echo "<br>Verbindung erfolgreich ge&ouml;ffnet:<br>";
    $strSQL = "SELECT count(*) as counter FROM address";
    $adoRS = $adoConn->Execute($strSQL, &$records);
    echo $adoConn->ConnectionString;
    echo "<br>" . $records . " Datens&auml;tze gefunden";
    echo "<br>Abfrage: " . $adoRS->Source;
    echo "<br>Status: " . $adoRS->State;
    echo "<br>Pos.: " . $adoRS->AbsolutePosition;
    echo "<br>" . $adoRS->fields->count . " Felder";
    echo "<br>" . $adoRS("counter") . " :";
}
?>
```

Diese Funktionsgruppe ist noch in der Entwicklung und funktioniert nicht zuverlässig.

com_load**com_load**

Lädt ein Modul von dem angegebenen COM-Server.



```
string com_load(string module name [, string server name])
```

com_invoke**com_invoke**

Führt eine Methode des angegebenen Objekts aus.



```
mixed com_invoke(resource object, string function_name
    [, mixed function parameters, ...])
```

com_propget**com_propget**

Liest eine Eigenschaft eines COM-Objekts.



```
mixed com_propget(resource object, string property)
```

com_get ist ein Alias für com_propget.

com_propput

Setzt eine Eigenschaft eines COM-Objekts.

mixed **com_propput**(resource object, string property)

com_propset und com_set sind Aliase für com_propput.

com_propput

Variable, Konstanten und Parameter

define

Diese Funktion definiert eine Konstante. Konstanten können zur Laufzeit nicht mehr geändert werden.

void **define**(string name, mixed value)

define

defined

Diese Funktion untersucht, ob eine Konstante definiert wurde, und gibt dann TRUE zurück.

bool **defined**(string name)

defined

func_get_arg

Die Funktion gibt den Wert des angegebenen Parameters zurück. Die Parameter zählen nullbasiert.

mixed **func_get_arg**(int arg_num)

func_get_arg

Diese Funktion kann nur innerhalb des Körpers einer benutzerdefinierten Funktion aufgerufen werden.

func_get_args

Die Funktion gibt ein Array mit den Parametern zurück, die übergeben wurden.

array **func_get_args**(void)

func_get_args

Diese Funktion kann nur innerhalb des Körpers einer benutzerdefinierten Funktion aufgerufen werden.

func_num_args

Gibt die Anzahl der Parameter zurück, die an eine nutzerdefinierte Funktion übergeben wurden, für die keine Parameter spezifiziert sind.

func_num_args

◆ `int func_num_args(void)`

Diese Funktion kann nur innerhalb des Körpers einer benutzerdefinierten Funktion aufgerufen werden.

global

global

Mit diesem Schlüsselwort werden globale Variable innerhalb einer Funktion sichtbar gemacht, was normalerweise nicht der Fall ist.



```
$a = 45;
$b = 23;
function MeineFunktion() {
    global $a, $b;
    ...
}
```

static

static

Variable innerhalb einer Funktion sind immer lokal. Sie können den Inhalt einer lokalen Variablen erhalten, indem das Schlüsselwort `static` benutzt wird.



```
$b = 23;
function MeineFunktion() {
    static $b;
    ...
}
```

\$

\$

In PHP gibt es variable Variablen. Eine Variable enthält dann keinen Wert, sondern den Namen einer anderen Variablen.



```
$a = "Hallo";
$$a = "Welt";

echo "$a ${$a}";
```

Das Beispiel gibt »Hallo Welt« aus.

Typkonvertierungen

(double)

(double)

Mit `(double)` wird der Inhalt einer Variablen im Fließkommaformat dargestellt.

◆ `$fkz = (double) $string;`

Alternativ kann auch `(real)` geschrieben werden.

(integer)

Mit (integer) wird der Inhalt einer Variablen im Integer-Format dargestellt.

```
$fkz = (integer) $string;
```

Alternativ kann auch (int) geschrieben werden.

(string)

Mit (string) wird der Inhalt einer Variablen als Zeichenkette dargestellt.

```
$str = (string) $zahl;
```

(array)

Mit (array) wird der Inhalt einer Variablen als Array dargestellt.

```
$fkz = (array) $string;
```

(object)

Mit (object) wird der Inhalt einer Variablen als Objekt dargestellt.

```
$obj = (object) $string;
```

Das resultierende Objekt enthält eine Eigenschaft, die den Namen der konvertierten Variablen trägt und folgendermaßen abgefragt werden kann:

```
echo $obj->string;
```

A.7 Funktionsreferenz

Die Funktionsreferenz zeigt nicht alle Funktionen, da dies den Rahmen des Buches sprengen würde. Außerdem sind viele Funktionen auf sehr spezielle Einsatzgebiete zugeschnitten. Zugleich wächst die Anzahl der Funktionen ständig. Wenn Sie eine Funktion vermissen, schreiben Sie an joerg@krause.net.

URL und HTML

parse_url

parse_url parse_url zerlegt ein URL und gibt die Komponenten "scheme", "host", "port", "user", "pass", "path", "query" und "fragment" zurück. Die Bestandteile werden Indizes eines assoziativen Arrays.

◆ `array parse_url(string url);`

urldecode

urldecode urldecode dekodiert eine Zeichenkette, die URL-kodiert war (mit %##-Zeichen).

◆ `string urldecode(string str);`

urlencode

urlencode urlencode kodiert eine Zeichenkette so, dass sie per URL übertragen werden kann.

◆ `string urlencode(string str);`

base64_encode

base64_encode base64_encode kodiert Daten nach dem MIME Base64-Verfahren.

◆ `string base64_encode(string data);`

Siehe auch base64_decode, chunk_split, RFC 2045 Sektion 6.8.

base64_decode

base64_decode base64_decode dekodiert Daten aus dem Base64-Format.

◆ `string base64_decode(string encoded_data);`

Die Funktion dekodiert Daten aus dem Base64-Format, das vor allem bei der Übertragung von Binärdateien per E-Mail zum Einsatz kommt.

Siehe auch `base64_encode`, RFC 2045 Sektion 6.8.

header

`header` sendet eine Zeile eines HTTP-Headers. HTTP ist in RFC 2068 spezifiziert. Header müssen vor jeder anderen Ausgabe im Skript erstellt und gesendet werden.

```
int header(string string);
```

Eine Weiterleitung auf eine andere Netzwerkadresse erreichen Sie mit der folgenden Anweisung:

```
header("Location: www.redirect.com");
```

Der Befehlssteil »Location: « weist den Browser an, die Umleitung durch Anforderung der angegebenen URL auszulösen.

header



Beispiel

headers_sent

Die Funktion gibt TRUE zurück, wenn die Header bereits gesendet wurden.

```
boolean headers_sent(void)
```

Mit dieser Funktion können Sie den lästigen Fehler umgehen, der erscheint, wenn nach dem Ausgeben irgendwelcher Zeichen noch versucht wird, die Header zu verändern:

```
if (!headers_sent) {
    header("Location: www.yoolia.de");
}
```

headers_sent



Beispiel

setcookie

`setcookie` sendet ein Cookie an den Browser. Cookies werden im Header übertragen, der Einsatz macht also nur Sinn, wenn noch keine Daten an den Browser übertragen wurden.

```
int setcookie(string name, string value, int expire,
               string [path], string [domain],
               string [secure]);
```

Das folgende Cookie ist die einfachste Anwendung der Funktion:

```
setcookie("TestCookie", "Test Value");
```

setcookie



Beispiel

Die Angabe einer Zeit erfolgt mit Hilfe des UNIX-Zeitstempels und einem Offset in Sekunden. Das folgende Cookie hat eine Lebensdauer von einer Stunde:



Beispiel

```
setcookie("TestCookie", $value, time()+3600);
```

Eine vollständige Definition mit Pfad und Domainangabe. Der Browser sendet dieses Cookie nur an die Domain zurück und auch nur dann, wenn die Pfadangabe übereinstimmt. Außerdem muss eine sichere Verbindung bestehen (`secure=1`):



Beispiel

```
setcookie("TestCookie", $value, time()+3600,  
        "/~rasmus/", ".utoronto.ca", 1);
```

Die Abfrage der Cookies kann in einem PHP-Skript wie folgt aussehen:

```
echo $TestCookie;  
echo $HTTP_COOKIE_VARS["TestCookie"];
```

Variablenfunktionen

call_user_func

call_user_func Diese Funktion ruft eine nutzerdefinierte Funktion auf, die Parameter.



```
mixed call_user_func(string function_name  
                    [, mixed parameter [, mixed ...]])
```

function_name ist der Name einer Funktion als Zeichenkette.

gettype

gettype gettype ermittelt den Typ einer Variablen *var*.



```
string gettype(mixed var);
```

Die Rückgabewerte können folgende sein:

- »integer«
- »double«
- »string«
- »array«
- »object«
- »unknown type«

Siehe auch bei `settype`.

intval

`intval` ermittelt den Ganzzahlwert einer Variablen zur angegebenen Basis *base*. Der Standardwert ist 10, die Angabe ist optional.

```
int intval(mixed var, int [base]);
```

Siehe auch `doubleval`, `strval`, `settype` und unter dem Stichwort Typkonvertierung.

doubleval

`doubleval` gibt den Inhalt einer Variablen als doppelt genaue Gleitkommazahl wieder. Sie müssen ein Skalar verwenden, Objekte oder Arrays sind nicht zulässig.

```
double doubleval(mixed var);
```

Siehe auch `intval`, `strval`, `settype`.

empty

`empty` ermittelt, ob eine Variable leer ist. Die Funktion gibt TRUE zurück, wenn die Variable leer ist.

```
int empty(mixed var);
```

Beachten Sie, dass eine Zeichenkettenvariable leer ist, wenn sie eine leere Zeichenkette enthält. Eine numerische Variable ist leer, wenn sie die Zahl 0 enthält. Um zu prüfen, ob eine Variable einen definierten Wert enthält, verwenden Sie `isset`.

Siehe auch `isset` and `unset`.

is_array

Die Funktion `is_array` gibt TRUE zurück, wenn die zu überprüfende Variable ein Array ist.

```
int is_array(mixed var);
```

Siehe auch `is_double`, `is_float`, `is_int`, `is_integer`, `is_real`, `is_string`, `is_long` und `is_object`.

is_double

is_double Die Funktion `is_double` gibt TRUE zurück, wenn der Zahlenwert, den die Variable speichert, doppelte Genauigkeit hat.

◆ `int is_double(mixed var);`

Siehe auch `is_array`, `is_float`, `is_int`, `is_integer`, `is_real`, `is_string`, `is_long` und `is_object`.

is_float

is_float Die Funktion `is_float` gibt TRUE zurück, wenn die zu überprüfende Variable eine Zahl doppelter Genauigkeit ist. Die Funktion entspricht exakt `is_double`.

◆ `int is_float(mixed var);`

Siehe auch `is_array`, `is_real`, `is_int`, `is_integer`, `is_real`, `is_string`, `is_long` und `is_object`.

is_int

is_int `is_int` ermittelt, ob eine Ganzzahl (Integer) vorliegt. Die Funktion ist ein Alias für `is_long`.

◆ `int is_int(mixed var);`

Siehe auch `is_double`, `is_float`, `is_integer`, `is_string`, `is_real`, `is_object`, `is_array` und `is_long`.

is_integer

is_integer `is_integer` ermittelt, ob eine Ganzzahl (Integer) vorliegt. Die Funktion ist ein Alias für `is_long`.

◆ `int is_integer(mixed var);`

Siehe auch `is_array`, `is_float`, `is_int`, `is_double`, `is_real`, `is_string`, `is_long` und `is_object`.

is_long

is_long `is_long` ermittelt, ob eine Ganzzahl (Integer) vorliegt.

◆ `int is_long(mixed var);`

Siehe auch `is_array`, `is_float`, `is_int`, `is_integer`, `is_real`, `is_string`, `is_double` und `is_object`.

`is_object`

`is_object` gibt TRUE zurück, wenn die Variable ein Objekt ist.

`is_object`

```
int is_object(mixed var);
```



Siehe auch `is_array`, `is_float`, `is_int`, `is_integer`, `is_real`, `is_string` und `is_long`.

`is_real`

`is_real` gibt TRUE zurück, wenn die Variable eine Zahl doppelter Genauigkeit ist. Die Funktion ist ein Alias für `is_double`.

`is_real`

```
int is_real(mixed var);
```



Siehe auch `is_array`, `is_float`, `is_int`, `is_integer`, `is_double`, `is_string`, `is_long` und `is_object`.

`is_string`

Die Funktion `is_string` gibt TRUE zurück, wenn es sich um eine Zeichenkette (String) handelt.

`is_string`

```
int is_string(mixed var);
```



Siehe auch `is_array`, `is_float`, `is_int`, `is_integer`, `is_real`, `is_double`, `is_long` und `is_object`.

`isset`

Mit `isset` prüfen Sie, ob eine Variable bereits definiert wurde. Die Variable kann trotzdem leer sein.

`isset`

```
int isset(mixed var);
```



Siehe auch `empty` und `unset`.

`settype`

`settype` setzt den Typ einer Variablen und wandelt den Inhalt entsprechend um. Wenn die Umwandlung erfolgreich war, wird TRUE zurückgegeben, andernfalls FALSE.

`settype`

```
int settype(string var, string type);
```



Mögliche Werte für *type* sind:

- "integer"
- "double"

- "string"
- "array"
- "object"

Siehe auch `gettype`.

strval

strval ermittelt aus einer Zeichenkette einen Wert. Sie können nur Skalare überprüfen, Arrays und Objekte sind nicht zulässig.



```
string strval(mixed var);
```

Siehe auch `doubleval`, `intval`, `settype`.

unset

unset setzt eine Variable auf den undefinierten Zustand zurück. Die Funktion gibt immer TRUE zurück.



```
int unset(mixed var);
```



```
unset( $foo );
```

```
unset( $bar['quux'] );
```

Siehe auch `isset` und `empty`.

Array-Funktionen

array

array erzeugt aus gegebenen Werten ein Array. Für assoziative Arrays kann der =>-Operator verwendet werden.



```
array array(...);
```

Dies ist eigentlich ein Sprachkonstrukt, wird hier jedoch bei den Funktionen aufgeführt, da das naheliegender ist. Das folgende Beispiel zeigt die Anwendung für ein zweidimensionales Array:



```
$fruits = array("fruits" =>
    array("a"=>"orange", "b"=>"banana", "c"=>"apple"),
    "numbers" =>
    array(1, 2, 3, 4, 5, 6),
    "holes" =>
    array("first", 5 => "second", "third")
);
```

Siehe auch `list`.

array_walk

array_walk wendet eine nutzerdefinierte Funktion auf jedes Element eines Arrays an.

```
int array_walk(array arr, string func);
```

Wenn die Funktion mehr als einen Parameter erwartet, wird bei jedem Aufruf eine Warnung ausgegeben. Diese Warnung kann unterdrückt werden, indem das Zeichen @ dem Aufruf vorangestellt wird.

Bedenken Sie, dass diese Funktion jedes Element des Arrays direkt behandelt, also das ursprüngliche Array ändert.

```
$fruits = array("d"=>"lemon",
               "a"=>"orange",
               "b"=>"banana",
               "c"=>"apple");
```

```
function test_alter( $item1 ) {
    $item1 = 'bogus';
}
```

```
function test_print( $item2 ) {
    echo "$item2<br>\n";
}
```

```
array_walk( $fruits, 'test_print' );
array_walk( $fruits, 'test_alter' );
array_walk( $fruits, 'test_print' );
```

Siehe auch `each` und `list`.

array_walk**arsort**

arsort sortiert ein Array rückwärts. Die Zuordnung der Indizes zu den Elementen bleibt erhalten. Die Sortierung erfolgt nach den Elementen.

```
void arsort(array array);
```

```
$cities = array("d"=>"Berlin",
               "a"=>"Moskau",
               "b"=>"Paris",
               "c"=>"London");
```

```
arsort($cities);
for(reset($cities); $key = key($cities); next($cities)) {
    echo "cities[$key] = ".$cities[$key]."\n";
}
```

Das Beispiel zeigt folgende Reihenfolge an:

arsort

```
cities[b] = Paris
cities[a] = Moskau
cities[c] = London
cities[d] = Berlin
```

Siehe auch `asort`, `rsort`, `ksort` und `sort`.

asort

asort

`asort` sortiert ein Array vorwärts (aufsteigend). Die Zuordnung der Indizes bleibt dabei erhalten.



```
void asort(array array);

$cities = array("d"=>"Berlin",
               "a"=>"Moskau",
               "b"=>"Paris",
               "c"=>"London");

asort($cities);
for(reset($cities); $key = key($cities); next($cities)) {
    echo "cities[$key] = ".$cities[$key]."\n";
}
```

Das Ergebnis des Skripts ist:

```
cities[d] = Berlin
cities[c] = London
cities[a] = Moskau
cities[b] = Paris
```

Siehe auch `arsort`, `rsort`, `ksort` und `sort`.

count

count

`count` ermittelt die Anzahl der Elemente eines Array. Es kann prinzipiell auch jeder andere Variablentyp angegeben werden.



```
int count(mixed var);
```

Die Funktion gibt 1 zurück, wenn `var` kein Array ist. Die Funktion gibt 0 zurück, wenn die Variable nicht definiert war. Wenn ein nicht initialisiertes Array vorliegt, wird ebenfalls 0 zurückgegeben. Testen Sie sicherheitshalber mit `isset`, ob eine Variable bereits existiert.

Siehe auch `sizeof`, `isset` und `is_array`.

current

current

`current` gibt das aktuelle Element eines Arrays zurück. Der interne Zeiger wird dabei nicht bewegt, das Array wird nicht verändert.

```
mixed current(array array);
```

Jedes Array hat einen internen Zeiger, der auf ein bestimmtes Element zeigt. Nach dem Hinzufügen zeigt der Zeiger auf dieses Element. Wenn der Zeiger seine Zuordnung verloren hat, wird FALSE zurückgegeben.

Wenn ein Element mit dem Wert FALSE im Array ermittelt wurde, wird ebenfalls FALSE zurückgegeben. Wenn dieser Fall auftreten kann und Sie sicher unterscheiden möchten, ob der Zeiger korrekt positioniert ist, müssen Sie die Funktion each verwenden.



Hinweis

Siehe auch each, end, next, prev und reset.

each

each gibt das nächste Schlüssel/Werte-Paar eines assoziativen Arrays zurück.

each

```
array each(array array);
```

Zurückgegeben werden vier Elemente mit den Indizes 0, 1, key und value. 0 und key enthalten den Schlüssel, 1 und value den Wert des Elements. Wenn der Zeiger nicht auf ein Element des Arrays positioniert ist, wird FALSE zurückgegeben.

Die Funktion each setzt den internen Zeiger des Arrays um ein Element weiter. Am Ende des Arrays verbleibt der Zeiger auf dem letzten Element.

```
$foo = array("joerg", "uwe", "chris", "andreas");  
$bar = each($foo);
```



Beispiel

\$bar enthält nun die folgenden Werte:

```
0 =>0  
1 => 'joerg'  
key => 0  
value => 'joerg'
```

```
$foo = array("Janine" => "Janni", "Clemens" => "Clemi");  
$bar = each($foo);
```



Beispiel

\$bar enthält nun die folgenden Werte:

```
0 => 'Janine'  
1 => 'Janni'  
key => 'Janine'  
value => 'Janni'
```

each wird typischerweise zusammen mit list verwendet, um ein Array zu durchlaufen, zum Beispiel das \$HTTP_POST_VARS-System-Array:



```
echo "Folgende Werte wurden per POST &uuml;bermittelt:<br>";
while (list($key, $val) = each($HTTP_POST_VARS)) {
    echo "$key => $val<br>";
}
```

Siehe auch key, list, current, reset, next und prev.

end

end end setzt den internen Zeiger eines Arrays auf das letzte Element.



```
mixed end(array array);
```

Siehe auch current, each, next und reset.

key

key key ermittelt einen Schlüssel von der aktuellen Position des internen Zeigers eines Arrays.



```
mixed key(array array);
```

Siehe auch current, next.

ksort

ksort ksort sortiert ein Array aufsteigend anhand der Schlüssel. Die Zuordnung zu den Elementen bleibt dabei erhalten. Die Anwendung ist nur für assoziative Arrays sinnvoll.



```
int ksort(array array);

$cities = array("d"=>"Berlin",
               "a"=>"Moskau",
               "b"=>"Paris",
               "c"=>"London");

ksort($cities);
for(reset($cities); $key = key($cities); next($cities)) {
    echo "cities[$key] = ".$cities[$key]."\n";
}
```

Dieses Beispiel gibt aus:

```
cities [a] = Moskau
cities [b] = Paris
cities [c] = London
cities [d] = Berlin
```

Siehe auch `asort`, `arsort`, `sort` und `rsort`.

krsort

Sortiert ein Array wie `ksort`, aber in absteigender Ordnung.

```
int krsort(array array);
```

Für ein Beispiel siehe `ksort`.

krsort



list

`list` weist einer Gruppe Variablen mehrere Werte zu, meist zur Auflösung eines Arrays oder ähnlicher Auflistung.

```
void list(...);
```

Diese Funktion ist ein Sprachkonstrukt.

```
<table>
<tr>
<th>Mitarbeiter</th>
<th>Gehalt</th>
</tr>
<?php
$result = mysql($conn, "SELECT id, name, salary
                    FROM employees");
while (list($id, $name, $salary) =
        mysql_fetch_row($result)) {
    print(" <tr>\n" .
          " <td><a href=\"info .
          php3?id=$id\">$name</a></td>\n" .
          " <td>$salary</td>\n" .
          " </tr>\n");
}
?></table>
```



list



Siehe auch `each` und `array` sowie alternativ `foreach`.

next

`next` setzt den internen Zeiger eines Elements um eine Position weiter und gibt das Element an der neuen Position zurück. Wenn das Array keine Elemente mehr enthält, wird `FALSE` zurückgegeben.

```
mixed next(array array);
```

Wenn das Array Elemente mit dem Wert `FALSE` enthält, können Sie nicht sicher feststellen, ob der Zeiger den Wertebereich verlassen

next



hat oder das Element zurückgegeben wird. Verwenden Sie anstatt `next` dann die Funktion `each`.

Siehe auch `current`, `end`, `prev` und `reset`.

pos

pos `pos` holt das aktuelle Element eines Arrays. Dies ist ein Alias für `current`.

◆ `mixed pos(array array);`

Siehe auch `end`, `next`, `prev` und `reset`.

prev

prev `prev` setzt den internen Zeiger eines Arrays um ein Element zurück. Wenn sich der Zeiger über den Anfang des Arrays bewegt, wird `FALSE` zurückgegeben. Die Funktion gibt außerdem das Element zurück, auf das der Zeiger nach der Operation zeigt.

◆ `mixed prev(array array);`

Siehe auch `current`, `end`, `next` und `reset`.

range

range Die Funktion `range` erzeugt ein Array mit Ganzzahlen, die aus dem angegebenen Zahlenbereich stammen, die Werte `low` und `high` inklusive.

◆ `array range(int low, int high);`

Siehe `shuffle` für ein Anwendungsbeispiel.

reset

reset `reset` setzt den internen Zeiger eines Arrays auf das erste Element. Das erste Element des Arrays wird zurückgegeben.

◆ `mixed reset(array array);`

Siehe auch `current`, `each`, `next`, `prev` und `reset`.

rsort

rsort `rsort` sortiert ein Array absteigend (rückwärts). Die Funktion geht davon aus, dass es sich um ein eindimensionales Array handelt.

◆ `void rsort(array array);`

```
$cities = array("Berlin","London","New York","Rio");
rsort($cities);
for(reset($cities); list($key,$value) = each($cities);)
{
    echo "cities[$key] = ", $value, "\n";
}
```



Dieses Skript gibt aus:

```
cities[0] = Rio
cities[1] = New York
cities[2] = London
cities[3] = Berlin
```

Siehe auch `arsort`, `asort`, `ksort`, `sort` und `usort`.

shuffle

`shuffle` bringt ein Array in »Unordnung«. Die Funktion ordnet die Elemente mit Hilfe des Zufallsgenerators neu.

shuffle

```
void shuffle(array array);

$numbers = range(1,20);
srand(time());
shuffle($numbers);
while (list(,$number) = each($numbers)) {
    echo "$number ";
}
```



Siehe auch `arsort`, `asort`, `ksort`, `rsort`, `sort` und `usort`.

sizeof

`sizeof` ermittelt die Anzahl der Elemente eines Arrays.

sizeof

```
int sizeof(array array);
```



Siehe auch `count`.

sort

`sort` sortiert ein eindimensionales Array aufsteigend.

sort

```
void sort(array array);

$fruits = array("Zitrone","Orange","Banane","Apfel");
sort($fruits);
for(reset($fruits); $key = key($fruits); next($fruits))
{
    echo "fruits[$key] = ".$fruits[$key]."\n";
}
```



Das Beispiel gibt Folgendes aus:

```
fruits[0] = Apfel
fruits[1] = Banane
fruits[2] = Orange
fruits[3] = Zitrone
```

Siehe auch `arsort`, `asort`, `ksort`, `rsort` und `usort`.

uasort

uasort

`uasort` sortiert ein Array anhand einer selbst definierten Funktion *cmp_function*. Die Funktion muss einen Vergleich zwischen zwei Elementen ausführen und `-1`, `0` oder `1` für kleiner, gleich oder größer zurückgeben.



```
void uasort(array array, function cmp_function);
```

uksort

uksort

`uksort` sortiert ein assoziatives Array anhand der Schlüssel und mit Hilfe einer selbst definierten Funktion *cmp_function*. Die Funktion muss einen Vergleich zwischen zwei Elementen ausführen und `-1`, `0` oder `1` für kleiner, gleich oder größer zurückgeben.



```
void uksort(array array, function cmp_function);
```



```
function mycompare($a, $b) {
    if ($a == $b) return 0;
    return ($a > $b) ? -1 : 1;
}

$a = array(4 => "four", 3 => "three",
          20 => "twenty", 10 => "ten");
uksort($a, mycompare);
while(list($key, $value) = each($a)) {
    echo "$key: $value\n";
}
```

Folgende Anzeige erfolgt:

```
20: twenty 10: ten 4: four 3: three
```

Siehe auch `arsort`, `asort`, `uasort`, `ksort`, `rsort` und `sort`.

usort

usort

`usort` sortiert ein assoziatives Array anhand der Werte und mit Hilfe einer selbst definierten Funktion *cmp_function*. Die Funktion muss einen Vergleich zwischen zwei Elementen ausführen und `-1`, `0` oder `1` für kleiner, gleich oder größer zurückgeben.

```
void usort(array array, function cmp_function);  
  
function cmp($a,$b) {  
    if ($a == $b) return 0;  
    return ($a > $b) ? -1 : 1;  
}  
$a = array(3,2,5,6,1);  
usort($a, cmp);  
while(list($key,$value) = each($a)) {  
    echo "$key: $value\n";  
}
```



Die Ausgabe dieses Beispiels:

```
0: 6  
1: 5  
2: 3  
3: 2  
4: 1
```

Siehe auch arsort, asort, ksort, rsort und sort.

Neue Array-Funktionen in PHP4

In PHP4 kamen eine ganze Reihe neuer Funktionen hinzu. Damit die (heute noch) überwiegende Anzahl von PHP3-Nutzern nicht ständig mit diesen Funktionen konfrontiert wird, stehen diese in einem eigenen Abschnitt.



array_count_values

`array_count_values` erstellt ein Array, das die Elemente des übergebenen Arrays als Schlüssel und die Häufigkeit des Auftretens als Wert enthält.

array_count_values

```
array array_count_values(array input);  
  
$array = array(1, "hello", 1, "world", "hello");  
$count = array_count_value($array);
```



Das Beispiel gibt das folgende Array *\$count* zurück:

```
1 => 2, "hello" => 2, "world" => 1
```

Die Ziffer »1« und das Wort »hello« sind zweimal, das Wort »world« nur einmal enthalten.

array_diff

Die Funktion `array_diff` vergleicht zwei oder mehrere Arrays miteinander und gibt ein Array zurück, das die Unterschiede enthält.

array_diff

◆ array **array_diff**(array ar1, array ar2 [, array arX])

array_flip

array_flip Die Funktion vertauscht alle Schlüssel mit den Werten in einem assoziativen Array.

```
array array_flip(array input);
```

array_keys

array_keys array_keys gibt die Schlüssel eines zweidimensionalen Arrays als eindimensionales Array zurück.

◆ array **array_keys**(array input, mixed [search_value]);

Der Parameter *search_value* ist optional. Wird er angegeben, werden nur jene Schlüssel zurückgegeben, denen der entsprechende Wert zugeordnet ist.



Beispiel

```
$array = array(0 => 100, "color" => "red");
$result = array_keys($array);
```

Diese Beispiel gibt (0, "color") zurück.

```
$array = array(1, 100, 2, 100);
$result = array_keys($array, 100);
```

In diesem Beispiel wird die Suche eingeschränkt. Das zurückgegebene Array enthält (0, 2).

Siehe auch array_values.

array_merge

array_merge array_merge verbindet zwei Arrays, indem ein Array an das andere angehängt wird.

◆ array **array_merge**(array array1, array array2);

Wenn die Funktion auf zweidimensionale assoziative Arrays angewendet wird und gleiche Schlüssel auftreten, überschreibt der zuletzt gelieferte Wert den vorhergehenden.



Beispiel

```
$array1 = array("color" => "red", 2, 4);
$array2 = array("a","b","color" => "green","shape" => "trapez");
$merge = array_merge($array1, $array2);
```

Das Beispiel ergibt folgendes Array:

```
"color" => "green", 2, 4, "a", "b", "shape" => "trapez"
```

array_merge_recursive

Diese Funktion verbindet zwei oder mehrere Arrays. Dabei wird ein Array immer an das vorhergehende gehängt. Wenn die Eingabearrays dieselben Schlüssel haben, werden diese Schlüssel zusammengefasst und aus den zugehörigen Werten wird ein Array gebildet. Dieser Prozess läuft rekursiv ab.

array_merge_recursive

```
array array_merge_recursive(array ar1, array ar2 [, array arX])
```

**array_multisort**

Diese Funktion sortiert ein mehrdimensionales oder mehrere Arrays. Das Array wird im Original geändert, die Funktion gibt TRUE zurück, wenn die Sortierung erfolgreich war.

array_multisort

```
bool array_multisort(array input [, mixed arg [, mixed ...]])
```



Die Parameter sind im einfachsten Fall ein Array und ein Sortierkriterium. Wenn mehrere Arrays sortiert werden, wird immer ein Array von den Sortierkriterien gefolgt. Diese können folgende Konstanten sein:

Bestimmung der Sortierreihenfolge:

- SORT_ASC. Aufsteigend
- SORT_DESC. Absteigend

Bestimmung der Sortierart:

- SORT_REGULAR. Standardvergleich
- SORT_NUMERIC. Numerischer Vergleich
- SORT_STRING. Vergleich als Zeichenketten

Das folgende Beispiel zeigt, wie ein zweidimensionales Array sortiert wird:

```
$ar = array (array ("10", 100, 100, "a"), array (1, 3, "2", 1));
array_multisort ($ar[0], SORT_ASC, SORT_STRING,
                $ar[1], SORT_NUMERIC, SORT_DESC);
```



Beispiel

array_pad

array_pad vergrößert ein Array am Anfang oder am Ende. Dabei werden die neuen Elemente mit einem Standardwert aufgefüllt.

array_pad

```
array array_pad(array input, int pad_size, mixed pad_value);
```





```
$input = array(12, 10, 9);
$result = array_pad($input, 5, 0);
```

Dieses Beispiel ergibt folgendes Array:

```
12, 10, 9, 0, 0
```



```
$input = array(12, 10, 9);
$result = array_pad($input, -7, -1);
```

In diesem Beispiel wird das Array vorn aufgefüllt, das Resultat ist:

```
-1, -1, -1, -1, 12, 10, 9
```

Wird auf das Array *\$input* im letzten Beispiel der Befehl

```
array_pad($input, 2, "noop");
```

angewendet, passiert nichts.

array_pop

array_pop

`array_pop` entfernt das letzte Element eines Arrays und gibt es zurück. Das Array ist danach ein Element kleiner.



```
mixed array_pop(array array);
```



```
$stack = array("orange", "apple", "raspberry");
$fruit = array_pop($stack);
```

Nach der Ausführung des Beispiels enthält *\$fruit* den Wert »raspberry« und das Array *\$stack* enthält "orange", "apple".

array_push

array_push

`array_push` fügt ein Element am Ende eines Arrays hinzu. Das Array wächst dabei um ein Element.



```
int array_push(array array, mixed var, mixed [var2,...]);
```



```
$stack = array(1, 2);
array_push($stack, "+", 3);
```

Nach der Ausführung dieses Beispiels enthält *\$stack* vier Elemente:

```
1, 2, "+", 3
```

Siehe auch `array_pop`, `array_shift`, `array_unshift`.

array_reverse

array_reverse

`array_reverse` dreht ein Array um. Das letzte Element steht nun an der ersten Position und umgekehrt.

```
array array_reverse(array input);
```

Das Beispiel zeigt, wie sich die Funktion bei komplexen Arrays verhält:

```
$input = array("php", 4.0, array("green", "red"));
$result = array_reverse($input);
```

Das Ergebnis *\$result* könnte auch folgendermaßen dargestellt werden:

```
array(array("green", "red"), 4.0, "php");
```



Beispiel

array_shift

`array_shift` entfernt ein Element vom Anfang und gibt es zurück. Alle Elemente rutschen eine Position nach unten. Das Array ist nun um ein Element kleiner.

```
mixed array_shift(array input);
```

```
$args = array("-v", "-f");
$opt = array_shift($args);
```

Nach der Ausführung dieses Beispiels enthält *\$args* den Wert "-f" und *\$opt* den Wert "-v".

array_shift



Beispiel

array_unique

Die Funktion untersucht ein Array und entfernt alle doppelten Elemente.

```
array array_unique(array input)
```

Das folgende Beispiel zeigt dies anhand eines eindimensionalen Arrays:

```
$array = array("10999", "12683", "12587", "12683", "10999")
$array = array_unique($array)
```

Das Ergebnis in *\$array*: ("10999", "12683", "12587").

array_unique



Beispiel

array_unshift

`array_unshift` fügt mehrere Elemente am Anfang eines Arrays ein. Die Elemente werden einzeln, nicht als Array, übergeben. Das resultierende Array wächst dabei um mehrere Elemente.

```
int array_unshift(array input, mixed val, mixed [val2,...]);
```

```
$queue = array("p1", "p3");
array_unshift($queue, "p4", "p5", "p6");
```



Beispiel

Nach der Ausführung dieses Beispiels enthält das Array *\$queue* folgende Elemente:

```
"p4", "p5", "p6", "p1", "p3"
```

array_slice

array_slice

`array_slice` gibt einige (bestimmte) Elemente des Arrays zurück. Die Elemente werden anhand der Position im Array ausgewählt.



```
array array_slice(array input, int offset, int [length]);
```

Der Parameter *offset* gibt an, ab welcher Position Elemente zurückgegeben werden. Der optionale Parameter *length* gibt an, wie viele Elemente zurückgegeben werden. Ist *offset* negativ, wird vom Ende des Array gezählt. Ist *length* negativ, stoppt die Auswahl nach der angegebenen Anzahl, vom Ende des Array gezählt.



```
$input = array("a", "b", "c", "d", "e");
$output = array_slice($input, 2);
// Gibt "c", "d", "e" zurück
```

```
$output = array_slice($input, 2, -1);
// Gibt "d" zurück
```

```
$output = array_slice($input, -2, 1);
// Gibt "a", "b", "c" zurück
```

Siehe auch unter `array_splice`.

array_splice

array_splice

`array_splice` ersetzt einige (bestimmte) Elemente des Arrays. Die durch die Position im Array beschriebenen Elemente werden durch bestimmte Werte ersetzt.



```
array array_splice(array input, int offset,
                   int [length], array [replacement]);
```

Die Parameter *offset* und *length* bestimmen die Position und Anzahl der zu ersetzenden Elemente in *input*. Beide Parameter verhalten sich wie bei `array_slice` beschrieben. Wird *replacement* angegeben, werden die Elemente ersetzt, ansonsten gelöscht. Wenn die Parameter *offset* und *length* Werte haben, durch die keine Elemente ausgewählt sind, wird *replacement* dem Array hinzugefügt.



```
$input = array("red", "green", "blue", "yellow");
array_splice($input, 2);
// $input enthält jetzt: "red", "green"
```

```
array_splice($input, 1, -1);
```

```
// $input enthält jetzt: "red", "yellow"

array_slice($input, -1, 1, array("black", "maroon"));
// $input enthält jetzt:
// "red", "green", "blue", "black", "maroon"
```

Mit der sehr universellen Funktion können einige Sonderfälle anderer Funktionen nachgebildet werden:

```
array_push($input, $x, $y);
```

entspricht

```
array_splice($input, count($input), 0, array($x, $y);
```

```
array_pop($input);
```

entspricht

```
array_splice($input, -1);
```

```
array_shift($input);
```

entspricht

```
array_splice($input, 0, 1);
```

```
array_unshift($input, $x, $y);
```

entspricht

```
array_splice($input, 0, 0, array($x, $y);
```

```
$a[$x] entspricht array_splice($input, $x, 1, $y);
```



Beispiel

array_values

array_values gibt alle Werte eines zweidimensionalen, assoziativen Arrays zurück.

```
array array_values(array input);
```

```
$input = array("size" => "XL", "color" => "gold");
$result = array_values($input);
```

Das Beispiel gibt in *\$result* folgendes Array zurück:

```
"XL", "gold"
```



Beispiel

compact

compact überführt Variablen und deren Werte in ein Array. Die Funktion kann zum Prüfen von Variablen genutzt werden, außerdem vereinfacht sie die Parameterübergabe.

compact



```
array compact(string varname | array varnames [, ...]);
```

Im resultierenden Array wird der Schlüssel der Variablenname, der Wert enthält den Inhalt der Variablen.



```
$city = "San Fransisco";
$state = "CA";
$event = "SIGGRAPH";
$location_vars = array("city", "state");
$result = compact("event", $location_vars);
```

Dieses Beispiel generiert das folgende Array in *\$result*:

```
"event" => "SIGGRAPH", "city" => "San Fransisco", "state" => "CA"
```

Siehe auch unter `extract` (PHP3-Funktion).

extract

extract

Diese Funktion erzeugt Variablen aus einem Array, indem die Namen in der Symboltabelle angelegt werden.

```
void extract(array var_array [, int extract_type
                [, string prefix]])
```

Das Array mit den Daten muss Paare aus Schlüsseln, welche die Namen ergeben, und Werte, die den Variablen zugewiesen werden, gefüllt sein.

Der Parameter `extract_type` steuert, wie das Verhalten beim Überschreiben vorhandener Werte gesteuert wird. Möglich sind folgende Werte:

- `EXTR_OVERWRITE`. Überschreiben (Standardwert)
- `EXTR_SKIP`. Nicht Überschreiben
- `EXTR_PREFIX_SAME`. Neue Variable mit Präfix *prefix*
- `EXTR_PREFIX_ALL`. Alle Variablen mit Präfix versehen *prefix*



```
<?
$var_array = array("color" => "#DDEEDD",
                  "font-size" => "12",
                  "font-family" => "Helvetica, Univers");
extract($var_array, EXTR_PREFIX_SAME, "css");
echo "$css_color, $css_font-size, $css_font-family";
?>
```

`in_array`

`in_array` untersucht ein Array auf das Vorhandensein von Elementen mit bestimmten Eigenschaften und gibt einen Booleschen Wert zurück.

```
bool in_array(mixed needle, array haystack);
```

```
$os = array("Mac", "NT", "Irix", "Linux");  
if in_array("Irix", $os) {  
    echo "Irix wird unterstützt";  
}
```

`in_array`



Beispiel

Zeichenkettenfunktionen

`addslashes`

`addslashes` versieht in einer Zeichenkette bestimmte Elemente mit einem Backslash. Die Zeichenkette wird modifiziert zurückgegeben.

```
string addslashes(string str);
```

Erkannt werden:

- Einfaches Anführungszeichen (')
- Doppeltes Anführungszeichen (")
- Backslash (\)
- NUL (0-Byte).

Siehe auch `stripslashes`, `htmlspecialchars` und `quotemeta`.

`addslashes`



`addcslashes`

`addcslashes` versieht in einer Zeichenkette bestimmte Elemente mit einem Backslash. Die Zeichenkette wird modifiziert zurückgegeben.

```
string addcslashes(string str, string charlist);
```

Die Zeichenkette `charlist` enthält alle Zeichen, die mit einem Backslash versehen werden sollen. Zeichen, die nicht per Tastatur angegeben werden können, schreiben Sie im Oktalformat. Beispielsweise werden von der folgenden Anweisung die Zeichen `\`, `@`, `!` und `\188` umgewandelt:

```
$withslash = addcslashes($toconvert, "\\@!\\188");
```



Beispiel

Es können außerdem auch Bereiche definiert werden. Alle alphanumerischen Zeichen werden folgendermaßen umgewandelt:

```
$withslash = addslashes($toconvert, "\0..\37");
```

Siehe auch `stripslashes`, `htmlspecialchars` und `quotemeta`.

bin2hex

bin2hex

`bin2hex` konvertiert eine als Zeichenkette übermittelte Binärzahl (Folge von Nullen und Einsen) als Hexadezimalzahl. Die Umwandlung erfolgt byteweise mit dem höchsten Nibbel (4 Bit-Teil) zuerst.

◆ `string bin2hex(string str);`

chop

chop

`chop` entfernt Leerzeichen vom Ende der Zeichenkette und gibt die modifizierte Zeichenkette zurück.

◆ `string chop(string str);`



Beispiel

```
$trimmed = chop($line);
```

Siehe auch `trim`.

chr

chr

`chr` gibt das Zeichen zu einem gegebenen ASCII-Wert zurück.

◆ `string chr(int ascii);`

Die Umkehrfunktion dazu ist `ord`.



Beispiel

Die beiden folgenden Varianten sind gleichwertig:

```
$str .= chr(27);
```

```
$str = sprintf("The string ends in escape: %c", 27);
```

chunk_split

chunk_split

Die Funktion `chunk_split` teilt eine Zeichenkette in kleinere Stücke konstanter Länge.

◆ `string chunk_split(string string, int [chunklen], string [end]);`

Die Funktion wird angewendet, um bei der `base64_encode`-Funktion der Restriktion der RFC 2045 zu entsprechen, die nur maximal 76 Byte lange Zeichenketten zulässt. Die Funktion gibt eine neue Zei-

chenkette zurück und lässt den übergebenen Parameter unverändert. Die Funktion ist signifikant schneller als `ereg_replace`.

```
// format $data using RFC 2045 semantics
$new_string = chunk_split(base64_encode($data));
```



Die Funktion ist signifikant schneller als `ereg_replace`.

convert_cyr_string

`convert_cyr_string` konvertiert Zeichen zwischen kyrillischen Zeichensätzen.

convert_cyr_string

string **convert_cyr_string**(string str, string from, string to);



Unterstützt werden folgende kyrillische Zeichensätze:

- k - koi8-r
- w - windows-1251
- i - iso8859-5
- a - x-cp866
- d - x-cp866
- m - x-mac-cyrillic

count_chars

Ermittelt die Anzahl gleicher Zeichen in einer Zeichenkette.

count_chars

mixed **count_chars**(string string [, mode])



Der optionale Parameter *mode* ist standardmäßig 0 und je nach Einstellung wird die Funktion folgendermaßen arbeiten:

- 0. Ein Array mit dem Bytewert als Schlüssel und der Anzahl des Auftretens als Wert.
- 1. Wie 0, aber nur mit Werten größer 0.
- 2. Wie 0, aber nur mit Werten gleich 0.
- 3. Eine Zeichenkette mit allen verwendeten Bytes wird zurückgegeben.
- 4. Eine Zeichenkette mit allen nicht verwendeten Bytes wird zurückgegeben.

crc32

crc32 Berechnet das CRC32-Polynom (Prüfsumme) einer Zeichenkette.

◆ `int crc32(string str)`

Die Funktion erzeugt die zyklisch redundante Prüfsumme mit 32 Bit Länge. Diese Prüfsumme kann eingesetzt werden, um die Integrität von Daten zu prüfen.

Siehe auch `md5`.

crypt

crypt `crypt` verschlüsselt eine Zeichenkette nach DES (Standard UNIX DES).

◆ `string crypt(string str, string [salt]);`

Das Argument *str* ist der zu verschlüsselnde Text. PHP stellt fest, ob Ihr System den besseren MD5-Algorithmus unterstützen kann, und liefert einen entsprechenden Startwert für *salt*, wenn keiner angegeben wird. Sie können mit der Konstante `CRYPT_SALT_LENGTH` feststellen, ob DES oder MD5 verwendet wird. DES hat 2 Byte Startwerte, MD5 12 Byte. Der tatsächlich verwendete Wert wird in den ersten beiden Bytes der ausgegebenen Zeichenkette gespeichert. Systeme, die mehrere Verschlüsselungsoptionen unterstützen, zeigen dies mit mehreren Konstanten an:

- `CRYPT_STD_DES` – Standard-DES-Verschlüsselung mit 2-Byte-Startwert.
- `CRYPT_EXT_DES` – Erweiterte-DES-Verschlüsselung mit 9-Zeichen-Startwert.
- `CRYPT_MD5` – MD5-Verschlüsselung mit 12-Zeichen-Startwert, der mit der Zeichenfolge `1` beginnt.
- `CRYPT_BLOWFISH` – Erweiterte DES-Blowfish-Verschlüsselung mit 16-Zeichen Startwert und der Zeichenfolge `2` am Anfang.

echo

echo `echo` gibt Zeichenketten aus. Es können beliebig viele Parameter angegeben werden, die alle nacheinander ausgegeben werden. Innerhalb von Zeichenketten mit doppelten Anführungszeichen werden Variablen erkannt und ausgewertet, bei einfachen Anführungszeichen erfolgt dies nicht.

```
echo(string arg1, string [argn]...);
```

echo ist ein Sprachkonstrukt. Die Klammern sind deshalb optional.

```
echo "Hello World";
echo Text;
echo "Es ist heute $wochentag";
echo 'Das Spiel kostet $17.';
```



Beispiel

Mehrere Zeichenketten kombinieren Sie mit dem Operator ».«.

Bei Variablen mit Bindestrich muss die Angabe in der Zeichenkette folgendermaßen aussehen:

```
echo "Der Buchtitel lautet ${"T-Title"}";
```

Siehe auch `print`, `printf`, `flush`.



Tipp

explode

`explode` zerteilt eine Zeichenkette anhand eines Separators und gibt die Elemente als Array zurück.

```
array explode(string separator, string string);

$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
```

Siehe auch `split` und `implode`.

explode



Beispiel

get_html_translation_table

Diese Funktion gibt die Übersetzungstabelle für die HTML-Entitäten zurück. In dieser Tabelle steht die direkte Zuordnung zwischen Zeichencodes und HTML-Darstellung.

```
array get_html_translation_table(int table)
```

Der Parameter *table* kann einen der beiden folgenden Konstanten annehmen:

- `HTML_ENTITIES`
- `HTML_SPECIALCHARS`

Siehe auch `htmlentities` und `htmlspecialchars`.

get_html_translation_table



get_meta_tags

`get_meta_tags` liest eine Datei, ermittelt darin alle `<META>`-Tags und gibt die gefundenen Tags als assoziatives Array zurück. Der Name wird zum Schlüssel, der Inhalt zum Wert des Arrays.

get_meta_tags

◆ `array get_meta_tags(string filename, int [use_include_path]);`

Die Funktion nutzt Aufrufe des Betriebssystems. So kann eine Datei mit Zeilenumbrüchen von einem Mac-System auf einer UNIX-Maschine Probleme bereiten. Sonderzeichen werden durch den Unterstrich ersetzt.



```
<meta name="author" content="name">
<meta name="tags" content="php3 documentation">
</head> <!-- parsing stops here -->
```

htmlspecialchars

htmlspecialchars htmlspecialchars wandelt Sonderzeichen in die entsprechenden HTML-Codes um.

◆ `string htmlspecialchars(string string);`

Die folgenden Zeichen werden erkannt:

- & (ampersand) wird zu &
- " (double quote) wird zu "
- < (less than) wird zu <
- > (greater than) wird zu >

Verwenden Sie `htmlentities`, wenn Sie auch Umlaute wandeln möchten.

Siehe auch `htmlentities` und `n12br`.

htmlentities

htmlentities htmlentities wandelt alle bekannten Sonderzeichen in die entsprechenden HTML-Codes um. Der ISO-8859-1-Zeichensatz wird zur Erkennung der Sonderzeichen verwendet.

◆ `string htmlentities(string string);`

Siehe auch `htmlspecialchars` und `n12br`.

implode

implode implode verbindet die Elemente eines Arrays mit einem Separator zu einer Zeichenkette.

◆ `string implode(string separator, array pieces);`



```
$colon_separated = implode(":", $array);
```

Siehe auch `explode`, `join` und `split`.

join

`join` ist ein Alias für `implode`.

```
string join(string glue, array pieces);
```

join



ltrim

`ltrim` entfernt Leerzeichen vom Anfang einer Zeichenkette.

```
string ltrim(string str);
```

Siehe auch `chop` und `trim`.

ltrim



md5

`md5` errechnet den MD5-Code einer Zeichenkette.

```
string md5(string str);
```

MD5 ist der Message-Digest Algorithm der RSA Data Security, Inc. Weitere Informationen finden Sie unter der folgenden Adresse:

- <http://ds.internic.net/rfc/rfc1321.txt>

md5



metaphone

Die Funktion erzeugt einen »Klangschlüssel« eines Wortes. Damit können phonetische Vergleiche durchgeführt werden.

```
string metaphone(string str)
```

Die Funktion ist exakter als `soundex`, weil grundlegende Regeln der englischen Sprache beachtet werden. Allerdings funktioniert das mit deutschen Wörtern nur begrenzt.

Siehe auch `soundex`.

metaphone



n12br

`n12br` wandelt Zeilenumbrüche und den HTML-Code `
` um.

```
string n12br(string string);
```

Siehe auch `htmlspecialchars` und `htmlentities`.

n12br



ord

ord ord gibt den ASCII-Wert eines Zeichens zurück. Wird eine Zeichenkette mit mehr als einem Zeichen übergeben, wird der Wert nur für das erste Zeichen ermittelt.

◆ `int ord(string string);`

Die Umkehrfunktion dazu ist `chr`.



Beispiel

```
if (ord($str) == 10) {
    echo("Das erste Zeichen in \"$str\"
        ist ein Zeilenvorschub.\n");
}
```

Siehe auch `chr`.

parse_str

parse_str `parse_str` durchsucht eine Zeichenkette, als handelte es sich um den GET-Parameter (QueryString), und gibt die Werte als Elemente eines Arrays zurück.

◆ `void parse_str(string str);`



Beispiel

```
$str = "eins=Wert1&zwei[]=Wert2&drei[]=Wert3";
parse_str($str);
echo $eins; /* gibt "Wert1" aus */
echo $zwei[0]; /* gibt "Wert2" aus */
echo $drei[1]; /* gibt "Wert3" aus */
```

print

print `print` gibt eine Zeichenkette aus.

◆ `print(string arg);`

Siehe auch `echo`, `printf` und `flush`.

printf

printf `printf` gibt eine Zeichenkette aus und beachtet dabei Formatierungsanweisungen. Eine genaue Beschreibung der Parameter finden Sie unter `sprintf` auf Seite 127.

◆ `int printf(string format, mixed [args]...);`

Siehe auch `print`, `sprintf` und `flush`.

`quoted_printable_decode`

`quoted_printable_decode` konvertiert eine Zeichenkette, die »Quoted Printable«-codiert ist, in 8-Bit-Werte. »Quoted Printable« wird bei der Codierung von MIME-Anhängen verwendet.

`quoted_printable_decode`

```
string quoted_printable_decode(string str);
```



Siehe auch `imap_qprint`.

`quotemeta`

`quotemeta` fügt Zeichen mit besonderer Bedeutung einen Backslash hinzu.

`quotemeta`

```
int quotemeta(string str);
```



Erkannt werden:

- `. \ + * ? [^] ($)`

Siehe auch `addslashes`, `htmlentities`, `htmlspecialchars`, `nl2br` und `stripslashes`.

`rawurldecode`

`rawurldecode` decodiert URL-codierte Zeichenketten.

`rawurldecode`

```
string rawurldecode(string str);
```



Bei der URL-Codierung werden Sonderzeichen durch % und einen zweistelligen Hexadezimalwert des ASCII-Codes des Zeichens ersetzt. Die Funktion `rawurlencode` macht diese Codierung rückgängig.

```
foo%20bar%40baz
```

wird zu:

```
foo bar@baz
```

Siehe auch `rawurlencode`.



Beispiel

`rawurlencode`

`rawurlencode` kodiert Zeichenketten nach RFC 1738. Dabei werden alle Sonderzeichen außer »-« und »_« durch % und einen zweistelligen Hexadezimalwert des ASCII-Codes des Zeichens ersetzt.

`rawurlencode`

```
string rawurlencode(string str);
```





```
echo '<A HREF="ftp://user:', rawurlencode ('foo @+%/'),
      '@ftp.my.com/x.txt">';
```

```
echo '<A HREF="http://x.com/department_list_script/',
      rawurlencode ('sales and marketing/Miami'), '>';
```

Siehe auch `rawurldecode`.

setlocale

setlocale

`setlocale` setzt Lokalisierungsinformationen.



```
string setlocale(string category, string locale);
```

category gibt die Kategorie an:

- LC_ALL für alle folgenden Werte.
- LC_COLLATE für Zeichenkettenvergleiche.
- LC_CTYPE Zeichenkettenumwandlungen wie `strtoupper`.
- LC_MONETARY für `localeconv()`.
- LC_NUMERIC für die Dezimaltrennzeichen.
- LC_TIME für die Formatierung von Datumswerten mit `strftime`.

Wird als Parameter »0« übergeben, wird der aktuelle Wert zurückgegeben, aber nicht verändert.

*Tabelle B.Fehler!
Kein Text mit
angegebener
Formatvorlage im
Dokument.1:
Abkürzungen der
Gebietsschemen. Die
Auflistung ist eine
Garantie dafür, dass
die Schemen von
PHP auch
unterstützt werden.*

Gebietsschema	locale	Gebietsschema	locale
Afrikaans	af	Japanisch	ja
Albanisch	sq	Jiddisch	ji
Arabisch - Ägypten	ar-eg	Katalanisch	ca
Arabisch - Algerie	ar-dz	Koreanisch	ko
Arabisch - Bahrai	ar-bh	Kroatisch	hr
Arabisch - Ira	ar-iq	Lettisch	lv
Arabisch - Jeme	ar-ye	Litauisch	lt
Arabisch - Jordanien	ar-jo	Malaiisch - Malaysi	ms
Arabisch - Kuwai	ar-kw	Maltesisch	mt
Arabisch - Libano	ar-lb	Mazedonisch	mk
Arabisch - Libye	ar-ly	Niederländisch	nl
Arabisch - Marokko	ar-ma	Niederländisch - Belgien	nl-be
Arabisch - Oman	ar-om	Norwegisch - Bokmål	no

Gebietsschema	locale	Gebietsschema	locale
Arabisch – Qatar	ar-qa	Polnisch	pl
Arabisch – Saudi-Arabien	ar-sa	Portugiesisch – Brasilien	pt-br
Arabisch – Syrien	ar-sy	Portugiesisch – Standard	pt
Arabisch – Tunesien	ar-tn	Rätoromanisch	rm
Arabisch – V.A.E.	ar-ae	Rumänisch	ro
Baskisch	eu	Rumänisch - Moldawien	ro-mo
Bulgarisch	bg	Russisch	ru
Chinesisch	zh	Russisch - Moldawien	ru-mo
Chinesisch – Hongkong	zh-hk	Schwedisch	sv
Chinesisch – Singapur	zh-sg	Schwedisch - Finnland	sv-fi
Chinesisch – Taiwan	zh-tw	Serbisch - Kyrillisch	sr
Chinesisch – VRC	zh-cn	Setschuan	tn
Dänisch	da	Slowakisch	sk
Deutsch – Liechtenstein	de-li	Slowenisch	sl
Deutsch – Luxemburg	de-lu	Sorbisch	sb
Deutsch – Österreich	de-at	Spanisch - Argentinien	es-ar
Deutsch – Schweiz	de-ch	Spanisch – Bolivien	es-bo
Deutsch – Standard	de	Spanisch – Chile	es-cl
Englisch	en	Spanisch – Costa Rica	es-cr
Englisch – Australien	en-au	Spanisch – Dominikanische Republik	es-do
Englisch – Belize	en-bz	Spanisch – Ecuador	es-ec
Englisch – Großbritannien	en-gb	Spanisch – El Salvador	es-sv
Englisch – Irland	en-ie	Spanisch – Guatemala	es-gt
Englisch – Jamaika	en-jm	Spanisch – Honduras	es-hn
Englisch – Kanada	en-ca	Spanisch – Kolumbien	es-co
Englisch – Neuseeland	en-nz	Spanisch – Mexiko	es-mx
Englisch – Südafrika	en-za	Spanisch – Nicaragua	es-ni
Englisch – Trinidad	en-tt	Spanisch – Panama	es-pa
Englisch – USA	en-us	Spanisch – Paraguay	es-py
Estonisch	et	Spanisch – Peru	es-pe

Gebietsschema	locale	Gebietsschema	locale
Färöisch	fo	Spanisch - Puerto Rico	es-pr
Farsi	fa	Spanisch - Standard	es
Finnisch	fi	Spanisch - Uruguay	es-uy
Französisch - Belgien	fr-be	Spanisch - Venezuela	es-ve
Französisch - Kanada	fr-ca	Sutu	sx
Französisch - Luxemburg	fr-lu	Thai	th
Französisch - Schweiz	fr-ch	Tschechisch	cs
Französisch - Standard	fr	Tsonga	ts
Gälisch - Schottland	gd	Türkisch	tr
Griechisch	el	Ukrainisch	uk
Hebräisch	he	Ungarisch	hu
Hindi	hi	Urdu - Pakistan	ur
Indonesisch	in	Vietnamesisch	vi
Isländisch	is	Weißrussisch	be
Italienisch - Schweiz	it-ch	Xhosa	xh
Italienisch - Standard	it	Zulu	zu

similar_text

similar_text

`similar_text` ermittelt eine Ähnlichkeit zwischen zwei Zeichenketten. Zurückgegeben wird die Anzahl übereinstimmender Zeichen absolut oder in Prozent (mit der Angabe *percent*).

◆ `int similar_text(string first, string second, double [percent]);`

soundex

soundex

`soundex` berechnet den Lautwert einer Zeichenkette für Ähnlichkeitsvergleiche.

◆ `string soundex(string str);`

Das Ergebnis ist eine Zeichenkette aus vier Zeichen, die mit einer Zahl beginnt und von drei Buchstaben gefolgt wird.



Beispiel

```
soundex("Euler") == soundex("Ellery") == 'E460';
soundex("Gauss") == soundex("Ghosh") == 'G200';
soundex("Knuth") == soundex("Kant") == 'H416';
soundex("Lloyd") == soundex("Ladd") == 'L300';
```

```
soundex("Lukasiewicz") == 'L222';  
soundex("Lissajous") == 'L222';
```

sprintf

`sprintf` gibt eine Zeichenkette formatiert zurück. Das Argument *format* bestimmt, wie die einzelnen Argumente *args* formatiert werden.

```
sprintf(string format, mixed [args]...);
```

Die folgende Beschreibung gilt auch für `printf`.

- Sie können ein Zeichen zur Auffüllung bestimmen, beispielsweise eine 0 oder ein Leerzeichen. Das Zeichen wird von einem einfachen Anführungszeichen angeführt.
- Sie können auch ein Ausrichtzeichen angeben, um die Ergebnisse links- oder rechtsbündig anzuordnen.
- Sie können eine Zahl angeben, welche die Anzahl der Stellen angibt.
- Optional kann auch die Anzahl der Dezimalstellen angegeben werden.
- Mit speziellen Zeichen definieren Sie die Zeichenbehandlung:
 - % Das Prozentzeichen zeigt an, dass keine Konvertierung erfolgt.
 - b behandelt den Wert als Ganzzahl (Integer).
 - c behandelt den Wert als Ganzzahl, interpretiert aber als ASCII und gibt das dazu passende Zeichen aus.
 - d behandelt den Wert als Ganzzahl (Integer) und gibt ihn als Dezimalzahl aus.
 - f behandelt den Wert als Gleitkommazahl (float).
 - o behandelt den Wert als Ganzzahl (Integer) und gibt ihn als Oktalzahl aus.
 - s gibt eine Zeichenkette aus.
 - x behandelt den Wert als Ganzzahl (Integer) und gibt ihn als Hexadezimalzahl aus; verwendet Kleinbuchstaben.
 - X behandelt den Wert als Ganzzahl (Integer) und gibt ihn als Hexadezimalzahl aus; verwendet Großbuchstaben.



Beispiel

Siehe auch `printf`, `number_format`.

```
$isodate=sprintf("%04d-%02d-%02d",$year,$month, $day);
$money1 = 68.75;
$money2 = 54.35;
$money = $money1 + $money2;
// echo $money zeigt an: "123.1";
$formatted = sprintf ("%01.2f", $money);
// echo $formatted zeigt an: "123.10"
```

strchr

strchr

`strchr` findet das erste Auftreten eines Zeichens *needle* in der Zeichenkette *haystack*. Diese Funktion ist ein Alias für `strstr`.



```
string strchr(string haystack, string needle);
```

strcmp

strcmp

`strcmp` vergleicht zwei Zeichenketten auf Binärebene. Der Vergleich berücksichtigt daher Groß- und Kleinschreibung. Wenn *str1* kleiner als *str2* ist, wird `-1` zurückgegeben, bei Gleichheit `0`, sonst `+1`.



```
int strcmp(string str1, string str2);
```

Siehe auch `ereg`, `substr` und `strstr`.

strcspn

strcspn

Die Funktion `strcspn` gibt die Anzahl der Zeichen in *str1* zurück, die *nicht* in *str2* enthalten sind.



```
int strcspn(string str1, string str2);
```

Siehe auch `strspn`.

strip_tags

strip_tags

`strip_tags` entfernt alle PHP- und HTML-Tags aus einer Zeichenkette. Ein Laufzeitfehler tritt auf, wenn sinnlose oder unvollkommene Tags gefunden werden.



```
string strip_tags(string str);
```

Siehe auch `fgetss`.

stripslashes

`stripslashes` kehrt die Funktion `addslashes` um und entfernt die Backslashes vor Sonderzeichen.

```
string stripslashes(string str);
```

Siehe auch `addslashes`.

stripslashes**strlen**

`strlen` ermittelt die Länge einer Zeichenkette *str*.

```
int strlen(string str);
```

strlen**strrpos**

`strrpos` findet das letzte Auftreten einer Zeichenkette *charneedle* in der Zeichenkette *haystack*. Beachten Sie, dass *charneedle* nur ein Zeichen sein darf, keine Zeichenkette. Wenn dennoch eine Zeichenkette übergeben wird, interpretiert die Funktion nur das erste Zeichen. Wird keine Übereinstimmung gefunden, gibt die Funktion `FALSE` zurück.

```
int strrpos(string haystack, charneedle);
```

Siehe auch `strpos`, `strchr`, `substr` und `strstr`.

strrpos**strpos**

`strpos` findet das erste Auftreten der Zeichenkette *needle* in der Zeichenkette *haystack*. Der Vergleich beginnt an der mit *offset* bestimmten Position. Wird keine Übereinstimmung gefunden, gibt die Funktion `FALSE` zurück.

```
int strpos(string haystack, string needle, int[offset]);
```

Wird *offset* verwendet, ist die Angabe der gefundenen Position relativ. Die Funktion verarbeitet auch Zeichenketten, im Gegensatz zu `strrpos`.

Siehe auch `strrpos`, `strchr`, `substr` und `strstr`.

strpos**strchr**

`strchr` findet das letzte Vorkommen eines Zeichens *needle* in einer Zeichenkette *haystack*. Wird keine Übereinstimmung gefunden, gibt die Funktion `FALSE` zurück.

```
string strchr(string haystack, string needle);
```

strchr



```
// gibt das letzte Verzeichnis in $PATH zurück
$dir = substr( strrchr( $PATH, ":" ), 1 );
// alles nach dem Zeilenumbruch
$text = "Line 1\nLine 2\nLine 3";
$last = substr( strrchr( $text, 10 ), 1 );
```

Siehe auch `substr` und `strstr`.

strrev

strrev

`strrev` dreht eine Zeichenkette um.



```
string strrev(string string);
```

strspn

strspn

`strspn` ermittelt die Anzahl der Zeichen in `str1`, die mit `str2` übereinstimmen.



```
int strspn(string str1, string str2);
```

Siehe auch `strcspn`.

strstr

strstr

`strstr` findet das erste Auftreten einer Zeichenkette *needle* in der Zeichenkette *haystack*. Wird keine Übereinstimmung gefunden, gibt die Funktion `FALSE` zurück. Wenn *needle* kein Zeichen ist, wird der Wert als Zahl interpretiert, dann als ASCII-Wert und daraus ein Zeichen ermittelt.



```
string strstr(string haystack, string needle);
```

Siehe auch `strrchr`, `substr` und `ereg`.

strtok

strtok

`strtok` zerlegt eine Zeichenkette anhand eines Trennzeichens in mehrere Teile (Token).



```
string strtok(string arg1, string arg2);
```



```
$string = "Dies ist ein einfaches Beispiel";
$tok = strtok($string, " ");
while($tok) {
    echo "Wort = $tok<br>";
    $tok = strtok(" ");
}
```

Die zu zerlegende Zeichenkette muss nur beim ersten Aufruf angegeben werden. Alle folgenden Aufrufe benötigen nur das Trenn-

Trennzeichen. Im Gegensatz zu `explode` wird kein Array erzeugt, sondern immer nur das nächste Token zurückgegeben. Es ist auch erlaubt, mehrere Trennzeichen anzugeben, die dann ebenso in der Zeichenkette enthalten sein müssen.

Siehe auch `split` und `explode`.

strtolower

`strtolower` wandelt die Zeichen der Zeichenkette `str` in Kleinbuchstaben um.

```
string strtolower(string str);
```

Siehe auch `strtoupper` und `ucfirst`.

strtoupper

`strtoupper` gibt eine Zeichenkette in Großbuchstaben zurück. Umlaute werden nicht umgewandelt.

```
string strtoupper(string string);
```

Siehe auch `strtolower` und `ucfirst`.

str_replace

`str_replace` ersetzt alle Vorkommen von `needle` in `haystack` mit `str`. Die Funktion ersetzt auf der Basis eines Binärvergleichs.

```
string str_replace(string needle, string str, string haystack);
```

Die Funktion ist signifikant schneller als `ereg_replace`.

```
$bodytag = str_replace("%body%", "black",  
                      "<body text=%body%>");
```

Siehe auch `ereg_replace`.

strtr

`strtr` tauscht Zeichen anhand einer Austauschliste aus. Die Zeichenkette `str` wird durchsucht, jedes Auftreten eines Zeichens in `from` wird durch das an gleicher Position befindliche Zeichen in `to` ersetzt.

```
string strtr(string str, string from, string to);
```

Wenn `to` mehr Zeichen als `from` enthält, werden die übrigen Zeichen ignoriert.

strtolower



strtoupper



str_replace



Beispiel

strtr





```
$addr = strstr($addr, "ääö", "ao");
```

Siehe auch `ereg_replace`.

substr

substr

`substr` gibt den Teil einer Zeichenkette *string* zurück, beginnend an der Position *start* und mit *length* Zeichen.



```
string substr(string string, int start, int [length]);
```

Der Parameter *start* kann negativ sein, dann beginnt die Zählung von hinten.



```
$rest = substr("abcdef", 1); // ergibt "bcdef"
$rest = substr("abcdef", 1, 3); // ergibt "bcd"
$rest = substr("abcdef", -1); // ergibt "f"
$rest = substr("abcdef", -2); // ergibt "ef"
$rest = substr("abcdef", -3, 1); // ergibt "d"
$rest = substr("abcdef", 1, -1); // ergibt "bcde"
```

Siehe auch `strchr` und `ereg`.

trim

trim

`trim` entfernt Leerzeichen vom Anfang und vom Ende einer Zeichenkette.



```
string trim(string str);
```

Siehe auch `chop` und `ltrim`.

ucfirst

ucfirst

`ucfirst` wandelt das erste Zeichen einer Zeichenkette in einen Großbuchstaben.



```
string ucfirst(string str);
```

Siehe auch `strtoupper` und `strtolower`.

ucwords

ucwords

`ucwords` wandelt das erste Zeichen jedes Wortes einer Zeichenkette in einen Großbuchstaben.



```
string ucwords(string str);
```

Siehe auch `strtoupper`, `strtolower` und `ucfirst`.

Funktionen für Rechtschreibprüfung

Die Bibliotheken für die Rechtschreibprüfung können Sie von folgender Adresse beziehen:

- <http://pspell.sourceforge.net>

Kompilieren Sie PHP mit `--with-pspell[=dir]`.

»PSpell« steht für »Portable Spell Checker Interface Library«.

Die alten `aspell`-Funktionen unterstützen nicht die neuen Bibliotheken und sollten nicht weiter verwendet werden.

pspell_new

Diese Funktion lädt ein neues Verzeichnis.

```
int pspell_new(string language [, string spelling
               [, string jargon [, string encoding
               [, int mode]]]])
```

pspell_new



`pspell_new` öffnet ein neues Verzeichnis und gibt ein Handle zurück, das die anderen Funktionen verwenden können. Der Parameter *language* ist der Buchstabencode für eine Sprache nach ISO 639. Zusätzlich kann eine Sprachvariante nach ISO 3166 nach einem Minuszeichen oder Unterstrich angefügt werden. *spelling* gibt einen Sprachtyp an, wenn eine Sprache mehrere Varianten kennt. So gibt es für Englisch, Code »en«, die Varianten »american« und »british«. Der Parameter *jargon* unterscheidet zwei verschiedene Wörterlisten, wenn die übrigen Parameter diese nicht unterscheiden können. Mit *encoding* wird die Art der Zeichencodierung angegeben. Gültige Werte sind »utf-8«, »iso8859-*«, »koi8-r«, »viscii«, »cp1252«, »machine unsigned 16«, »machine unsigned 32«.

Der Parameter *mode* kann der folgenden Liste entnommen werden:

- `PSPELL_FAST`. Schneller Mode, wenige Vorschläge
- `PSPELL_NORMAL`. Normale Geschwindigkeit
- `PSPELL_BAD_SPELLERS`. Langsam, sehr viele Vorschläge
- `PSPELL_RUN_TOGETHER`. Nimmt zusammengeschiedene Wörter als korrekt an, wenn die Teile im Wörterbuch gefunden wurden. So wäre »funhouse« korrekt, weil »fun« und »house« korrekt sind.

Der letzte Parameter ist ein Bitwert, die ersten drei Konstanten schließen sich jedoch gegenseitig aus. Ansonsten kombinieren Sie mit | (oder).

pspell_check

pspell_check Mit dieser Funktion wird ein Wort im Wörterverzeichnis nachgeschlagen.



boolean **pspell_check**(int dictionary_link, string word)

Prüft ein Wort im Verzeichnis und gibt TRUE zurück, wenn es gefunden wurde.

```
$pspell_link = pspell_new ("en");
if (pspell_check ($pspell_link, "housse")) {
    echo "Dieses Wort wurde richtig geschrieben";
} else {
    echo "Dieses Wort wurde falsch geschrieben";
}
```

pspell_suggest

pspell_suggest Diese Funktion gibt einen Vorschlag für eine Schreibweise zurück.



array **pspell_suggest**(int dictionary_link, string word)

Das folgende Beispiel zeigt die Anwendung:

```
$pspell_link = pspell_new ("en");
if (!pspell_check($pspell_link, "testt")) {
    $suggestions = pspell_suggest($pspell_link, "testt");
    for ($i=0; $i < count ($suggestions); $i++) {
        echo "Mögliche Schreibweisen: ".$suggestions[$i]."<br>";
    }
}
```

GNU Gettext Sprachunterstützung

Diese Funktionssammlung unterstützt den Aufbau von Applikationen mit mehreren Sprachen. Detaillierte Informationen finden Sie unter der folgenden Adresse:

- <http://www.gnu.org/software/gettext/gettext.html>

bindtextdomain

bindtextdomain Diese Funktion bindet eine Domain an eine bestimmte Sprachdatei.



string **bindtextdomain**(string domain, string directory)

dcgettext

Diese Funktion überschreibt die aktuelle Domainzuordnung für eine einzige Abfrage. Außerdem kann die Kategorie ausgewählt werden.

dcgettext

```
string dcgettext(string domain, string message, int category)
```

**dgettext**

Diese Funktion überschreibt die aktuelle Domainzuordnung für eine einzige Abfrage.

dgettext

```
string dgettext(string domain, string message)
```

**gettext**

Diese Funktion übersetzt eine Zeichenkette anhand der Übersetzungstabelle und der voreingestellten Domain. Wird keine Übersetzung gefunden, gibt die Funktion die Anfrage zurück.

gettext

```
string gettext(string message)
```



```
<?php
putenv("LANG=de");
bindtextdomain ("myPHPApp", "./locale");
textdomain ("myPHPApp");
echo gettext("Welcome to My PHP Application");
?>
```



Beispiel

textdomain

Diese Funktion stellt die Standarddomain ein, die verwendet wird, wenn `gettext` aufgerufen wird. Die Domain muss zuvor mit der Übersetzungstabelle verbunden worden sein.

textdomain

```
int textdomain([string library])
```



Wenn die Funktion `textdomain` ohne Parameter aufgerufen wird, gibt sie die aktuellen Einstellungen zurück, ohne diese zu ändern.

Funktionen für reguläre Ausdrücke

Die Standardfunktionen verwenden die POSIX 1003.2-Syntax. Die Perlfunktionen verwenden eine erweiterte Syntax. Eine ausführliche Beschreibung der Standardfunktionen finden Sie in Kapitel 3, die komplexeren Perl-Funktionen werden in Kapitel **Fehler! Ver-**

weisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden. beschrieben.



```
ereg("abc",$string);
// Gibt TRUE zurück, wenn irgendwo »abc« steht
ereg("^abc",$string);
// Gibt TRUE zurück, wenn am Anfang »abc« steht
ereg("abc$", $string);
// Gibt TRUE zurück, wenn am Ende »abc« steht
eregi("(ozilla.[23]|MSIE.3)", $HTTP_USER_AGENT);
// Gibt TRUE zurück, wenn es Netscape ist
ereg("([[:alnum:]]+) ([[:alnum:]]+) ([[:alnum:]]+)",
    $string, $regs);
/* Stellt drei durch Leerzeichen getrennte Worte in die Arrays
$regs[1], $regs[2] and $regs[3]. */
$string = ereg_replace("^", "<BR>", $string);
// Setzt <BR> an den Beginn der Zeichenkette
$string = ereg_replace("$", "<BR>", $string);
// Setzt <BR> an das Ende der Zeichenkette
$string = ereg_replace("\n", "", $string);
// Entfernt alle Zeilenumbrüche
```

ereg

ereg ereg durchsucht eine Zeichenkette *string* nach einem regulären Ausdruck *pattern*.

◆ int **ereg**(string pattern, string string, array [regs]);

Die Suche berücksichtigt Groß- und Kleinschreibung. Mehrere Gruppen werden im Array *regs* übergeben. Wenn das Suchmuster gefunden wurde, wird TRUE zurückgegeben.

ereg_replace

ereg_replace ereg_replace sucht und ersetzt eine Zeichenkette nach einem regulären Ausdruck.

◆ string **ereg_replace**(string pattern, string replacement, string string);

Die Funktion durchsucht *string* nach dem regulären Ausdruck *pattern* und ersetzt gefundene Teile durch *replacement*.

Die veränderte Zeichenkette wird zurückgegeben, das Original wird aber nicht verändert. Wenn Gruppen enthalten sind, kann darauf mit dem Code `\\X` reagiert werden, wobei *X* für eine der Gruppen 0 bis 9 steht.

eregi

eregi durchsucht eine Zeichenkette *string* nach einem regulären Ausdruck *pattern*.

```
int ereg(string pattern, string string, array [regs]);
```

Die Suche berücksichtigt Groß- und Kleinschreibung nicht. Mehrere Gruppen werden im Array *regs* übergeben. Wenn das Suchmuster gefunden wurde, wird TRUE zurückgegeben.

eregi_replace

eregi_replace sucht und ersetzt eine Zeichenkette nach einem regulären Ausdruck.

```
string ereg_replace(string pattern, string replacement,  
                    string string);
```

Die Suche berücksichtigt Groß- und Kleinschreibung nicht. Die Funktion durchsucht *string* nach dem regulären Ausdruck *pattern* und ersetzt gefundene Teile durch *replacement*.

Die veränderte Zeichenkette wird zurückgegeben, das Original wird aber nicht verändert. Wenn Gruppen enthalten sind, kann darauf mit dem Code `\\X` reagiert werden, wobei X für eine der Gruppen 0 bis 9 steht.

split

split zerlegt eine Zeichenkette und gibt die Fragmente als Array zurück. Die Splitfunktion ist ein regulärer Ausdruck.

```
array split(string pattern, string string, int [limit]);
```

Siehe auch explode und implode.

sql_regcase

sql_regcase überführt ein Suchmuster in einen regulären Ausdruck, der unabhängig von Groß- und Kleinschreibung ist.

```
string sql_regcase(string string);
```

```
echo sql_regcase( "Foo bar" );
```

Dieses Beispiel gibt folgenden Text aus:

```
[Ff][Oo][Oo][ ] [Bb][Aa][Rr]
```

eregi**eregi_replace****split****sql_regcase**

Beispiel

preg_match

preg_match

`preg_match` durchsucht eine Zeichenkette nach einem regulären Ausdruck.



```
int preg_match(string pattern, string subject, array [matches]);
```

Die Zeichenkette *subject* wird anhand des Suchmusters *pattern* durchsucht. Die Funktion gibt TRUE zurück, wenn die Suche erfolgreich war. Wenn Gruppen gebildet wurden, werden die Ergebnisse in dem Array *matches* zurückgegeben. *matches[0]* enthält die gesamte Zeichenkette, *matches[1]* die erste Gruppe usw.



```
// Ermittelt die Seitenzahl in einem Satz
if (preg_match("/page\s+(\d+)/i",
              "Gehe zu Seite #9.", $parts))
    print "Next page is $parts[1]";
else
    print "Page not found.";
```

Siehe auch `preg_match_all`, `preg_replace` und `preg_split`.

preg_match_all

preg_match_all

`preg_match_all` sucht global nach einem Suchmuster.



```
int preg_match_all(string pattern, string subject,
                  array matches, int [order]);
```

Die Zeichenkette *subject* wird anhand des Suchmusters *pattern* durchsucht. Die Funktion gibt TRUE zurück, wenn die Suche erfolgreich war. Wenn Gruppen gebildet wurden, werden die Ergebnisse in dem Array *\$matches* zurückgegeben. *\$matches[0]* enthält die gesamte Zeichenkette, *\$matches[1]* die erste Gruppe usw.

Wurde ein Muster gefunden, wird die Suche fortgesetzt. *order* bestimmt, wie die Ergebnisse ausgegeben werden:

- `PREG_PATTERN_ORDER` verhält sich wie `preg_match` (Standard).
- `PREG_SET_ORDER`. Diese Option sortiert die Ergebnisse so, dass *\$matches[0]* den ersten Ergebnissatz enthält, *\$matches[1]* den zweiten usw.



```
preg_match_all("|<[^>]+>(.*</[^>]+>|U",
              "<b>Beispiel: </b><div align=left>
              Ein Test</div>",
              $out,
              PREG_PATTERN_ORDER);
print $out[0][0].", ".$out[0][1]."\n";
print $out[1][0].", ".$out[1][1]."\n";
```

Das Beispiel hat folgende Ausgabe:

```
<b>Beispiel: </b>, <div align=left>Ein Test</div>
Beispiel: , Ein Test
```

```
preg_match_all("|<[^>]+>(.*?)</[^>]+>|U",
    "<b>Beispiel: </b><div align=left>
    Ein Test</div>",
    $out,
    PREG_SET_ORDER);
print $out[0][0].", ".$out[0][1]."\n";
print $out[1][0].", ".$out[1][1]."\n"
```



Das Beispiel hat folgende Ausgabe:

```
<b>Beispiel: </b>, Beispiel:
<div align=left>Ein Test</div>, Ein Test
```

```
// ermittelt gruppierte Telefonnummern
preg_match_all("/\(?(\\d{3})?\\)?(?!(1)[\\-\\s])\\d{3}-\\d{4}/x",
    "Call 555-1212 or 1-800-555-1212", $phones);
```



preg_replace

preg_replace sucht nach einem regulären Ausdruck und ersetzt die Fundstelle.

preg_replace

```
mixed preg_replace(mixed pattern, mixed replacement,
    mixed subject);
```



Die Zeichenkette *subject* wird anhand des Suchmusters *pattern* durchsucht und an den Fundstellen durch *replacement* ersetzt.

Innerhalb der Ersatzzeichenkette kann mit \\n auf die Gruppen im Muster Bezug genommen werden, wobei n eine Zahl zwischen 0 und 99 ist, \\0 bezieht sich auf die gesamte Zeichenkette. Der Beginn einer Gruppe wird mit einer runden Klammer eingeleitet.

Alle Parameter können auch Arrays sein. Die Suche wird dann in jedem Element des Arrays ausgeführt und ein Array wird zurückgegeben.

Die Option /e behandelt den Code in *replacement* als PHP-Code. Ist der Code fehlerhaft, wird ein Parser-Fehler ausgegeben.

```
$patterns = array("/(19|20\\d{2})-(\\d{1,2})-(\\d{1,2})/",
    "/^\\s*{\\(\\w+)}\\s*="/);
$replace = array("\\3/\\4/\\1", "$\\1 =");
print preg_replace($patterns,
    $replace,
    "{startDate} = 1999-5-27");
```



Dieses Beispiel zeigt Folgendes an:



Beispiel

```
$startDate = 5/27/1999
preg_replace("/(<\/?)(\w+)([>]*>)/e",
            "'\1'.strtoupper('\2').'\3'",
            $html_body);
```

Dieses Beispiel macht den Inhalt aller HTML-Tags in der Zeichenkette zu Großbuchstaben.

Siehe auch `preg_match`, `preg_match_all` und `preg_split`.

preg_split

preg_split

`preg_split` zerlegt eine Zeichenkette anhand eines regulären Ausdrucks.



```
array preg_split(string pattern, string subject, int [limit]);
```

Zurückgegeben wird ein Array mit den Teilen, die maximale Anzahl kann mit *limit* eingeschränkt werden.



Beispiel

```
$keywords = preg_split("/[\s,]+/",
                    "hypertext, programming");
```

Das Beispiel zerteilt die Zeichenkette anhand des Kommas.

Siehe auch `preg_match`, `preg_match_all` und `preg_replace`.

preg_quote

preg_quote

`preg_quote` markiert alle Sonderzeichen, die in regulären Ausdrücken eine Bedeutung haben so, dass sie ihre Bedeutung verlieren. Dazu wird ein Backslash davor gesetzt.



```
string preg_quote(string str);
```

Die betroffenen Zeichen sind:

- `. \ + * ? [^] $ () { } = ! < > | :`

preg_grep

preg_grep

`preg_grep` gibt in einem Array Teile der Zeichenkette zurück, für die eine Übereinstimmung mit dem regulären Ausdruck gefunden wurde.



```
array preg_grep(string pattern, array input);
```

Durchsucht wird das Array *input* elementweise.



Beispiel

Das folgende Beispiel sucht alle Gleitkommazahlen in einem Array:

```
preg_grep("/^(\\d+)?\\.\\d+$/", $array);
```

Die Musterschalter der regulären Ausdrücke

Die folgende Liste zeigt alle PCRE-Optionen (Perl Compatible Regular Expressions).

- `i` (`PCRE_CASELESS`). Ignoriert Groß- und Kleinschreibung.
- `m` (`PCRE_MULTILINE`). Normalerweise wird der Suchausdruck als eine Zeichenkette betrachtet, auch wenn es mehrere Zeilen sind. Das Startzeichen `^` wirkt nur auf den absoluten Anfang, `$` nur auf das absolute Ende. Wird diese Option gesetzt, gilt jeder Zeilenanfang und jedes Zeilenende als Start und Ende für Suchmuster. Auf Zeichenketten, die keine `»\n«` enthalten, hat dies keinen Einfluss.
- `s` (`PCRE_DOTALL`). Wenn diese Option gesetzt ist, erfüllt der Punkt `.` alle Zeichen, auch `»\n«`.
- `x` (`PCRE_EXTENDED`). Wird diese Option gesetzt, werden alle Whitespaces¹ ignoriert.
- `e`. Mit dieser Option werden die Angaben in der Ersatzzeichenkette bei `preg_replace` als PHP-Code interpretiert.
- `A` (`PCRE_ANCHORED`). Schränkt die Suche auf den Anfang der Zeichenkette ein.
- `E` (`PCRE_DOLLAR_ENDONLY`). Schränkt die Suche auf das Ende der Zeichenkette ein. Wird ignoriert, wenn gleichzeitig `m` genutzt wird.
- `S`. Wird ein Muster mehrfach verwendet, kann die Ausführungsgeschwindigkeit gesteigert werden, indem diese Option eingesetzt wird. PHP speichert den übersetzten Ausdruck dann zwischen.
- `U` (`PCRE_UNGREEDY`). Bestimmt die Arbeitsweise der Musteranweisungen bezüglich der Mehrfachauswahl.
- `X` (`PCRE_EXTRA`). Schaltet zusätzliche Funktionen, die derzeit nicht verwendet werden, ein.

¹ Whitespaces sind: Leerzeichen, Tabulatoren, Zeilenumbrüche.

Datums-, Kalender- und Zeitfunktionen

checkdate

checkdate

checkdate prüft ein Datum auf Gültigkeit.



```
int checkdate(int month, int day, int year);
```

Gibt TRUE zurück, wenn das Datum gültig ist, sonst FALSE. Das Datum muss folgende Kriterien erfüllen:

- *year* ist zwischen 0 und 32 767 (einschließlich).
- *month* ist zwischen 1 und 12 (einschließlich).
- *day* ist zwischen 1 und der durch den Monat bestimmten Anzahl. Schaltjahre werden beachtet.

date

date

date formatiert eine Zeichenkette zur Ausgabe von Datum und Uhrzeit.



```
string date(string format, int [timestamp]);
```

Die Ausgabe wird anhand eines Musters erstellt, das folgende Platzhalter verwendet:

- a – »am« oder »pm«
- A – »AM« oder »PM«
- d – Tag im Monat, zwei Ziffern mit führender Null: »01« bis »31«
- D – Tag der Woche mit drei Buchstaben, beispielsweise »Fri«
- F – Ausgeschriebener Monat: »January«
- h – Stunde im 12-Stunden-Format: »01« bis »12«
- H – Stunde im 24-Stunden-Format: »00« bis »23«
- g – Stunde im 12-Stunden-Format ohne führende Null: »1« bis »12«
- G – Stunde im 24-Stunden-Format ohne führende Null: »00« bis »23«
- i – Minuten: »00« bis »59«
- j – Tag des Monats ohne führende Null: »1« bis »31«

- l (kleines 'l') – Ausgeschriebener Tag der Woche: »Friday«
- L -- Boolescher Wert für das Schaltjahr: "0" oder "1"
- m – Monat: »01« bis »12«
- n – Monat ohne führende Null: »1« bis »12«
- M – Monat mit drei Buchstaben: »Jan«
- s – Sekunden: »00« bis »59«
- S – Suffix der englischen Ordnungszahlen: »th«, »nd« usw.
- t – Anzahl der Tage des Monats: »28« bis »31«
- U – Sekunden der UNIX-Epoche (seit 1.1.1970)
- w – Tag der Woche, beginnend mit »0« (Sonntag) bis »6« (Samstag)
- Y – Vierstelliges Jahr: »1999«
- y – Zweistelliges Jahr: »99«
- z – Tag im Jahr: »0« bis »365«
- Z – Offset der Zeitzonen gegen GTM in Minuten: »-43 200« bis »43 200«

Alle anderen Zeichen werden unverändert ausgegeben.

```
print (date("%l dS of F Y h:i:s A"));
print ("July 1, 2000 is on a " .
      date("%l", mktime(0,0,0,7,1,2000)));
```



```
// Nutzung mit mktime, um Daten in der
// Zukunft oder Vergangenheit zu berechnen
$tomorrow = mktime(0,0,0,date("m") ,
                  date("d")+1,date("Y"));
$lastmonth = mktime(0,0,0,date("m")-1,date("d"),
                  date("Y"));
$nextyear = mktime(0,0,0,date("m"),
                  date("d", date("Y")+1));
```



Um Datumsangaben in anderen Sprachen zu erhalten, nutzen Sie die Funktionen `setlocale` und `strftime`.

Siehe auch `gmdate` und `mktime`.

strftime**strftime**

strftime formatiert eine Datumsangabe nach den Angaben der Region.



```
string strftime(string format, int timestamp);
```

Ausgaben von Monats- oder Wochentagsnamen werden nach der mit `setlocale` eingestellten Sprachangabe formatiert.

Folgende Zeichen sind für die Steuerung der Anzeige zulässig:

- %a – Abkürzung des Wochentagsnamens
- %A – Voller Wochentagsname
- %b – Abkürzung des Monats
- %B – Voller Monatsname
- %c – Standardwerte des Betriebssystems für Datum und Uhrzeit
- %d – Tag im Monat von 00 bis 31
- %H – Stunde im 24-Stunden-Format
- %I – Stunde im 12-Stunden-Format
- %j – Tag im Jahr von 001 bis 366
- %m – Monat als Zahl 01 bis 12
- %M – Minute
- %p – am oder pm
- %S – Sekunde
- %U – Wochennummer, die erste Woche beginnt am ersten Sonntag im Jahr
- %W – Wochennummer, die erste Woche beginnt am ersten Montag im Jahr
- %w – Tag der Woche als Zahl, der Sonntag ist 0
- %x – Zeit, abhängig von den lokalen Einstellungen
- %X – Datum, abhängig von den lokalen Einstellungen
- %y – Jahr, zweistellig

- %Y – Jahr, vierstellig
- %Z – Zeitzone
- %% – Das %-Zeichen

```
setlocale ("LC_TIME", "C");
print(strftime("%A in Finnish is "));
setlocale ("LC_TIME", "fi");
print(strftime("%A, in French "));
setlocale ("LC_TIME", "fr");
print(strftime("%A and in German "));
setlocale ("LC_TIME", "DE_de");
print(strftime("%A.\n"));
```



Siehe auch `setlocale` und `mktime`. Die Abkürzungen für die Gebietsschemen werden unter `setlocale` beschrieben.

gmstrftime

`gmstrftime` formatiert eine GMT-Zeitangabe. Die Parameter entsprechen `strftime`.

gmstrftime

```
string gmstrftime(string format, int timestamp);

setlocale ('LC_TIME', 'en_US');
echo strftime ("%b %d %Y %H:%M:%S",
               mktime(20,0,0,12,31,98))."\n";
ho gmstrftime ("%b %d %Y %H:%M:%S",
               mktime(20,0,0,12,31,98))."\n";
```



Siehe auch `strftime`.

getdate

`getdate` ermittelt Zeit- und Datumsinformationen.

getdate

```
array getdate(int timestamp);
```



Die Funktion gibt ein assoziatives Array mit Datums- und Zeitangaben zurück. Die Elemente des Arrays werden folgendermaßen indiziert:

- »seconds«. Sekunden
- »minutes«. Minuten
- »hours«. Stunden
- »mday«. Tag im Monat
- »wday«. Tag in der Woche, numerisch

- »mon«. Monat, numerisch
- »year«. Jahr, numerisch
- »yday«. Tag im Jahr, numerisch
- »weekday«. Wochentagsname, ausgeschrieben
- »month«. Monatsname, ausgeschrieben

gettimeofday

gettimeofday gettimeofday ermittelt die aktuelle Zeit.

◆ array **gettimeofday**(void);

Die Funktion gibt ein assoziatives Array mit Zeitangaben zurück. Die Elemente des Arrays werden folgendermaßen indiziert:

- »sec«. Sekunden
- »usec«. Mikrosekunden
- »minuteswest«. Minuten westlich von Greenwich
- »dsttime«. Typ der Sommerzeitkorrektur

gmdate

gmdate gmdate formatiert eine GTM-Zeitangabe.

◆ string **gmdate**(string format, int timestamp);

Die Funktion entspricht date, nur dass als Zeitangabe immer GMT (Greenwich Mean Time) verwendet wird.



Beispiel

```
echo date( "M d Y H:i:s", mktime(0,0,0,1,1,1998) );
echo gmdate( "M d Y H:i:s", mktime(0,0,0,1,1,1998) );
```

Siehe auch date, mktime und gmmktime.

localtime

localtime Die Funktion gibt ein Array mit Zeitinformationen über die Zeit zurück, die der übergebene Zeitstempel anzeigt. Wird kein Parameter übergeben, wird die aktuelle Server-Zeit eingesetzt.

◆ array **localtime**([int timestamp [, bool is_associative]])

Es wird ein numerisches Array erzeugt, wenn der zweite Parameter *is_associative* nicht angegeben wurde oder auf FALSE steht. An-

dernfalls wird ein assoziatives Array erzeugt. Die Schlüssel des assoziativen Arrays haben folgende Bedeutung:

- "tm_sec". Sekunden
- "tm_min". Minuten
- "tm_hour". Stunde
- "tm_mday". Tag des Monats
- "tm_mon". Monat im Jahr
- "tm_year". Jahr, nicht Jahr-2000-kompatibel
- "tm_wday". Tag in der Woche
- "tm_yday". Tag im Jahr
- "tm_isdst". Sommerzeit

strtotime

Diese Funktion erwartet ein Datum mit ausgeschriebenen Namen in englischer Sprache. Zurückgegeben wird der Unix-Zeitstempel.

```
int strtotime(string time [, int now])
```

Das folgende Beispiel zeigt eine typische Zeichenkette:

```
echo strtotime ("10 march 2000");
```

Siehe auch `date`, `time`.

strtotime



mktime

`mktime` ermittelt den Unix-Zeitstempel für ein gegebenes Datum.

```
int mktime(int hour, int minute, int second,
           int month, int day, int year, int [is_dst]);
```

Der Zeitstempel entsteht durch Zählung der Sekunden seit Beginn der Unix-Epoche, dem 1.1.1970. Mit der Funktion `mktime` können leicht Datumsberechnungen angestellt werden.

```
echo date( "M-d-Y", mktime(0,0,0,12,32,1997) );
echo date( "M-d-Y", mktime(0,0,0,13,1,1997) );
echo date( "M-d-Y", mktime(0,0,0,1,1,2001) );
```

Siehe auch `date` und `time`.

mktime



gmmktime

gmmktime gmmktime ermittelt den Zeitstempel nach GMT.

◆
`int gmmktime(int hour, int minute, int second,
 int month, int day, int year,
 int [is_dst]);`

Siehe auch `mktime`.

time

time `time` gibt den aktuellen UNIX-Zeitstempel zurück.

◆
`int time(void);`

Entspricht `mktime` für die aktuelle Systemzeit.

Siehe auch `date`.

microtime

microtime `microtime` ermittelt den Zeitstempel der Systemzeit mit Mikrosekunden.

◆
`string microtime(void);`

Die Funktion gibt eine Zeichenkette der Art »Mikrosekunden Sekunden« zurück; die Zeitwerte sind also durch ein Leerzeichen getrennt.

Siehe auch `time`.

Kalenderfunktionen

Die Kalenderfunktionen benötigen die Bibliothek `DL/CALENDER`. Mehr Informationen zum Umgang mit Kalendern finden Sie unter der Adresse:

- <http://genealogy.org/~scottlee/cal-overview.html>

jdtogregorian

jdtogregorian `jdtogregorian` konvertiert eine julianische Tageszählung in ein gregorianisches Datum der Form »Monat/Tag/Jahr«.

◆
`string jdtogregorian(int julianday);`

gregoriantojd

gregoriantojd konvertiert ein gregorianisches Datum in eine julianische Tageszählung.

```
int gregoriantojd(int month, int day, int year);
```

Der gregorianische Kalender reicht von 4717 vor Chr. bis 9999 n. Chr. Allerdings wurde der Kalender erst am 15. Oktober 1582 eingeführt. Einige Länder gingen erst viel später auf diesen Kalender über, beispielsweise England 1752, Rußland 1918 und Griechenland 1923.

```
<?php
$jd = GregorianToJD(10,11,1970);
echo("$jd\n");
$gregorian = JDToGregorian($jd);
echo("$gregorian\n");
?>
```



Beispiel

jdtojulian

jdtojulian konvertiert die julianische Tageszählung in ein julianisches Datum.

```
string jdtojulian(int julianday);
```

juliantojd

juliantojd konvertiert ein julianisches Datum in die julianische Tageszählung.

```
int juliantojd(int month, int day, int year);
```

jdtojewish

jdtojewish konvertiert die julianische Tagszählung in den jüdischen Kalender.

```
string jdtojewish(int julianday);
```

jewishtojd

jewishtojd konvertiert ein jüdisches Datum in die julianische Tagszählung.

```
int jewishtojd(int month, int day, int year);
```

gregoriantojd

jdtojulian

juliantojd

jdtojewish

jewishtojd

jdtofrrench

jdtofrrench jdtofrrench konvertiert die julianische Tageszählung in den Kalender der französischen Republik.

◆ string **jdtofrrench**(int month, int day, int year);

frenchtojd

frenchtojd frenchtojd konvertiert den Kalender der französischen Republik in die julianische Tageszählung.

◆ int **frenchtojd**(int month, int day, int year);

Die Funktion kann Daten nach dem gregorianischen Kalender vom 22. September 1792 bis zum 22. September 1806 verarbeiten, Das ist das Jahr 1 bis 14 der Revolution. Beachten Sie aber, dass Sie diese Daten zuvor in die julianische Tageszählung umwandeln müssen.

jdmonthname

jdmonthname jdmonthname gibt den Monatsnamen zurück.

◆ string **jdmonthname**(int julianday, int mode);

mode kann folgende Werte annehmen:

- 0. Gregorianisch abgekürzt
- 1. Gregorianisch
- 2. Julianisch abgekürzt
- 3. Julianisch
- 4. Jüdisch
- 5. Französische Republik

jddayofweek

jddayofweek jddayofweek gibt den Wochentag zurück.

◆ mixed **jddayofweek**(int julianday, int mode);

mode kann folgende Werte annehmen:

- 0. Tag als Zahl (0=Sonntag, 1=Montag usw.)
- 1. Wochentag, englisch-gregorianisch

- 2. Abkürzung des Wochentags, englisch-gregorianisch

easter_date

easter_date ermittelt den Zeitstempel des Ostersonntags 00:00 Uhr für das angegebene Jahr.

easter_date

```
int easter_date(int year);

echo date( "M-d-Y", easter_date(1999) );
/* "Apr-04-1999" */
echo date( "M-d-Y", easter_date(2000) );
/* "Apr-23-2000" */
echo date( "M-d-Y", easter_date(2001) );
/* "Apr-15-2001" */
```



Die Funktion kann keine Osterdaten außerhalb der Unix-Epoche ermitteln. Siehe easter_days für Berechnungen vor 1970 oder nach 2037.

easter_days

easter_days gibt die Anzahl der Tage zwischen Ostern und dem 21. März des angegebenen Jahres zurück.

easter_days

```
int easter_days(int year);

echo easter_days(1999);      /* 14, d.h. 4. April */
echo easter_days(1492);     /* 32, d.h. 22. April */
echo easter_days(1913);     /* 2, d.h. 23. März */
```



Siehe auch easter_date.

Mathematische Funktionen

Konstanten

- M_PI. 3.14159265358979323846, der Wert der Konstanten π
- M_E. 2.7182818284590452354, e
- M_LOG2E. 1.4426950408889634074, $\log_2 e$
- M_LOG10E. 0.43429448190325182765, $\log_{10} e$
- M_LN2. 0.69314718055994530942, $\log_e 2$
- M_LN10. 2.30258509299404568402, $\log_e 10$
- M_PI_2. 1.57079632679489661923, $\pi/2$
- M_PI_4. 0.78539816339744830962, $\pi/4$

- `M_1_PI`. 0.31830988618379067154, $1/\pi$
- `M_2_PI`. 0.63661977236758134308, $2/\pi$
- `M_SQRTPI`. 1.77245385090551602729, $\sqrt{\pi}$
- `M_2_SQRTPI`. 1.12837916709551257390, $2/\sqrt{\pi}$
- `M_SQRT2`. 1.41421356237309504880, $\sqrt{2}$
- `M_SQRT3`. 1.73205080756887729352, $\sqrt{3}$
- `M_SQRT1_2`. 0.70710678118654752440, $1/\sqrt{2}$
- `M_LNPI`. 1.14472988584940017414, $\log_e(\pi)$
- `M_EULER`. 0.57721566490153286061, Eulersche Konstante

abs

abs abs gibt den absoluten Betrag zurück; der Datentyp bleibt unverändert.

◆ `mixed abs(mixed number);`

acos

acos acos ist der Arcus Kosinus. Das Argument ist in Radiant anzugeben.

◆ `float acos(float arg);`
Siehe auch asin und atan.

asin

asin asin ist der Arcus Sinus. Das Argument ist in Radiant anzugeben.

◆ `float asin(float arg);`
Siehe auch acos und atan.

atan

atan atan ist der Arcus Tangens. Das Argument ist in Radiant anzugeben.

◆ `float atan(float arg);`
Siehe auch acos und atan.

atan2

atan2 ist der Arcus Tangens zwischen zwei Werten.

```
float atan2(float y, float x);
```

Die Argumente sind in Radiant anzugeben. Das Ergebnis ist zwischen Pi und -Pi.

Siehe auch acos und atan.

atan2**base_convert**

base_convert konvertiert Zahlenbasen.

```
string base_convert(string number, int frombase, int tobase);
```

Bei einer Zahlenbasis größer als 10 werden die nötigen Ziffern aus den Buchstaben a - z gebildet. Der höchste Wert einer Zahlenbasis ist 36.

```
$binary = base_convert($hexadecimal, 16, 2);
```

base_convert**bindec**

bindec konvertiert Binärzahlen nach Dezimal.

```
int bindec(string binary_string);
```

Siehe auch decbin.

bindec**ceil**

ceil rundet immer auf.

```
int ceil(float number);
```

Siehe auch floor und round.

ceil**cos**

Kosinus für ein Argument in Radiant.

```
float cos(float arg);
```

Siehe auch sin und tan.

cos**decbin**

decbin konvertiert eine Dezimalzahl in das Binärformat.

```
string decbin(int number);
```

decbin

Siehe auch `bindec`.

dechex

dechex dechex konvertiert eine Dezimalzahl in das Hexadezimalformat.

◆ `string dechex(int number);`

Siehe auch `hexdec`.

decoct

decoct decoct konvertiert eine Dezimalzahl in das Oktalformat.

◆ `string decoct(int number);`

deg2rad

deg2rad Wandelt einen Winkel von Grad in Radiant um.

`double deg2rad(double number)`

Siehe auch `rad2deg`.

exp

exp exp berechnet Potenzen der Eulerschen Zahl e .

◆ `float exp(float arg);`

Siehe auch `pow`.

floor

floor floor rundet immer ab.

◆ `int floor(float number);`

Siehe auch `ceil` und `round`.

getrandmax

getrandmax getrandmax gibt die größtmögliche Zufallszahl zurück.

◆ `int getrandmax(void);`

Siehe auch `rand`, `srand`, `mt_rand`, `mt_srand` und `mt_getrandmax`.

hexdec

hexdec hexdec konvertiert eine Hexadezimalzahl in eine Dezimalzahl.

```
int hexdec(string hex_string);
```



Siehe auch `dechex`.

log

`log` ermittelt den natürlichen Logarithmus.

log

```
float log(float arg);
```



log10

`log10` ermittelt den Logarithmus zur Basis 10.

log10

```
float log10(float arg);
```



max

`max` findet den größten Wert einer variablen Anzahl Parameter.

max

```
mixed max(mixed arg1, mixed arg2, mixed argn);
```



Wenn der erste Parameter ein Array ist, ermittelt die Funktion den höchsten Wert in diesem Array. Wenn der erste Wert kein Array ist, wird das zweite Argument unbedingt benötigt. Die Anzahl der Argumente ist nicht begrenzt. Wenn eines der Argumente eine Gleitkommazahl ist, werden alle Argumente konvertiert und als Gleitkommazahlen behandelt.

min

`min` ermittelt den niedrigsten Wert einer variablen Anzahl Parameter.

min

```
mixed min(mixed arg1, mixed arg2, mixed argn);
```



Wenn der erste Parameter ein Array ist, ermittelt die Funktion den kleinsten Wert in diesem Array. Wenn der erste Wert kein Array ist, wird das zweite Argument unbedingt benötigt. Die Anzahl der Argumente ist nicht begrenzt. Wenn eines der Argumente eine Gleitkommazahl ist, werden alle Argumente konvertiert und als Gleitkommazahlen behandelt.

mt_rand

`mt_rand` erzeugt einen besseren Zufallswert.

mt_rand

```
int mt_rand([int min], [int max]);
```



Diese Funktion akzeptiert einen unteren und oberen Grenzwert. Vergessen Sie nicht, den Startwert zu setzen.

Siehe auch `mt_srand`, `mt_getrandmax`, `srand`, `rand` und `getrandmax`.

mt_srand

mt_srand

`mt_srand` setzt den Startwert für den `mt_rand`-Zufallsgenerator.



```
void mt_srand(int seed);
```



```
// Startwert aus der laufenden Zeit
mt_srand((double)microtime()*1000000);
$randval = mt_rand();
```

Siehe auch `mt_rand`, `mt_getrandmax`, `srand`, `rand` und `getrandmax`.

mt_getrandmax

mt_getrandmax

`mt_getrandmax` zeigt den größtmöglichen Zufallszahlenwert der `mt`-Funktionen an.



```
int mt_getrandmax(void);
```

Siehe auch `mt_rand`, `mt_srand`, `rand`, `srand` und `getrandmax`.

number_format

number_format

`number_format` formatiert Zahlen als Zeichenkette.



```
string number_format(float number,
                    [[int decimals],
                    [string dec_point],
                    [string thousands_sep]]);
```

Die Funktion benötigt ein, zwei oder vier Parameter, die folgende Bedeutung haben:

- Wenn nur ein Parameter angegeben wurde, wird die Zahl ohne Dezimalstellen und mit Komma als Tausendergruppentrennung gebildet.
- Bei zwei Parametern wird der Dezimalpunkt verwendet.
- Mit den Parametern *dec_point* geben Sie ein anderes Dezimaltrennzeichen an (im deutschsprachigen Raum das Komma) und mit *thousand_sep* die Tausendergruppentrennung (normalerweise der Punkt).

octdec

octdec konvertiert eine Oktalzahl in das Dezimalformat.

```
int octdec(string octal_string);
```

Siehe auch decoct.

octdec**pi**

pi gibt die Zahl Pi zurück.

```
double pi(void);
```

pi**pow**

pow berechnet eine Potenz.

```
float pow(float base, float exp);
```

Siehe auch exp.

pow**rad2deg**

Wandelt einen Winkel in Radiant in Grad um.

```
double rad2deg(double number)
```

Siehe auch deg2rad.

rad2deg**rand**

rand erzeugt eine Zufallszahl.

```
int rand([int min], [int max]);
```

Sie können eine obere und eine untere Grenze angeben. Die obere Grenze wird außerdem von der Konstanten `RAND_MAX` bestimmt. Vergessen Sie nicht, den Startwert zu setzen.

Siehe auch `srand`, `getrandmax`, `mt_rand`, `mt_srand` und `mt_getrandmax`.

rand**round**

round rundet mathematisch korrekt auf ganze Zahlen.

```
double round(double val [, int precision]);
```

```
$foo = round(3.4);           // ergibt 3  
$foo = round(3.5);           // ergibt 4  
$foo = round(183.664, 1);     // ergibt 183,7
```

round

Beispiel

Die Funktion rundet in den Fällen nicht korrekt, in denen die interne, binäre Zahlendarstellung nicht exakt die Zahl abbildet.

Siehe auch `ceil` und `floor`.

sin

sin

Sinus für das Argument in Radiant.



```
float sin(float arg);
```

Siehe auch `cos` und `tan`.

sqrt

sqrt

`sqrt` berechnet die Quadratwurzel.



```
float sqrt(float arg);
```

srand

srand

`srand` setzt den Startwert des Zufallszahlengenerators.



```
void srand(int seed);
```



Beispiel

```
// seed with microseconds since last "whole" second
srand((double)microtime()*1000000);
$randval = rand();
```

Siehe auch `rand`, `getrandmax`, `mt_rand`, `mt_srand` und `mt_getrandmax`.

tan

tan

`tan` ermittelt den Tangens für das in Radiant übergebene Argument.



```
float tan(float arg);
```

Siehe auch `sin` und `cos`.

Funktionen beliebiger Genauigkeit

bc-Funktionen

Für die Nutzung dieser Funktionen muss die `bc`-Bibliothek mit übersetzt werden.

bcadd

bcadd

`bcadd` addiert zwei Zahlen beliebiger Genauigkeit.



```
string bcadd(string left operand,
             string right operand,
             int [scale]);
```

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcsb`.

bccomp

`bccomp` vergleicht zwei Zahlen beliebiger Genauigkeit.

```
int bccomp(string left operand, string right operand,  
           int [scale]);
```

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an. Die Funktion gibt 0 zurück, wenn die Zahlen gleich sind. Ist der linke Operand größer, wird +1 zurückgegeben, andernfalls -1.

bcddiv

`bcddiv` dividiert zwei Zahlen beliebiger Genauigkeit.

```
string bcddiv(string left operand,  
              string right operand,  
              int [scale]);
```

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcmul`.

bcmod

`bcmod` ermittelt den Rest einer Ganzzahldivision.

```
string bcmod(string left operand, string modulus);
```

Siehe auch `bcddiv`.

bcmul

`bcmul` multipliziert zwei Zahlen beliebiger Genauigkeit.

```
string bcmul(string left operand,  
             string right operand,  
             int [scale]);
```

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcddiv`.

bccomp



bcddiv



bcmod



bcmul



bcpow**bcpow**

bcpow potenziert eine Zahl *x* mit *y*.

◆ `string bcpow(string x, string y, int [scale]);`

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcsqrt`.

bcscale**bcscale**

bcscale setzt die in den anderen Funktionen optionale Angabe der Genauigkeit auf einen festen Wert.

◆ `string bcscale(int scale);`

Der Parameter *scale* gibt die Anzahl der Nachkommastellen an.

bcsqrt**bcsqrt**

bcsqrt berechnet die Quadratwurzel.

◆ `string bcsqrt(string operand, int scale);`

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcpow`.

bcsub**bcsub**

bcsub subtrahiert zwei Zahlen beliebiger Genauigkeit.

◆ `string bcsub(string left_operand,
 string right_operand,
 int [scale]);`

Der optionale Parameter *scale* gibt die Anzahl der Nachkommastellen an.

Siehe auch `bcadd`.

Funktionen für das Dateisystem

Die Funktionen für das Dateisystem umfassen Datei und Verzeichnisse. Die Funktionen sind aus naheliegenden Gründen hier im Gegensatz zur Originaldokumentation zusammengefasst worden.

dir

dir erstellt eine Verzeichnisklasse.

```
new dir(string directory);
```

Die Funktion stellt ein Pseudo-Objekt des Verzeichnisses dar, auf das mit den folgenden Methoden zugegriffen werden kann:

- `read`. Liest den nächsten Eintrag und setzt den Zeiger weiter. Wenn keine Datei mehr zu lesen ist, wird FALSE zurückgegeben.
- `rewind`. Setzt den internen Zeiger wieder auf den Anfang.
- `close`. Schließt das Verzeichnis

Außerdem sind die folgenden Eigenschaften verfügbar:

- `handle`. Gibt das handle auf das Verzeichnis zurück.
- `path`. Zeigt den aktuellen Pfad an.

```
$mydir = dir("./");
echo "Aktuelles Handle: ".$mydir->handle . "<br>\n";
echo "Pfad: ".$mydir->path . "<br>\n";
while($file=$mydir->read()) {
    echo $file."<br>\n";
}
$mydir->close();
```



Sie können anstatt der Methoden und Eigenschaften auch die folgenden Funktionen verwenden, die als Argument jeweils ein gültiges Verzeichnis-Handle erwarten:

- `$handle = opendir($path)` anstatt `$handle = dir($path)`
- `closedir($handle)` anstatt `$handle->close()`
- `readdir($handle)` anstatt `$handle->read()`
- `rewinddir($handle)` anstatt `$handle->rewind()`
- `getcwd($handle)` anstatt `$handle->path()`
- `closedir($handle)` anstatt `$handle->close()`

chdir

chdir wechselt das aktuelle Verzeichnis.

```
int chdir(string directory);
```

Die Funktion gibt TRUE zurück, wenn der Vorgang erfolgreich war, sonst FALSE.

closedir

closedir `closedir` schließt ein Verzeichnis.

◆ `void closedir(int dir_handle);`

getcwd

getcwd Die Methode gibt das aktuelle Arbeitsverzeichnis zurück.

◆ `string getcwd(void);`

opendir

opendir `opendir` öffnet ein Verzeichnis und erzeugt ein Handle darauf, das andere Funktionen zum Zugriff auf das Verzeichnis nutzen.

◆ `int opendir(string path);`

Siehe auch `closedir`, `readdir` und `rewinddir`.

readdir

readdir `readdir` liest einen Eintrag (Dateinamen) aus dem Verzeichnis.

◆ `string readdir(int dir_handle);`

Die Rückgabe der Dateinamen erfolgt nicht in einer bestimmten Reihenfolge. Für eine sortierte Ausgabe müssen Sie die Liste in einem Array ablegen und dieses sortieren.



Beispiel

```
<?php
$handle = opendir('.');
echo "Das Verzeichnis Handle ist: $handle<br>";
echo "Dateien in diesem Verzeichnis:<br>";
while ($file = readdir($handle)) {
    echo "$file<br>";
}
closedir($handle);
?>
```

rewinddir

rewinddir `rewinddir` setzt den Zeiger auf das Verzeichnis zurück.

◆ `void rewinddir(int dir_handle);`

basename

basename gibt den Namensbestandteil der Datei in einem Pfad zurück.

```
string basename(string path);
```

In Windows-Umgebungen werden als Trennzeichen sowohl normale »/« als auch windowstypische »\« Trennzeichen akzeptiert. Das folgende Beispiel gibt »index.php4« zurück:

```
$path = "/home/httpd/html/index.php4";  
$file = basename($path);
```

Siehe auch `dirname`.

basename



chgrp

chgrp weist die Datei einer Benutzergruppe zu.

```
int chgrp(string filename, mixed group);
```

Die Funktion hat unter Windows keinen Effekt. Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Sie müssen Superuser sein, um die Funktion ausführen zu können.

Siehe auch `chown` und `chmod`.

chgrp



chmod

chmod ändert die Dateizugriffsrechte.

```
int chmod(string filename, int mode);
```

Wenn Sie die übliche Notation unter Unix verwenden möchten, sollten Sie Oktalzahlen verwenden.

```
chmod( "/somedir/somefile", 755 );  
// decimal; möglicherweise inkorrekt  
chmod( "/somedir/somefile", 0755 );  
// octal; korrekt
```

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.

Siehe auch `chown` und `chgrp`.

chmod



chown

chown ändert den Eigentümer der Datei.

```
int chown(string filename, mixed user);
```

chown



Die Funktion hat unter Windows keinen Effekt. Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Sie müssen Superuser sein, um die Funktion ausführen zu können.

Siehe auch `chown` und `chmod`.

clearstatcache

clearstatcache `clearstatcache` löscht den Zwischenspeicher mit den Dateistatuswerten.

◆ `void clearstatcache(void);`

Folgende Funktionen betreffen den Dateistatus: `stat`, `lstat`, `file_exists`, `is_writeable`, `is_readable`, `is_executable`, `is_file`, `is_dir`, `is_link`, `filectime`, `fileatime`, `filemtime`, `fileinode`, `filegroup`, `fileowner`, `filesize`, `filetype` und `fileperms`.

copy

copy `copy` kopiert eine Datei von *source* nach *dest*.

◆ `int copy(string source, string dest);`

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.



Beispiel

```
if (!copy($file, $file.'.bak')) {
    print("failed to copy $file...<br>\n");
}
```

Siehe auch `rename`.

dirname

dirname `dirname` gibt den Pfadanteil einer vollständigen Pfadangabe zurück.

◆ `string dirname(string path);`

In Windows-Umgebungen werden als Trennzeichen sowohl normale »/« als auch windowstypische »\« Trennzeichen akzeptiert.



Beispiel

```
$path = "/etc/passwd";
$file = dirname($path); // $file is set to "/etc"
```

Siehe auch `basename`.

diskfreespace

diskfreespace `diskfreespace` gibt den freien Speicherplatz eines Verzeichnisses in Byte zurück.

◆ `float diskfreespace(string directory);`

```
echo "Im root sind " . diskfreespace("/")
    . " Byte frei";
```



fclose

fclose schließt einen Dateizeiger.

```
int fclose(int fp);
```

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Die Datei muss zuvor mit fopen geöffnet worden sein.

fclose



feof

feof testet den Dateizeiger auf das Dateieneende.

```
int feof(int fp);
```

Am Dateieneende wird TRUE zurückgegeben, sonst FALSE. Die Datei muss zuvor mit fopen, popen oder fsockopen geöffnet worden sein.

feof



fgetc

fgetc holt ein Zeichen von der aktuellen Position des Dateizeigers.

```
string fgetc(int fp);
```

Im Erfolgsfall wird das Zeichen zurückgegeben, am Dateieneende FALSE.

Siehe auch fread, fopen, popen, fsockopen, und fgets.

fgetc



fgetcsv

fgetcsv holt eine Zeile aus der Datei und zerlegt die Zeile anhand des Trennzeichens *delimiter*. Das Ergebnis wird als Array zurückgegeben.

```
array fgetcsv(int fp, int length, string [delimiter]);
```

Im Erfolgsfall wird ein Array zurückgegeben, am Dateieneende und bei Lesefehlern FALSE. Eine leere Zeile wird als Array mit einem Feld mit dem Wert Null zurückgegeben, nicht aber als Fehler.

```
$row=1;
$fp = fopen("test.csv","r");
while ($data = fgetcsv($fp,1000, ",")) {
    num = count($data);
    print "<p> $num fields in line $row: <br>";
    $row++;
    for ( $c=0; $c<$num; $c++ ) {
```



```

        print $data[$c] . "<br>";
    }
}
fclose($fp);

```

fgets

fgets

fgets liest eine Zeile von der aktuellen Position des Dateizeigers.



```
string fgets(int fp, int length);
```

Im Erfolgsfall wird eine Zeichenkette zurückgegeben, am Dateiende FALSE. Die Zeile endet am Zeilenumbruch oder bei *length* Zeichen.



Beispiel

```

$fd = fopen("/tmp/inputfile.txt", "r");
while ($buffer = fgets($fd, 4096)) {
    echo $buffer;
}
fclose($fd);

```

Siehe auch fread, fopen, popen, fgetc und fsockopen.

fgetss

fgetss

fgetss liest eine Zeile und entfernt alle HTML- und PHP-Tags.



```
string fgetss(int fp, int length);
```

Siehe auch fgets, fopen, fsockopen, popen und strip_tags.

file

file

file liest eine Textdatei komplett in ein Array ein. Jede Zeile ist ein Element. Das Zeilenumbruchzeichen "\n" bleibt erhalten.



```
array file(string filename);
```

Siehe auch readfile, fopen und popen.

file_exists

file_exists

file_exists prüft, ob eine Datei existiert.



```
int file_exists(string filename);
```

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

fileatime

fileatime

fileatime ermittelt Datum und Uhrzeit des letzten Dateizugriffs.

```
int filetime(string filename);
```



Im Erfolgsfall wird die Zeit zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

filetime

filetime ermittelt Datum und Uhrzeit der letzten Änderung des Dateizeigers Inode.

filetime

```
int filetime(string filename);
```



Im Erfolgsfall wird die Zeit zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

filegroup

filegroup ermittelt die Dateigruppenzugehörigkeit.

filegroup

```
int filegroup(string filename);
```



Im Erfolgsfall wird die Gruppen-ID des Eigentümers zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen. Die Funktion hat unter Windows keine Bedeutung.

fileinode

fileinode gibt den Dateizeiger Inode aus.

fileinode

```
int fileinode(string filename);
```



Im Erfolgsfall wird die Inode-Nummer zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen. Die Funktion hat unter Windows keine Bedeutung.

filemtime

filemtime ermittelt Datum und Uhrzeit der letzten Änderung an der Datei.

filemtime

```
int filemtime(string filename);
```



Im Erfolgsfall wird Datum und Uhrzeit zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

fileowner

fileowner fileowner ermittelt den Eigentümer der Datei.

◆ `int fileowner(string filename);`

Im Erfolgsfall wird die ID des Eigentümers zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

fileperms

fileperms fileperms ermittelt die Zugriffsrechte auf die Datei.

◆ `int fileperms(string filename);`

Im Erfolgsfall wird das Zugriffsrecht als Zahl zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

filesize

filesize filesize ermittelt die Dateigröße.

◆ `int filesize(string filename);`

Im Erfolgsfall wird die Größe in Bytes zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen.

filetype

filetype filetype ermittelt den Dateityp.

◆ `string filetype(string filename);`

Im Erfolgsfall wird der Dateityp als Zeichenkette zurückgegeben, sonst FALSE. Die Abfrage wird gespeichert, siehe clearstatcache für mehr Informationen. Folgende Werte können auftreten:

- fifo, char, dir, block, link, file und unknown.

flock

flock flock verriegelt eine Datei ohne Unterstützung durch das Dateisystem.

◆ `bool flock(int fp, int operation);`

Damit diese Funktion arbeiten kann, müssen alle Zugriffe in der gleichen Art und Weise erfolgen. Der Parameter *operation* bestimmt die Art der Verriegelung, der aktuelle Zustand wird als TRUE (Datei offen) oder FALSE (Datei gesperrt) zurückgegeben.

operation kann sein:

- 1. Verteilt, nur lesen
- 2. Exklusiv zum Schreiben
- 3. Datei freigeben
- 4. Zugriffe während der Sperre erlauben

fopen

fopen öffnet eine Datei vom lokalen Dateisystem oder über eine URL.

fopen

```
int fopen(string filename, string mode);
```



Wenn der Dateiname mit *http://* beginnt, wird eine HTTP 1.0-Verbindung eröffnet. Weiterleitungen werden nicht ausgeführt, die Datei muss deshalb direkt und vollständig mit abschließendem *»/«* beschrieben werden.

Wenn der Dateiname mit *ftp://* beginnt, wird eine FTP-Verbindung aufgebaut. Der Server muss den passiven Mode unterstützen. Sie können per FTP auch Dateien schreiben.

Wenn der Dateiname anders beginnt, wird auf das lokale Dateisystem zugegriffen und ein Dateizeiger wird zurückgegeben. Schlägt die Operation fehl, wird FALSE zurückgegeben.

Der Parameter *mode* kann folgende Werte annehmen:

- 'r' - öffnet nur zum Lesen und setzt den Dateizeiger auf den Anfang der Datei.
- 'r+' - öffnet zum Lesen und Schreiben und setzt den Dateizeiger auf den Anfang der Datei.
- 'w' - öffnet nur zum Schreiben, setzt den Dateizeiger auf den Anfang und die Länge der bestehenden Datei auf 0 Byte. Wenn die Datei nicht existiert, wird sie angelegt.
- 'w+' - öffnet zum Lesen und Schreiben, setzt den Dateizeiger auf den Anfang und die Länge der bestehenden Datei auf 0 Byte. Wenn die Datei nicht existiert, wird sie angelegt.

- 'a' – öffnet nur zum Schreiben, setzt den Dateizeiger an das Ende der Datei und fügt weitere Zeichen ein.
- 'a+' – öffnet zum Lesen und zum Schreiben, setzt den Dateizeiger an das Ende der Datei und fügt weitere Zeichen ein.

Zusätzlich kann der Buchstabe »b« hinzugefügt werden, der die Behandlung von Binärdateien erlaubt.



Beispiel

```
$fp = fopen("/home/rasmus/file.txt", "r");
$fp = fopen("http://www.php.net/", "r");
$fp = fopen("ftp://user:password@example.com/", "w");
```



Hinweis

Zum Schreiben müssen Sie sicherstellen, dass der Webserver Schreibrechte hat. Weichen Sie gegebenenfalls auf FTP aus.



Tipp

Auf Windows-Plattformen ist es empfehlenswert, normale Schrägstriche »/« als Verzeichnistrenner zu verwenden, um plattformunabhängig programmieren zu können. Wenn Sie dennoch Backslashes verwenden, müssen Sie diese escapen »\\«.



Beispiel

```
$fp = fopen("c:\\data\\info.txt", "r");
```

Siehe auch `fclose`, `fsockopen` und `popen`.

fpassthru

fpassthru

`fpassthru` gibt alle Daten von einem Dateizeiger direkt aus.



```
int fpassthru(int fp);
```

Die Funktion liest Daten ab Dateizeiger bis zum Ende der Datei und gibt sie direkt an den Browser aus. Im Fehlerfall wird `FALSE` zurückgegeben. Am Ende wird die Datei geschlossen, Sie können den Dateihandle `fp` nicht weiter verwenden.

Siehe auch `readfile`, `fopen`, `popen` und `fsockopen`.

fputs

fputs

`fputs` schreibt Daten an die Position des Dateizeigers. Die Funktion ist mit `fwrite` identisch.



```
int fputs(int fp, string str, int [length]);
```

fread

fread

`fread` liest Binärdaten aus einer Datei.



```
string fread(int fp, int length);
```

Die Funktion liest *length* Bytes, maximal jedoch bis zum Dateiende. Die Daten werden als Zeichenkette zurückgegeben.

```
// get contents of a file into a string
$filename = "/usr/local/something.txt";
$fd = fopen( $filename, "r" );
$content = fread( $fd, filesize( $filename ) );
fclose( $fd );
```



Siehe auch `fwrit`, `fopen`, `fsockopen`, `popen`, `fgets`, `fgetss`, `file` und `fpasssthru`.

fseek

`fseek` setzt den Dateizeiger um einen Offset *offset* vor oder zurück.

fseek

```
int fseek(int fp, int offset);
```



Im Erfolgsfall wird 0 zurückgegeben, sonst -1. Die Überschreitung des Dateiendes führt nicht zu einer Fehlermeldung, erst der nachfolgende Schreib- oder Leseversuch generiert einen Fehler.

Dateien, die per HTTP oder FTP geöffnet wurden, können so nicht gelesen werden.

Siehe auch `ftell` und `rewind`.

ftell

`ftell` ermittelt die aktuelle Position des Dateizeigers.

ftell

```
int ftell(int fp);
```



Gibt die Position in Byte vom Dateianfang oder im Fehlerfall `FALSE` zurück.

Siehe auch `fopen`, `popen`, `fseek` und `rewind`.

ftruncate

Die Funktion entfernt alle Zeichen in einer Datei ab der angegebenen Position *size*. Die Datei muss zuvor im Schreibmodus im Zugriff sein.

ftruncate

```
int ftruncate(int fp, int size)
```



fwrite

`fwrite` schreibt Binärdaten in eine Datei.

fwrite

```
int fwrite(int fp, string string, int [length]);
```



Die Funktion schreibt den Inhalt der Zeichenkette *string* in die Datei *fp*. Der Schreibvorgang wird bis zum Ende der Zeichenkette oder der durch *length* angegebenen Anzahl Zeichen fortgesetzt. Die Zeichenkette wird in keinerlei Weise behandelt, auch wenn die Option `magic_quotes_runtime` eingeschaltet wurde.

Siehe auch `fread`, `fopen`, `fsockopen`, `popen` und `fputs`.

set_file_buffer

set_file_buffer `set_file_buffer` setzt die Zeichenpufferung für eine Datei.

◆ `int fwrite(int fp, int buffer);`

Der Parameter *buffer* muss 0 sein, um die Schreibpufferung auszuschalten. Im Fehlerfall wird `FALSE` zurückgegeben. Standardmäßig werden 8-KByte-Puffer verwendet.

Siehe auch `fopen`.

is_dir

is_dir `is_dir` ermittelt, ob ein Dateiname ein Verzeichnis ist, und gibt dann `TRUE` zurück.

◆ `bool is_dir(string filename);`

Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren, müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_file` und `is_link`.

is_executable

is_executable `is_executable` ermittelt, ob die Datei ausführbar ist und gibt dann `TRUE` zurück.

◆ `bool is_executable(string filename);`

Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren, müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_file` und `is_link`.

is_file

is_file `is_file` ermittelt, ob der Name eine Datei ist. Existiert die Datei gibt die Funktion `TRUE` zurück.

```
bool is_file(string filename);
```



Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_dir` und `is_link`.

is_link

`is_link` ermittelt, ob die Datei ein symbolischer Link ist, und gibt dann `TRUE` zurück.

is_link

```
bool is_link(string filename);
```



Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren, müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_dir` und `is_file`.

is_readable

`is_readable` ermittelt, ob die Datei lesbar ist, existiert und gibt dann `TRUE` zurück.

is_readable

```
bool is_readable(string filename);
```



Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren, müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_writeable`.

is_writeable

`is_writeable` ermittelt, ob in die Datei geschrieben werden kann, und gibt dann `TRUE` zurück.

is_writeable

```
bool is_writeable(string filename);
```



Denken Sie daran, dass die Funktion das Konto des Webservers verwendet, der möglicherweise nicht über die erwarteten Rechte verfügt.

Das Ergebnis der Abfrage wird zwischengespeichert. Um den Speicher zu aktualisieren, müssen Sie die Funktion `clearstatcache` aufrufen.

Siehe auch `is_readable`.

link**link**

link erzeugt einen absoluten Link *link* mit dem Ziel *target*.

◆ `int link(string target, string link);`

Siehe auch `symlink` und `readlink`.

linkinfo**linkinfo**

linkinfo ermittelt Informationen über den Link.

◆ `int linkinfo(string path);`

Siehe auch `symlink`, `link` und `readlink`.

mkdir**mkdir**

mkdir erzeugt ein Verzeichnis *pathname*.

◆ `int mkdir(string pathname, int mode);`

Der Parameter *mode* gibt die Zugriffsrechte an. Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.



Beispiel

```
mkdir("/path/to/my/dir", 0700);
```

Siehe auch `rmdir`.

pclose**pclose**

pclose schließt ein Prozessdateihandle.

◆ `int pclose(int fp);`

Wenn ein Prozess verbunden ist und läuft, wird er terminiert.

Siehe auch `popen`.

popen**popen**

popen öffnet den Prozesszeiger.

◆ `int popen(string command, string mode);`

Die Funktion startet einen Prozess, der durch *command* angegeben wurde. Mit Hilfe der Funktionen `fgets`, `fgetss` und `fputs` können Daten vom und zum Prozess transportiert werden.



Beispiel

```
$fp = popen( "/bin/ls", "r" );
```

Siehe auch `pclose`.

readfile

`readfile` gibt eine Datei zum Browser aus und lässt das Dateihand-
le offen.

```
int readfile(string filename);
```

Um eine Fehlermeldung zu unterdrücken, wird die Funktion mit `@readfile` aufgerufen. Der Dateiname kann auch mit `http://` beginnen. Dann wird eine HTTP 1.0-Verbindung geöffnet. Ebenso ist der Zugriff auf Dateien via FTP möglich. Der FTP-Server muss den passiven Mode unterstützen.

Siehe auch `fpasssthu`, `file`, `fopen`, `include`, `require` und `virtual`.

readfile



readlink

`readlink` ermittelt das Ziel eines symbolischen Links.

```
string readlink(string path);
```

Siehe auch `symlink`, `readlink` und `linkinfo`.

readlink



rename

`rename` benennt eine Datei um.

```
int rename(string oldname, string newname);
```

Im Erfolgsfall wird `TRUE` zurückgegeben, sonst `FALSE`.

rename



rewind

`rewind` setzt den Dateizeiger auf das erste Byte der Datei.

```
int rewind(int fp);
```

Im Erfolgsfall wird `TRUE` zurückgegeben, sonst `FALSE`.

Siehe auch `fseek` und `ftell`.

rewind



rmdir

`rmdir` entfernt ein Verzeichnis.

```
int rmdir(string dirname);
```

Im Erfolgsfall wird `TRUE` zurückgegeben, sonst `FALSE`.

Siehe auch `mkdir`.

rmdir



stat**stat**

stat ermittelt verschiedene Dateiinformationen und gibt diese in einem Array zurück.



```
array stat(string filename);
```

Das assoziative Array enthält folgende Elemente:

- »device«
- »inode«
- »inode protection mode«
- »number of links«
- »user id of owner«
- »group id owner«
- »device type if inode device« (nicht bei Windows)
- »size in bytes«
- »time of last access«
- »time of last modification«
- »time of last change«
- »blocksize for filesystem I/O« (nicht bei Windows)
- »number of blocks allocated«

Das Ergebnis wird zwischengespeichert, siehe auch `clearstatcache`.

lstat**lstat**

lstat ermittelt Informationen über symbolische Links und gibt sie als Array zurück.



```
array lstat(string filename);
```

Das Array enthält folgende Elemente:

- »device«
- »inode«
- »number of links«
- »user id of owner«

- »group id owner«
- »device type if inode device« (nicht bei Windows)
- »size in bytes«
- »time of last access«
- »time of last modification«
- »time of last change«
- »blocksize for filesystem I/O« (nicht bei Windows)
- »number of blocks allocated«

Das Ergebnis wird zwischengespeichert, siehe auch `clearstatcache`.

symlink

`symlink` erzeugt einen symbolischen Link.

```
int symlink(string target, string link);
```

Siehe auch `link` und `readlink`.

symlink



tempnam

`tempnam` erzeugt einen eindeutigen und einmaligen temporären Dateinamen.

```
string tempnam(string dir, string prefix);
```

```
$tmpfname = tempnam( "/tmp", "FOO" );
```

tempnam



Beispiel

touch

`touch` setzt das Datum der letzten Änderung einer Datei.

```
int touch(string filename, int time);
```

Wenn die Datei nicht existiert, wird sie erzeugt. Im Erfolgsfall wird `TRUE` zurückgegeben, sonst `FALSE`.

touch



umask

`umask` ändert die aktuelle Umask (Zugriffsrechte) oder gibt die aktuelle zurück.

```
int umask(int mask);
```

umask



unlink

unlink

unlink löscht eine Datei.



```
int unlink(string filename);
```

Im Fehlerfall wird FALSE zurückgegeben, sonst TRUE. Siehe auch rmdir zum Entfernen von Verzeichnissen.

Funktionen zur Sitzungssteuerung

Die Sitzungssteuerungsfunktionen wurden in PHP4 eingeführt.



php.ini

Konfiguration für Sessions

Sessions können in der PHP.INI vielfältig konfiguriert werden. Die folgende Beschreibung zeigt alle Schalter. Alle Schalter werden in der folgenden Form in die PHP.INI geschrieben:

```
session.schalter = "Neuer Wert" // Zeichenkette
session.schalter = 180         // Zahlenwert
```

- session.save_handler. Name der Prozedur, mit der Daten gespeichert werden. Der Standardwert ist files.
- session.save_path. Pfad, wo Dateien erzeugt werden; Standardwert ist /TMP.
- session.name. Name des Session-Cookies. Der Standardwert ist PHPSESSID.
- session.auto_start. Wenn 1, startet das Session-Modul zu Beginn einer Anfrage automatisch. Der Standardwert ist 0.
- session.cookie_lifetime. Lebensdauer des Session-Cookies in Sekunden. Der Wert 0 erzeugt ein Cookie für die Sitzungsdauer.
- session.serialize_handler. Art der Serialisierung der Daten, kann "php" oder "wddx" sein.
- session.gc_probability. Wahrscheinlichkeit, mit der die Aufräumroutine (garbage collection) startet. Der Standardwert ist 1.
- session.gc_maxlifetime. Zeit in Sekunden, nach der Daten aufgeräumt werden.
- session.referer_check. Referer, der bei jedem Zugriff geprüft wird. Stimmt der Referer nicht, wird die Sitzung gelöscht und

neu gestartet. Der Referer ist die letzte Adresse, von der der Seitenabruf kommt.

- `session.entropy_file`. Pfad zu einer Datei mit Startwerten für den Generator der Session-IDs.
- `session.entropy_length`. Anzahl der Bytes, die von der Datei mit den Startwerten gelesen werden. Der Standardwert ist 0.
- `session.use_cookies`. Wird auf 1 gesetzt, wenn Cookies verwendet werden sollen.
- `session.cookie_path`. Pfad, der im Cookie gespeichert wird.
- `session.cookie_domain`. Domain für das Cookie.
- `session.cache_limiter`. Methoden der Cacheverwaltung: "nocache", "private", "public". "nocache" ist der Standardwert.
- `session.cache_expire`. Zeit, die Daten im Cache bleiben. Der Standardwert ist 180. Die Angabe ist wirkungslos, wenn "nocache" eingestellt wurde.

session_decode

Die Funktion dekodiert die in der Session in Form einer Zeichenkette gespeicherten Daten und erzeugt die Variablen mit Werten wieder.

session_decode

bool `session_decode`(string data) ◆

Im Fehlerfall wird FALSE zurückgegeben, sonst TRUE.

session_destroy

Löscht alle Session-Daten. Das zugehörige Cookie wird jedoch nicht sofort gelöscht.

session_destroy

bool `session_destroy`(void) ◆

Im Fehlerfall wird FALSE zurückgegeben, sonst TRUE.

session_encode

Die Funktion kodiert die angegebenen Daten in Form einer Zeichenkette für die aktuelle Session.

session_encode

string `session_encode`(void) ◆

Übernommen werden die Daten, die zuvor mit `session_register` registriert wurden.

session_id

session_id Für die angegebene Session (Session-Name) wird die aktuelle Session-ID ermittelt und zurückgegeben.

◆ `string session_id([string id])`

session_is_registered

session_is_registered Die Funktion prüft, ob eine Variable bereits registriert ist. Als Argument wird der Name der Variablen als Zeichenkette übergeben, nicht die Variable selbst.

◆ `bool session_is_registered(string name)`

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Zur Anwendung siehe auch `session_register`.

session_module_name

session_module_name Setzt oder ermittelt das Modul, mit dem die Session-Informationen verarbeitet werden.

◆ `string session_module_name([string module])`

session_name

session_name Ermittelt oder setzt den Namen der Session. Dieser Wert wird als Name des Session-Cookies verwendet, wenn mit Cookies gearbeitet wird.

◆ `string session_name([string name])`

Der Name muss in jedem Skript gesetzt werden, da er immer wieder auf den Standardwert in `session.name` zurückfällt.

session_register

session_register Die Funktion registriert Variablen für eine Sitzung. Als Argument wird der Name der Variablen angegeben, nicht die Variable selbst.

◆ `bool session_register(mixed name [, mixed ...])`

Es ist möglich, mehrere Variablen anzugeben. Die registrierten Variablen werden in dem globalen Array `$HTTP_SESSION_VARS` gespeichert.

```
$varn = 124;  
$vars = "Zeichenkette";  
session_register("varn", "vars");
```

```
$varn = 124;  
$vars = "Zeichenkette";  
$vara = array("varn", "vars");  
session_register($vara);
```

Siehe auch `session_unregister`, `session_is_registered`.



Beispiel

session_save_path

Die Funktion setzt oder ermittelt den Pfad, unter dem die Session-Daten abgespeichert werden.

```
string session_save_path([string path])
```

Der Standardpfad ist `\TMP`, dies sollten Windows-Nutzer beachten, deren temporäres Verzeichnis `\TEMP` heißt.

session_
save_path



session_start

Diese Funktion startet eine neue Session und gibt immer `TRUE` zurück. Dabei wird eine neue Session-ID erzeugt.

```
bool session_start(void)
```

session_start



session_unregister

Die Funktion hebt die Registrierung einer Variablen wieder auf. Die Variable selbst wird im aktiven Skript nicht gelöscht.

```
bool session_unregister(string name)
```

Wie bei `session_register` wird auch hier nicht die Variable, sondern deren Name als Zeichenkette übergeben. Im Gegensatz zu `session_register` können nicht mehrere Werte zugleich angegeben werden.

session_
unregister



session_set_save_handler

Diese Funktion registriert nutzerdefinierte Funktionen, die als Routinen für die Verwaltung genutzt werden. Damit kann anstatt Dateien auch eine MySQL-Datenbank verwendet werden. Die Implementierung müssen Sie aber selbst vornehmen.

```
void session_set_save_handler(string open, string close,  
                               string read, string write,  
                               string destroy, string gc)
```

session_
set_save_handler



Die Parameter haben folgende Bedeutung:

- *open*. Öffnen der Session
- *close*. Schließen der Session
- *read*. Lesen der Variablen
- *write*. Schreiben der Variablen
- *destroy*. Zerstören der Session
- *gc*. Aufräumarbeiten



Die folgenden Funktionsrumpfe müssen mit entsprechenden Zugriffen auf die Datenbank gefüllt werden:

```
function mysess_open($save_path, $sess_name) {
    ...
}
function mysess_close($sess_name) {
    ...
}
function mysess_read($sess_id) {
    ...
}
function mysess_write($sess_id, $val) {
    ...
}
function mysess_destroy($sess_id) {
    ...
}
function mysess_gc($max_lifetime) {
    ...
}
session_set_save_handler("mysess_open", "mysess_close",
    "mysess_read", "mysess_write",
    "mysess_destroy", "mysess_gc");
```

Funktionen für den Netzwerkzugriff

Mit PHP4 sind spezielle FTP-Funktionen verfügbar. Diese werden im nächsten Abschnitt erläutert.

fsockopen

fsockopen

fsockopen öffnet eine Socketverbindung unter Unix oder im Internet.



```
int fsockopen(string hostname, int port,
    int [errno], string [errstr],
    double [timeout]);
```

Bei einer Internetverbindung wird eine TCP-Verbindung eröffnet, die Portangabe ist dann notwendig. Bei Unix-Verbindungen wird der Port auf 0 gesetzt. Die Angabe einer Fehlerwartezeit *timeout* ist optional. Die Funktion gibt ein Handle auf die Verbindung zurück, die einem Dateihandle ähnelt. Sie können das Handle zusammen mit den Dateifunktionen `fgets`, `fgetss`, `fputs`, `fclose` und `feof` verwenden. Wenn die Verbindung misslingt, wird `FALSE` zurückgegeben. Standardmäßig wird der Blockmode verwendet.

```
$fp = fsockopen("www.php.net", 80,
                &$errno, &$errstr, 30);
if(!$fp) {
    echo "$errstr ($errno)<br>\n";
} else {
    fputs($fp, "GET / HTTP/1.0\n\n");
    while(!feof($fp)) {
        echo fgets($fp, 128);
    }
    fclose($fp);
}
```



Beispiel

Siehe auch `psockopen`.

ip2long

Die Funktion wandelt eine IPv4-IP-Adresse im Byte-Format in einen numerischen Wert.

ip2long

```
int ip2long(string ip_address)
```



long2ip

Die Funktion wandelt einen numerischen Wert in die korrespondierende IP-Adresse im Format »AAA.BBB.CCC.DDD« um.

long2ip

```
string long2ip(int proper_address)
```



psockopen

`psockopen` öffnet eine persistente Socketverbindung.

psockopen

```
int psockopen(string hostname, int port,
              int [errno], string [errstr],
              int [timeout]);
```



Die Funktion entspricht `fsockopen`. Am Ende des Skripts wird die Verbindung aber nicht geschlossen, sondern für einen erneuten Zugriff offen gehalten. Dies funktioniert nicht bei der CGI-Version.

set_socket_blocking

set_socket_blocking set_socket_blocking setzt den Blockmode. Der Blockmode beeinflusst die Arbeitsweise der Funktion fgets.

◆ `int set_socket_blocking(int socket descriptor, int mode);`

Ist *mode* FALSE, wird der Blockmode ausgeschaltet. Wenn der Blockmode ausgeschaltet ist, liest fgets von der Verbindung und kehrt sofort zurück, unabhängig davon, ob Daten vorlagen oder nicht. Im Blockmode wartet die Funktion so lange, bis tatsächlich Daten vorliegen.

gethostbyaddr

gethostbyaddr gethostbyaddr ermittelt den Namen eines Hosts anhand der IP-Adresse.

◆ `string gethostbyaddr(string ip_address);`

Im Fehlerfall wird die IP-Adresse wieder zurückgegeben. Die Funktion benötigt einen direkten Zugriff auf einen Nameserver.

Siehe auch gethostbyname.

gethostbyname

gethostbyname gethostbyname ermittelt die IP-Adresse anhand eines Hostnamens.

◆ `string gethostbyname(string hostname);`

Siehe auch gethostbyaddr.

gethostbyname1

gethostbyname1 gethostbyname1 ermittelt alle IP-Adressen zu einem Hostnamen und gibt diese als Array zurück.

◆ `array gethostbyname1(string hostname);`

Siehe auch gethostbyname, gethostbyaddr, checkdnsrr und getmxrr.

checkdnsrr

checkdnsrr checkdnsrr prüft einen bestimmten, durch *type* ausgewählten DNS-Eintrag.

◆ `int checkdnsrr(string host, string [type]);`

Zulässige Werte für *type* sind:

- A, MX, NS, SOA, PTR, CNAME oder ANY. Der Standardwert ist MX.

Siehe auch `getmxrr`, `gethostbyaddr`, `gethostbyname` und `gethostbyname1`.

getmxrr

`getmxrr` prüft den MX-Eintrag im DNS für einen Hostnamen.

```
int getmxrr(string hostname, array mxhosts, array [weight]);
```

Siehe auch `checkdnsrr`, `gethostbyname`, `gethostby`.

getmxrr



FTP-Funktionen

Die FTP-Funktionen sind in PHP4 hinzugekommen. Die Nutzung von FTP-Servern wird damit wesentlich vereinfacht.



ftp_connect

Die Funktion `ftp_connect` verbindet mit einem FTP-Server und gibt ein Handle auf diese Verbindung zurück, das von allen folgenden FTP-Funktionen genutzt wird.

```
int ftp_connect(string host, int [port]);
```

Der Parameter `port` gibt eine optionale Portnummer an. Erfolgt keine Angabe, wird der Standardport 21 benutzt.

ftp_connect



ftp_login

Diese Funktion führt eine Anmeldung an einem FTP-Server aus.

```
int ftp_login(int ftp_stream, string username, string password);
```

Im Erfolgsfall wird `TRUE` zurückgegeben, sonst `FALSE`.

ftp_login



ftp_pwd

`ftp_pwd` gibt den Namen des aktuellen Verzeichnisses zurück.

```
mixed ftp_pwd(int ftp_stream);
```

War die Verbindung nicht möglich, wird `FALSE` zurückgegeben.

ftp_pwd



ftp_chdir

`ftp_chdir` wechselt das aktuelle Verzeichnis auf dem Server.

```
int ftp_chdir(int ftp_stream, string newdir);
```

ftp_chdir



War der Wechsel erfolgreich, wird TRUE zurückgegeben, sonst FALSE.

ftp_cdup

ftp_cdup ftp_cdup wechselt eine Verzeichnisebene höher.

◆ int **ftp_cdup**(int ftp_stream);

ftp_mkdir

ftp_mkdir ftp_mkdir erzeugt ein Verzeichnis *directory*.

◆ int **ftp_mkdir**(int ftp_stream, string directory);

ftp_rmdir

ftp_rmdir ftp_rmdir löscht das Verzeichnis *directory*.

◆ int **ftp_rmdir**(int ftp_stream, string directory);

ftp_nlist

ftp_nlist ftp_nlist listet die Dateien des Verzeichnisses *directory* auf.

◆ array **ftp_nlist**(int ftp_stream, string directory);

Die Liste der Dateinamen wird als Array zurückgegeben. Im Fehlerfall wird FALSE zurückgegeben.

ftp_rawlist

ftp_rawlist ftp_rawlist gibt eine detaillierte Liste von Dateien im Verzeichnis *directory* zurück.

◆ array **ftp_rawlist**(int ftp_stream, string directory);

Die Funktion führt intern das FTP-Kommando LIST aus. Das resultierende Array enthält die Ausgabe des LIST-Kommandos, wobei jedes Element eine Zeile enthält. Eine Bearbeitung erfolgt nicht, das Array enthält praktisch die Rohdaten.

ftp_systype

ftp_systype ftp_systype erkennt den Systemtyp des FTP-Servers.

◆ string **ftp_systype**(int ftp_stream);

ftp_pasv

ftp_pasv schaltet den passiven Modus ein (*pasv* ist TRUE) oder aus (*pasv* ist FALSE).

```
int ftp_pasv(int ftp_stream, int pasv);
```

ftp_get

ftp_get lädt eine Datei vom FTP-Server auf das lokale System (die Maschine, auf der PHP läuft – dies kann ebenso ein Server sein).

```
int ftp_get(int ftp_stream, string local_file,  
            string remote_file, int mode);
```

local_file bestimmt den Dateinamen im aktuellen Verzeichnis. Für *mode* kann eine der beiden folgenden Konstanten angegeben werden:

- FTP_ASCII. Übertragung als Textdatei.
- FTP_BINARY. Übertragung als Binärdatei.

Siehe auch ftp_cdup, ftp_pwd.

ftp_fget

ftp_fget lädt eine Datei vom FTP-Server in eine geöffnete Datei (die Maschine, auf der PHP läuft – dies kann ebenso ein Server sein).

```
int ftp_fget(int ftp_stream, int file_pointer,  
             string remote_file, int mode);
```

remote_file bestimmt den Dateinamen im Ziel. Für *mode* kann eine der beiden folgenden Konstanten angegeben werden:

- FTP_ASCII. Übertragung als Textdatei.
- FTP_BINARY. Übertragung als Binärdatei.

Siehe auch ftp_cdup, ftp_pwd.

ftp_put

ftp_put überträgt eine Datei auf den FTP-Server.

```
int ftp_fput(int ftp_stream, string remote_file,  
             string local_file, int mode);
```

local_file bestimmt den lokalen, *remote_file* den entfernten Dateinamen. Für *mode* kann eine der beiden folgenden Konstanten angegeben werden:

- FTP_ASCII. Übertragung als Textdatei.
- FTP_BINARY. Übertragung als Binärdatei.

Siehe auch ftp_fget, ftp_get.

ftp_fput

ftp_fput ftp_fput überträgt eine lokal geöffnete Datei auf den FTP-Server.

◆ `int ftp_fput(int ftp_stream, string remote_file,
string file_pointer, int mode);`

remote_file bestimmt den Dateinamen, *file_pointer* ist ein Handle auf eine geöffnete Datei. Für *mode* kann eine der beiden folgenden Konstanten angegeben werden:

- FTP_ASCII. Übertragung als Textdatei.
- FTP_BINARY. Übertragung als Binärdatei.

Siehe auch ftp_put, ftp_fget.

ftp_size

ftp_size ftp_size gibt die Größe einer Datei zurück.

◆ `int ftp_size(int ftp_stream, string remote_file);`

Die Funktion gibt die Größe in Byte zurück. Existiert die Datei nicht, wird -1 zurückgegeben. Die Funktion wird nicht von allen Servern unterstützt. Verwenden Sie dann ftp_rawlist.

ftp_mdtm

ftp_mdtm ftp_mdtm gibt das Datum der letzten Änderung einer Datei als Unix-Zeitstempel zurück.

◆ `int ftp_mdtm(int ftp_stream, string remote_file);`

Existiert die Datei nicht, wird -1 zurückgegeben. Die Funktion wird nicht von allen Servern unterstützt. Verwenden Sie dann ftp_rawlist.

ftp_rename

ftp_rename ftp_rename benennt die Datei *from* in *to* um.

```
int ftp_rename(int ftp_stream, string from, string to);
```



Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

ftp_delete

ftp_delete löscht eine Datei im Pfad *path*.

ftp_delete

```
int ftp_delete(int ftp_stream, string path);
```



Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

ftp_quit

ftp_quit beendet die Verbindung zu einem FTP-Server.

ftp_quit

```
int ftp_quit(int ftp_stream);
```



Verschlüsselungs-Funktionen

Für diese Funktionen benötigen Sie die Bibliothek MCRYPT, die Sie unter der folgenden Adresse bekommen:

- <ftp://argeas.cs-net.gr/pub/unix/mcrypt/>

Die Bibliothek unterstützt eine Reihe von Verschlüsselungsverfahren, u.a. DES, TripleDES, Blowfish (default), 3-WAY, SAFER-SK64, SAFER-SK128, TWOFISH, TEA, RC2 und GOST in CBC, OFB, CFB und ECB Modi.

```
<?php
$key = "this is a very secret key";
$input = "Let us meet at 9 o'clock at the secret place.";
$encrypted_data = mcrypt_ecb(MCRYPT_TripleDES, $key, $input,
MCRYPT_ENCRYPT);
?>
```



Beispiel

Liste der unterstützten Routinen:

- MCRYPT_BLOWFISH
- MCRYPT_DES
- MCRYPT_TripleDES
- MCRYPT_ThreeWAY
- MCRYPT_GOST
- MCRYPT_CRYPT
- MCRYPT_DES_COMPAT
- MCRYPT_SAFER64

- MCRYPT_SAFER128
- MCRYPT_CAST128
- MCRYPT_TEAN
- MCRYPT_RC2
- MCRYPT_TWOFISH (für ältere Mcrypt 2.x-Versionen)
- MCRYPT_TWOFISH128 (TWOFISHxxx sind in neueren 2.x-Versionen verfügbar)
- MCRYPT_TWOFISH192
- MCRYPT_TWOFISH256
- MCRYPT_RC6
- MCRYPT_IDEA

mcrypt_get_cipher_name

mcrypt_get_cipher_name

mcrypt_get_cipher_name ermittelt den Namen der Chiffre. Wenn die Chiffre nicht existiert, wird FALSE zurückgegeben.



Beispiel

```
string mcrypt_get_cipher_name(int cipher);
```

```
<?php
$cipher = MCRYPT_TripleDES;
print mcrypt_get_cipher_name($cipher);
?>
```

Dieses Beispiel gibt Folgendes aus:

TripleDES

mcrypt_get_block_size

mcrypt_get_block_size

mcrypt_get_block_size ermittelt die Blockgröße der Chiffre *cipher* in Bytes.



```
int mcrypt_get_block_size(int cipher);
```

Siehe auch mcrypt_get_key_size.

mcrypt_get_key_size

mcrypt_get_key_size

mcrypt_get_key_size ermittelt die Schlüssellänge der Chiffre *cipher* in Bytes.



```
int mcrypt_get_key_size(int cipher);
```

Siehe auch mcrypt_get_block_size.

mcrypt_create_iv

mcrypt_create_iv ermittelt den Initialisierungsvektor auf der Basis einer zufälligen Quelle *source*.

```
string mcrypt_create_iv(int size, int source);
```

source kann sein:

- MCRYPT_RAND (Zufallsgenerator des Betriebssystems)
- MCRYPT_DEV_RANDOM (liest Daten aus /dev/random)
- MCRYPT_DEV_URANDOM (liest Daten aus /dev/urandom).

Wenn MCRYPT_RAND verwendet wird, muss der Zufallszahlengenerator mit srand initialisiert werden.

```
<?php
$cipher = MCRYPT_TripleDES;
$block_size = mcrypt_get_block_size($cipher);
$iv = mcrypt_create_iv($block_size, MCRYPT_DEV_RANDOM);
?>
```



Beispiel

mcrypt_cbc

mcrypt_cbc ver- und entschlüsselt im CBC-Mode.

```
int mcrypt_cbc(int cipher, string key, string data,
               int mode, string [iv]);
```

cipher ist eine der MCRYPT_chiffrename-Konstanten. *key* ist der geheime Schlüssel des Algorithmus. *data* enthält die zu ver- oder entschlüsselnden Daten. *mode* ist MCRYPT_ENCRYPT oder MCRYPT_DECRYPT.

Siehe auch mcrypt_cfb, mcrypt_ecb, mcrypt_ofb.

mcrypt_cfb

mcrypt_cfb ver- und entschlüsselt im CFB-Mode.

```
int mcrypt_cfb(int cipher, string key, string data,
               int mode, string iv);
```

cipher ist eine der MCRYPT_chiffrename-Konstanten. *key* ist der geheime Schlüssel des Algorithmus. *data* enthält die zu ver- oder entschlüsselnden Daten. *mode* ist MCRYPT_ENCRYPT oder MCRYPT_DECRYPT.

Siehe auch mcrypt_cbc, mcrypt_ecb, mcrypt_ofb.

mcrypt_ecb

mcrypt_ecb

mcrypt_ecb ver- und entschlüsselt im ECB-Mode.



```
int mcrypt_ecb(int cipher, string key, string data, int mode);
```

cipher ist eine der MCRYPT_chiffrename-Konstanten. *key* ist der geheime Schlüssel des Algorithmus. *data* enthält die zu ver- oder entschlüsselnden Daten. *mode* ist MCRYPT_ENCRYPT oder MCRYPT_DECRYPT.

Siehe auch mcrypt_cbc, mcrypt_cfb, mcrypt_ofb.

mcrypt_ofb

mcrypt_ofb

mcrypt_ofb ver- und entschlüsselt im OFB-Mode.



```
int mcrypt_ofb(int cipher, string key, string data,
               int mode, string iv);
```

cipher ist eine der MCRYPT_chiffrename-Konstanten. *key* ist der geheime Schlüssel des Algorithmus. *data* enthält die zu ver- oder entschlüsselnden Daten. *mode* ist MCRYPT_ENCRYPT oder MCRYPT_DECRYPT.

Siehe auch mcrypt_cbc, mcrypt_cfb, mcrypt_ecb.

Hashfunktionen

Diese Funktionen benötigen die MHASH-Bibliothek. Mehr Informationen finden Sie unter der Adresse:

- <http://sasweb.de/mhash>

MHASH wird verwendet, um Prüfsummen oder Kodierungen zu erzeugen. Unterstützt werden Algorithmen wie MD5, SHA1, GOST und andere.



```
<?php
$input = "Let us meet at 9 o' clock at the secret place.";
$hash = mhash(MHASH_SHA1, $input);
print "The hash is ".bin2hex($hash)."\n";
?>
```

Dieses Beispiel gibt folgenden Satz aus:

The hash is d3b85d710d8f6e4e5efd4d5e67d041f9cecedafe

Folgende Hashes werden derzeit unterstützt:

- MHASH_MD5
- MHASH_SHA1
- MHASH_HAVAL

- MHASH_RIPEMD160
- MHASH_RIPEMD128
- MHASH_SNEFRU
- MHASH_TIGER
- MHASH_GOST
- MHASH_CRC32
- MHASH_CRC32B

mhash_get_hash_name

Die Funktion `mhash_get_hash_name` ermittelt den Namen des spezifischen Hashes *hash*.

```
string mhash_get_hash_name(int hash);
```

```
<?php
$hash = MHASH_MD5;
print mhash_get_hash_name($hash);
?>
```

Dieses Beispiel gibt Folgendes aus:

MD5

mhash_get_hash_name



Beispiel

mhash_get_block_size

`mhash_get_block_size` ermittelt die Blockgröße des Hashes in Bytes. Wenn der Hash nicht existiert, wird `FALSE` zurückgegeben.

```
int mhash_get_block_size(int hash);
```

mhash_get_block_size



mhash_count

`mhash_count` ermittelt die größte Hash-ID, beginnend mit 0.

```
int mhash_count;
```

```
<?php
$nr = mhash_count;
for($i = 0; $i <= $nr; $i++) {
    echo sprintf("Die Blockgröße von %s ist gleich %d<br>",
        mhash_get_hash_name($i),
        mhash_get_block_size($i));
}
?>
```

mhash_count



Beispiel

mhash

mhash

mhash berechnet einen Hash auf Grundlage der Daten *data*.



```
string mhash(int hash, string data);
```

WDDX-Funktionen

Diese Funktionen arbeiten mit WDDX. Mehr Informationen finden Sie unter der folgenden Adresse:

- <http://www.wddx.org>

Zwei Beispiele aus der PHP-Dokumentation zeigen, wie die WDDX-Funktionen angewendet werden:



```
<?php
print wddx_serialize_value("PHP to WDDX packet example",
                           "PHP packet");
?>
```

Dieses Beispiel gibt aus:

```
<wddxPacket version='0.9'>
  <header comment='PHP packet' />
  <data>
    <string>PHP to WDDX packet example
  </string>
  </data>
</wddxPacket>
```



```
<?php
$pi = 3.1415926;
$packet_id = wddx_packet_start("PHP");
wddx_add_vars($packet_id, "pi");
/* Suppose $cities came from database */
$cities = array("Austin", "Novato", "Seattle");
wddx_add_vars($packet_id, "cities");
$packet = wddx_packet_end($packet_id);
print $packet;
?>
```

Dieses Beispiel gibt aus:

```
<wddxPacket version='0.9'>
  <header comment='PHP' />
  <data>
    <struct>
      <var name='pi'>
        <number>3.1415926</number>
      </var>
      <var name='cities'>
        <array length='3'>
```

```

        <string>Austin</string>
        <string>Novato</string>
        <string>Seattle</string>
    </array>
</var>
</struct>
</data>
</wddxPacket>

```

wddx_serialize_value

wddx_serialize_value serialisiert einen Wert in ein WDDX-Paket und gibt das Paket zurück.

```
string wddx_serialize_value(mixed var, string [comment]);
```

wddx_serialize_value

wddx_serialize_vars

wddx_serialize_vars serialisiert mehrere Werte in ein WDDX-Paket und gibt das Paket zurück.

```
string wddx_serialize_vars(string var_name |
                           array var_names [, ... ] );
```

wddx_serialize_vars

```

<?php
$a = 1;
$b = 5.5;
$c = array("blue", "orange", "violet");
$d = "colors";
$c1vars = array("c", "d");
print wddx_serialize_vars("a", "b", $c1vars);
?>

```



Beispiel

Diese Beispiel gibt Folgendes aus:

```

<wddxPacket version='0.9'>
  <header/>
  <data>
    <struct>
      <var name='a'>
        <number>1</number>
      </var>
      <var name='b'>
        <number>5.5</number>
      </var>
      <var name='c'>
        <array length='3'>
          <string>blue</string>
          <string>orange</string>
          <string>violet</string>
        </array>
      </var>
    </struct>
  </data>
</wddxPacket>

```

```

        <var name='d'>
            <string>colors</string>
        </var>
    </struct>
</data>
</wddxPacket>

```

wddx_packet_start

wddx_packet_start

wddx_packet_start startet ein neues WDDX-Paket und gibt eine ID zur späteren Verwendung zurück.



```
int wddx_packet_start(string [comment]);
```

wddx_packet_end

wddx_packet_end

wddx_packet_end beendet ein WDDX-Paket, das durch die packet_id adressiert wird.



```
int wddx_packet_end(int packet_id);
```

wddx_add_vars

wddx_add_vars

wddx_add_vars serialisiert Variablen, fügt diese einem Paket hinzu und beendet das Paket dann.



```
void wddx_add_vars(int packet_id, ...);
```

wddx_deserialize

wddx_deserialize

wddx_deserialize deserialisiert ein WDDX-Paket.



```
mixed wddx_deserialize(string packet);
```

XML-Funktionen

XML (eXtensible Markup Language) ist eine universelle Sprache zur Definition von Auszeichnungssprachen. Mehr Informationen finden Sie im Web unter der folgenden Adresse:

- <http://www.w3.org/XML/>

Ereignisbehandlungsroutinen

Tabelle B.1:
Unterstützte XML-
Ereignisbehand-
lungsroutinen

Funktion	Beschreibung
xml_set_element_handler	Diese Funktion wird gefeuert, wenn der Parser ein XML-Element erreicht oder eines verlässt. Start- und End-Tags werden getrennt behandelt.

Funktion	Beschreibung
<code>xml_set_character_data_handler</code>	Diese Funktion behandelt alle Zeichen, die nicht als Tag gelten. Das betrifft auch Whitespaces.
<code>xml_set_processing_instruction_handler</code>	Hiermit werden die einleitenden Tags, wie <code><?php</code> , spezifiziert.
<code>xml_set_default_handler</code>	Diese Funktion wird ausgelöst, wenn sich keine andere für ein Ereignis zuständig fühlt.
<code>xml_set_unparsed_entity_decl_handler</code>	Diese Funktion behandelt nicht geparste Einheiten (NDATA).
<code>xml_set_notation_decl_handler</code>	Eine Notations-Deklaration wird hiermit behandelt.
<code>xml_set_external_entity_ref_handler</code>	Wenn eine externe Referenz gefunden wird, ist diese Funktion zuständig.

XML-Fehlercodes

XML_ERROR_NONE

XML_ERROR_NO_MEMORY

XML_ERROR_SYNTAX

XML_ERROR_NO_ELEMENTS

XML_ERROR_INVALID_TOKEN

XML_ERROR_UNCLOSED_TOKEN

XML_ERROR_PARTIAL_CHAR

XML_ERROR_TAG_MISMATCH

XML_ERROR_DUPLICATE_ATTRIBUTE

XML_ERROR_JUNK_AFTER_DOC_ELEMENT

XML_ERROR_PARAM_ENTITY_REF

XML_ERROR_UNDEFINED_ENTITY

XML_ERROR_RECURSIVE_ENTITY_REF

XML_ERROR_ASYNC_ENTITY

```
XML_ERROR_BAD_CHAR_REF
XML_ERROR_BINARY_ENTITY_REF
XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
XML_ERROR_MISPLACED_XML_PI
XML_ERROR_UNKNOWN_ENCODING
XML_ERROR_INCORRECT_ENCODING
XML_ERROR_UNCLOSED_CDATA_SECTION
XML_ERROR_EXTERNAL_ENTITY_HANDLING
```

xml_parser_create

xml_parser_create

xml_parser_create erzeugt einen neuen XML-Parser und gibt ein Handle darauf zurück.

```
◆ int xml_parser_create(string [encoding]);
```

Die Zeichenkette *encoding* ist optional und gibt an, welche Kodierung der Parser für die Zeichen nutzen soll. Zulässig sind drei Werte:

- ISO-8859-1 (default)
- US-ASCII
- UTF-8

Im Fehlerfall wird FALSE zurückgegeben, sonst ein Handle auf den neuen Parser.

xml_set_element_handler

xml_set_element_handler

xml_set_element_handler setzt die Start- und Ende-Ereignisbehandlungsroutinen

```
◆ int xml_set_element_handler(int parser,
                             string startElementHandler,
                             string endElementHandler);
```

Die Parameter *startElementHandler* und *endElementHandler* enthalten Funktionen, die beim Beginn eines Elements (öffnendes Tag) und am Ende eines Elements (schließendes Tag) aufgerufen werden. Die Funktionen sind dann für die eigentliche Datenverarbeitung zuständig.

Die Funktion für *startElementHandler* muss drei Parameter akzeptieren:

```
startElementHandler(int parser, string name, string attribs);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *name*. Dieser Parameter enthält den Namen des Elements. Wenn die Behandlungsmethode Case-Folding genutzt wird, sind die Zeichen immer Großbuchstaben.
- *attribs*. Der dritte Parameter enthält ein assoziatives Array mit den Attributen des Elements. Die Schlüssel sind die Attribute und die Elemente die Werte. Wenn die Behandlungsmethode »Case-Folding« genutzt wird, sind die Zeichen immer Großbuchstaben.

Die Funktion für *endElementHandler* muss zwei Parameter akzeptieren:

```
endElementHandler(int parser, string name);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *name*. Dieser Parameter enthält den Namen des Elements. Wenn die Behandlungsmethode »Case-Folding« genutzt wird, sind die Zeichen immer Großbuchstaben.

Wenn eine oder beide Funktionen auf eine leere Zeichenkette gesetzt werden, wird die Behandlungsroutine ausgeschaltet.

Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert. Behandlungsfunktionen auf Objektbasis werden derzeit nicht unterstützt.

xml_set_character_data_handler

`xml_set_character_data_handler` setzt die Behandlungsfunktionen für Zeichendaten.

xml_set_character_data_handler

```
int xml_set_character_data_handler(int parser, string handler);
```



handler bezeichnet eine Funktion, die Zeichendaten – das heißt Daten außerhalb aller Tags – behandelt. Die Funktion muss zwei Parameter akzeptieren:

```
handler(int parser, string data);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *data*. Dieser Parameter enthält die Daten, die bearbeitet werden müssen.

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert. Behandlungsfunktionen auf Objektbasis werden derzeit nicht unterstützt.

xml_set_processing_instruction_handler

xml_set_processing_instru

xml_set_processing_instruction_handler setzt die Behandlung von Prozessinformationen (PI).



```
int xml_set_processing_instruction_handler(int parser,
                                         string handler);
```

handler bezeichnet eine Funktion, die zur Prozessinformationsbehandlung dient. Eine Prozessinformation hat folgendes Format:

```
<?target data?>
```

Beachten Sie, dass die schließende Klammer analog der PHP-Tags geschrieben wird. Sie können nicht innerhalb des PI-Tags PHP verwenden, da andernfalls das schließende PHP-Tag den Parser durcheinander bringt.

Die Funktion *handler* muss drei Parameter akzeptieren:

```
handler(int parser, string target, string data);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *target*. Dieser Parameter enthält das target-Element des PI-Tags.
- *data*. Dieser Parameter enthält die Daten des PI-Tags.

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert. Behandlungsfunktionen auf Objektbasis werden derzeit nicht unterstützt.

xml_set_default_handler

xml_set_default_handler

xml_set_default_handler setzt eine Standardbehandlungsroutine auf *handler*.



```
int xml_set_default_handler(int parser, string handler);
```

handler muss eine gültige Funktion bezeichnen, die eingesetzt wird, wenn keine anderen Behandlungsroutinen in Frage kommen. Die Funktion *handler* muss zwei Parameter akzeptieren:

```
handler(int parser, string data);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *data*. Die Daten des Elements ohne weitere Sezierung.

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert. Behandlungsfunktionen auf Objektbasis werden derzeit nicht unterstützt.

xml_set_unparsed_entity_decl_handler

Die Funktion `xml_set_unparsed_entity_decl_handler` setzt eine Behandlungsroutine *handler* für externe ungeparste Entitäten des Parser *parser*. Parameter sind das Parser-Handle und die Handler-Funktion.

xml_set_unparsed_entity_decl_handler

```
int xml_set_unparsed_entity_decl_handler(int parser,
                                         string handler);
```



Die Behandlungsroutine wird für die Bearbeitung folgender externer NDATA Deklarationen eingesetzt:

```
<!ENTITY name {publicId | systemId} NDATA notationName>
```

Die Funktion *handler* muss sechs Parameter akzeptieren:

```
handler(int parser, string entityName, string base, string
systemId, string publicId, string notationName);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *entityName*. Name der zu definierenden Entität.
- *base*. Basis für die Auflösung der System-ID der externen Entität. Derzeit findet dieser Parameter keine Verwendung und ist immer eine leere Zeichenkette.
- *systemId*. System-ID der externen Entität.
- *publicId*. Public-ID der externen Entität.
- *notationName*. Name der Notation (siehe auch bei der Funktion `xml_set_notation_decl_handler`).

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert.

xml_set_notation_decl_handler

xml_set_notation_decl_handler xml_set_notation_decl_handler setzt eine Behandlungsroutine für die Bearbeitung von Notationen.

◆ `int xml_set_notation_decl_handler(int parser, string handler);`

Eine Notations-Deklaration ist Bestandteil der DTD des Dokuments und hat folgendes Format:

```
<!NOTATION name {systemId | publicId}>
```

Die Funktion *handler* muss fünf Parameter akzeptieren:

```
handler(int parser, string notationName, string base,
        string systemId, string publicId);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *notationName*. Name der Notation.
- *base*. Basis für die Auflösung der System-ID der externen Entität. Derzeit findet dieser Parameter keine Verwendung und ist immer eine leere Zeichenkette.
- *systemId*. System-ID der externen Entität.
- *publicId*. Public-ID der externen Entität.

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert.

xml_set_external_entity_ref_handler

xml_set_external_entity_ref_handler xml_set_external_entity_ref_handler setzt eine Behandlungsroutine für externe Entitäten.

◆ `int xml_set_external_entity_ref_handler(int parser, string handler);`

Die Funktion *handler* muss fünf Parameter akzeptieren und eine Ganzzahl zurückgeben. Gibt die Funktion FALSE zurück, stoppt der Parser und `xml_get_error_code` gibt XML_ERROR_EXTERNAL_ENTITY_HANDLING zurück.

```
int handler(int parser, string openEntityNames, string base,
           string systemId, string publicId);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *openEntityNames*. Eine durch Leerzeichen getrennte Liste der Namen der Entitäten.
- *base*. Basis für die Auflösung der System-ID der externen Entität. Derzeit findet dieser Parameter keine Verwendung und ist immer eine leere Zeichenkette.
- *systemId*. System-ID der externen Entität.
- *publicId*. Public-ID der externen Entität.

Wird eine leere Zeichenkette übergeben, wird die Funktion abgeschaltet. Die Funktion gibt TRUE zurück, wenn die Behandlungsfunktionen eingebunden werden konnten. FALSE wird zurückgegeben, wenn der Parser nicht existiert.

xml_parse

xml_parse startet das Parsen des Dokuments.

```
int xml_parse(int parser, string data, int [isFinal]);
```

- *parser*. Dies ist eine Referenz zum Parser.
- *data*. Daten, die zu bearbeiten sind. Dies kann ein komplettes Dokument oder ein Teil davon sein. Um zu erkennen, ob das Dokument in Teilen fortgesetzt wird, kann der optionale Parameter *isFinal* angegeben werden.
- *isFinal* (optional). Dieser Parameter wird auf TRUE gesetzt, wenn das letzte Dokumentenstück geliefert wird.

Während der Bearbeitung werden die Behandlungsfunktionen sooft wie nötig aufgerufen. Die Funktion gibt TRUE zurück, wenn das Dokument erfolgreich verarbeitet werden konnte, sonst FALSE. Fehlermeldungen können mit den folgenden Funktionen ermittelt werden:

- `xml_get_error_code`, `xml_error_string`, `xml_get_current_line_number`, `xml_get_current_column_number` und `xml_get_current_byte_index`.

xml_parse



xml_get_error_code

xml_get_error_code xml_get_error_code ermittelt den Fehlercode der letzten Aktion des Parsers.

◆
`int xml_get_error_code(int parser);`
parser ist eine Referenz zum Parser.

xml_error_string

xml_error_string xml_error_string ermittelt die Fehlermeldung der letzten Aktion des Parsers.

◆
`string xml_error_string(int code);`
code gibt den Fehlercode an, wie ihn die Funktion `xml_get_error_code` zurückgibt.

xml_get_current_line_number

xml_get_current_line_number xml_get_current_line_number gibt die aktuelle Zeile zurück, wo sich der Parser befindet.

◆
`int xml_get_current_line_number(int parser);`
parser ist eine Referenz zum Parser.

xml_get_current_column_number

xml_get_current_column_number xml_get_current_column_number gibt die aktuelle Spalte zurück, wo sich der Parser befindet.

◆
`int xml_get_current_column_number(int parser);`
parser ist eine Referenz zum Parser.

xml_get_current_byte_index

xml_get_current_byte_index xml_get_current_byte_index ermittelt den aktuellen Byte-Index (Zeichen in der Datei).

◆
`int xml_get_current_byte_index(int parser);`
parser ist eine Referenz zum Parser.

xml_parser_free

xml_parser_free xml_parser_free gibt den vom Parser verwendeten Speicher wieder frei.

```
string xml_parser_free(int parser);
```



parser ist eine Referenz zum Parser.

xml_parser_set_option

xml_parser_set_option setzt die Option *option* des Parsers auf den Wert *value*.

xml_parser_set_option

```
int xml_parser_set_option(int parser, int option, mixed value);
```



parser ist eine Referenz zum Parser. *option* kann eine der Angaben in der folgenden Tabelle sein, *value* der Wert der Option.

Konstanten für <i>option</i>	Datentyp	Syntax
XML_OPTION_CASE_FOLDING	integer	Kontrolliert, ob Case-Folding eingeschaltet ist, was standardmäßig der Fall ist.
XML_OPTION_TARGET_ENCODING	string	Setzt die Art des Dekodierens. Unterstützt werden ISO-8859-1, US-ASCII und UTF-8.

Tabelle B.2: XML-Parser-Optionen

xml_parser_get_option

xml_parser_get_option ermittelt die Option *option* des XML-Parsers *parser*.

xml_parser_get_option

```
mixed xml_parser_get_option(int parser, int option);
```



parser ist eine Referenz zum Parser. *option* gibt an, zu welcher Option der Wert ermittelt werden soll. Siehe unter `xml_parser_set_option` für eine Liste der Optionen.

utf8_decode

utf8_decode konvertiert UTF-8 kodierte Zeichen nach ISO-8859-1.

utf8_decode

```
string utf8_decode(string data);
```



Siehe `utf8_encode` für nähere Angaben zur UTF-8-Kodierung.

utf8_encode

utf8_encode kodiert ISO-8859-1 Zeichen nach UTF-8.

utf8_encode

```
string utf8_encode(string data);
```



UTF-8 ist ein Standardverfahren zur Kodierung von Unicode-Zeichen in ASCII-Dateien. Die Kodierung ist selbstsynchronisie-

rend, das heißt, Programme zum Lesen des Codes können leicht herausfinden, wo im ASCII-Text UTF-8 Zeichen beginnen und enden. Normale Funktionen zum Sortieren oder Zeichenvergleiche funktionieren auch mit UTF-8.

Tabelle B.3: UTF-8-Kodierung

Bytes	Bits	Darstellung
1	7	0bbbbbbb
2	11	110bbbb 10bbbbbb
3	16	1110bbbb 10bbbbbb 10bbbbbb
4	21	11110bbb 10bbbbbb 10bbbbbb 10bbbbbb

Jedes b repräsentiert ein Bit und kann zur Speicherung von Zeichen genutzt werden.

PDF-Funktionen

pdf_get_info

pdf_get_info

pdf_get_info gibt die Standardstruktur eines PDF-Dokuments aus.



```
info pdf_get_info(string filename);
```

Siehe auch pdf_set_info_creator, pdf_set_info_author, pdf_set_info_keywords, pdf_set_info_title, pdf_set_info_subject.

pdf_set_info_creator

pdf_set_info_creator

pdf_set_info_creator setzt den Wert für *creator* in einem PDF-Dokument.



```
void pdf_set_info_creator(info info, string creator);
```

Die Funktion muss nach pdf_get_info und vor pdf_open aufgerufen werden, andernfalls hat sie keinen Effekt.

Siehe auch pdf_get_info, pdf_set_info_keywords, pdf_set_info_title, pdf_set_info_subject.

pdf_set_info_title

pdf_set_info_title

pdf_set_info_title setzt den Wert für *title* in einem pdf-Dokument.



```
void pdf_set_info_title(info info, string title);
```

Die Funktion muss nach pdf_get_info und vor pdf_open aufgerufen werden, andernfalls hat sie keinen Effekt.

Siehe auch pdf_get_info, pdf_set_info_creator, pdf_set_info_author, pdf_set_info_keywords, pdf_set_info_subject.

pdf_set_info_subject

pdf_set_info_subject setzt den Wert für *subject* in einem PDF-Dokument.

```
void pdf_set_info_subject(info info, string subject);
```

Die Funktion muss nach pdf_get_info und vor pdf_open aufgerufen werden, andernfalls hat sie keinen Effekt.

Siehe auch pdf_get_info, pdf_set_info_creator, pdf_set_info_author, pdf_set_info_title, pdf_set_info_keywords.

pdf_set_info_subject



pdf_set_info_keywords

pdf_set_info_keywords setzt den Wert für *keywords* in einem pdf-Dokument.

```
void pdf_set_info_keywords(info info, string keywords);
```

Die Funktion muss nach pdf_get_info und vor pdf_open aufgerufen werden, andernfalls hat sie keinen Effekt.

Siehe auch pdf_get_info, pdf_set_info_creator, pdf_set_info_author, pdf_set_info_title, pdf_set_info_subject.

pdf_set_info_keywords



pdf_set_info_author

pdf_set_info_author setzt den Wert für *author* in einem PDF-Dokument.

```
void pdf_set_info_author(info info, string author);
```

Die Funktion muss nach pdf_get_info und vor pdf_open aufgerufen werden, andernfalls hat sie keinen Effekt.

Siehe auch pdf_get_info, pdf_set_info_creator, pdf_set_info_keywords, pdf_set_info_title, pdf_set_info_subject.

pdf_set_info_author



pdf_open

pdf_open öffnet ein neues PDF-Dokument.

```
int pdf_open(int file, int info);
```

Das Dokument ist zuvor physisch mit fopen zu öffnen und das Dateihandle wird als Argument *file* genutzt.

pdf_open



info ist die mit `pdf_get_info` erzeugte Dokumentenstruktur.

Siehe auch `fopen`, `pdf_get_info`, `pdf_close`.

pdf_close

pdf_close `pdf_close` schließt ein PDF-Dokument. Sie müssen die Datei außerdem noch mit `fclose` physisch schließen.

◆ `void pdf_close(int pdf_document);`

Siehe auch `pdf_open`, `fclose`.

pdf_begin_page

pdf_begin_page `pdf_begin_page` erzeugt eine neue Seite in einem PDF-Dokument. Der Parameter *height* gibt die Höhe der Seite an, *width* die Breite.

◆ `void pdf_begin_page(int pdf_document, double height,
double width);`

Siehe auch `pdf_end_page`.

pdf_end_page

pdf_end_page `pdf_end_page` beendet eine Seite. Danach kann die Seite nicht mehr verändert werden

◆ `void pdf_end_page(int pdf_document);`

Siehe auch `pdf_begin_page`.

pdf_show

pdf_show `pdf_show` gibt Text an der aktuellen Stelle in das PDF-Dokument aus.

◆ `void pdf_show(int pdf_document, string text);`

Siehe auch `pdf_show_xy`, `pdf_set_text_pos`, `pdf_set_font`.

pdf_show_xy

pdf_show_xy `pdf_show_xy` gibt Text an der bezeichneten Stelle in das PDF-Dokument aus.

◆ `void pdf_show_xy(int pdf_document, string text,
double x-koor, double y-koor);`

Siehe auch `pdf_show`.

pdf_set_font

pdf_set_font setzt den für die nächste Ausgabe verwendeten Font und die Größe.

pdf_set_font

```
void pdf_set_font(int pdf_document, string font name,  
                 double size,int encoding);
```



Als Font müssen Adobe Font Metrics-Dateien (afm-Dateien) vorhanden sein. Geben Sie einen vollständigen Pfad an. Der Parameter *encoding* kann sein:

- 0 = builtin
- 1 = pdfdoc
- 2 = macroman
- 3 = macexpert
- 4 = winansi. Andere Werte werden auch als »winansi« interpretiert.

pdf_set_leading

pdf_set_leading setzt den Zeilenabstand im Zusammenhang mit pdf_continue_text.

pdf_set_leading

```
void pdf_set_leading(int pdf_document, double distance);
```



Siehe auch pdf_continue_text.

pdf_set_text_rendering

pdf_set_text_rendering setzt die Textbehandlung für das Dokument *pdf_document*.

pdf_set_text_rendering

```
void pdf_set_text_rendering(int pdf_document, int mode);
```



mode kann sein:

- 0 = »fill text«. Gefüllt.
- 1 = »stroke text«. Geglättet.
- 2 = »fill and stroke text«. Gefüllt und geglättet.
- 3 = »invisible«. Unsichtbar.
- 4 = »fill text and add it to clipping path«. Gefüllt am Zeichenpfad.

- 5 = »stroke text and add it to clipping path«. Geglättet und gefüllt am Zeichenpfad.
- 6 = »fill and stroke text and add it to clipping path«. Gefüllt, geglättet und am Zeichenpfad.
- 7 = »add it to clipping path«. Nur am Zeichenpfad.

pdf_set_horiz_scaling

pdf_set_horiz_scaling pdf_set_horiz_scaling setzt das horizontale Skalieren des Textes in Prozent.

◆ void pdf_set_horiz_scaling(int pdf_document, double scale);

pdf_set_text_rise

pdf_set_text_rise pdf_set_text_rise setzt die Höhe, um die Text angehoben werden soll.

◆ void pdf_set_text_rise(int pdf_document, double value);

pdf_set_text_matrix

pdf_set_text_matrix pdf_set_text_matrix setzt eine Matrix, mit der der aktuelle Font gewandelt werden kann. Die Matrix ist ein Array mit sechs Elementen.

◆ void pdf_set_text_matrix(int pdf_document, array matrix);

pdf_set_text_pos

pdf_set_text_pos pdf_set_text_pos setzt die Textposition für den nächsten Aufruf von pdf_show.

◆ void pdf_set_text_pos(int pdf_document,
double x-koor, double y-koor);

Siehe auch pdf_show, pdf_show_xy.

pdf_set_char_spacing

pdf_set_char_spacing Die Funktion pdf_set_char_spacing setzt den Abstand von Zeichen auf den Wert *space*.

◆ void pdf_set_char_spacing(int pdf_document, double space);

Siehe auch pdf_set_word_spacing, pdf_set_leading.

pdf_set_word_spacing

pdf_set_word_spacing setzt den Wortabstand im Dokument *pdf_document* auf den Wert *space* fest.

```
void pdf_set_word_spacing(int pdf_document, double space);
```

Siehe auch pdf_set_char_spacing, pdf_set_leading.

pdf_set_word_spacing**pdf_continue_text**

pdf_continue_text gibt Text in die nächste Zeile aus. Der Abstand der Zeilen wird mit pdf_set_leading gesetzt.

```
void pdf_continue_text(int pdf_document, string text);
```

Siehe auch pdf_show_xy, pdf_set_leading, pdf_set_text_pos.

pdf_continue_text**pdf_stringwidth**

pdf_stringwidth setzt die Textbreite für den aktuellen Font. Der Font muss zuvor bestimmt worden sein.

```
double pdf_stringwidth(int pdf_document, string text);
```

Siehe auch pdf_set_font.

pdf_stringwidth**pdf_save**

pdf_save sichert die aktuelle Entwicklungsumgebung, ähnlich dem Postscript-Kommando *gsave*. Damit werden Änderungen an zuvor platzierten Objekten verhindert. Die Funktion sollte von pdf_restore gefolgt werden.

```
void pdf_save(int pdf_document);
```

Siehe auch pdf_restore.

pdf_save**pdf_restore**

pdf_restore holt die ursprüngliche Entwicklungsumgebung wieder zurück, ähnlich dem Postscript-Kommando *grestore*.

```
void pdf_restore(int pdf_document);
```

Siehe auch pdf_save.

pdf_restore**pdf_translate**

pdf_translate legt die Ursprungskoordinaten fest.

pdf_translate

◆ `void pdf_translate(int pdf_document,
double x-koor, double y-koor);`

Das folgende Beispiel zeigt, wie eine Linie auf der Seite mit Hilfe dieser Funktion erzeugt werden kann.



```
<?php pdf_moveto($pdf, 0, 0);
pdf_lineto($pdf, 100, 100);
pdf_stroke($pdf);
pdf_translate($pdf, 100, 100);
pdf_moveto($pdf, 0, 0);
pdf_lineto($pdf, 100, 100);
pdf_stroke($pdf);
?>
```

pdf_scale

pdf_scale pdf_scale setzt die Skalierung der internen Einheit. Die interne Einheit ist ein $1/72$ tel Zoll.

◆ `void pdf_scale(int pdf_document,
double x-scale, double y-scale);`

Das folgende Beispiel setzt die Skalierung auf 1 Zoll und zeichnet eine Linie mit der Länge 1 Zoll.



```
<?php pdf_scale($pdf, 72.0, 72.0);
pdf_lineto($pdf, 1, 1);
pdf_stroke($pdf);
?>
```

pdf_rotate

pdf_rotate pdf_rotate setzt einen Winkel, um den nachfolgende Objekte rotiert werden.

◆ `void pdf_rotate(int pdf_document, double angle);`

pdf_setflat

pdf_setflat pdf_setflat setzt die Abgeflachtheit auf einen Wert zwischen 0 und 100.

◆ `void pdf_setflat(int pdf_document, double value);`

pdf_setlinejoin

pdf_setlinejoin pdf_setlinejoin setzt einen Wert der bestimmt, wie Linien miteinander verbunden werden. Der Wert kann 0, 1 oder 2 sein.

◆ `void pdf_setlinejoin(int pdf_document, long value);`

pdf_setlinecap

pdf_setlinecap setzt den Wert für die Bestimmung der Form der Linienenden. Der Wert kann 0, 1 oder 2 sein.

```
void pdf_setlinecap(int pdf_document, int value);
```

pdf_setlinecap**pdf_setmiterlimit**

pdf_setmiterlimit setzt den Parameter *value* auf einen Wert größer oder gleich 1.

```
void pdf_setmiterlimit(int pdf_document, double value);
```

pdf_setmiterlimit**pdf_setlinewidth**

pdf_setlinewidth setzt die Breite einer Linie auf den Wert *width* fest.

```
void pdf_setlinewidth(int pdf_document, double width);
```

pdf_setlinewidth**pdf_setdash**

pdf_setdash setzt das Muster.

```
void pdf_setdash(int pdf_document, double white, double black);
```

pdf_setdash**pdf_moveto**

pdf_moveto setzt einen Punkt.

```
void pdf_moveto(int pdf_document, double x-koor, double y-koor);
```

pdf_moveto**pdf_curveto**

pdf_curveto zeichnet eine Bezier-Kurve vom aktuellen Punkt zu x_3 , y_3 mit x_1 , y_1 und x_2 , y_2 als Kontrollpunkt.

```
void pdf_curveto(int pdf_document,  
                double x1, double y1,  
                double x2, double y2,  
                double x3, double y3);
```

pdf_curveto

Siehe auch pdf_moveto, pdf_lineto, pdf_stroke.

pdf_lineto

pdf_lineto zeichnet eine Linie vom aktuellen Punkt zu den Koordinaten *x-koor* und *y-koor*.

```
void pdf_lineto(int pdf_document,  
               double x-koor, double y-koor);
```

pdf_lineto

Siehe auch pdf_moveto, pdf_curveto, pdf_stroke.

pdf_circle

pdf_circle pdf_circle zeichnet einen Kreis mit dem Mittelpunkt *x-koor* und *y-koor* und dem Radius *radius*.

◆

```
void pdf_circle(int pdf_document,
                double x-koor, double y-koor,
                double radius);
```

Siehe auch pdf_arc, pdf_stroke.

pdf_arc

pdf_arc pdf_arc zeichnet einen Kreisbogen um den Mittelpunkt *x-koor* und *y-koor* und dem Radius *radius* vom Startwinkel *start* bis zum Endwinkel *end*.

◆

```
void pdf_arc(int pdf_document,
             double x-koor, double y-koor,
             double radius, double start, double end);
```

Siehe auch pdf_circle, pdf_stroke.

pdf_rect

pdf_rect pdf_rect zeichnet ein Rechteck von den Koordinaten *x-koor* und *y-koor* und mit der Breite *width* und der Höhe *height*.

◆

```
void pdf_rect(int pdf_document,
              double x-koor, double y-koor,
              double width, double height);
```

Siehe auch pdf_stroke.

pdf_closepath

pdf_closepath pdf_closepath schließt einen Zeichenpfad ab. Damit wird das aktuelle Ende des Pfades mit dem Anfang verbunden. Viele Funktionen wie pdf_moveto, pdf_circle und pdf_rect starten einen neuen Pfad. Die Anwendung ist vor allem mit Linien sinnvoll.

◆

```
void pdf_closepath(int pdf_document);
```

pdf_stroke

pdf_stroke pdf_stroke zeichnet eine Linie an einem Pfad entlang. Der Pfad ergibt sich aus allen Linienzeichenbefehlen. Ohne pdf_stroke würde keine Linie erscheinen.

```
void pdf_stroke(int pdf_document);
```



Siehe auch pdf_closepath, pdf_closepath_stroke.

pdf_closepath_stroke

pdf_closepath_stroke schließt den Pfad und zeichnet die zuvor bereits generierte Linie.

pdf_closepath_stroke

```
void pdf_closepath_stroke(int pdf_document);
```



Siehe auch pdf_closepath, pdf_stroke.

pdf_fill

pdf_fill füllt den aktuellen Pfad. Es wird die aktuelle Füllfarbe verwendet. Der Pfad muss geschlossen sein, sonst wird die ganze Seite gefüllt.

pdf_fill

```
void pdf_fill(int pdf_document);
```



Siehe auch pdf_closepath, pdf_stroke, pdf_setgray_fill, pdf_setgray, pdf_setrgbcolor_fill, pdf_setrgbcolor.

pdf_fill_stroke

pdf_fill_stroke füllt und zeichnet den aktuellen Pfad.

pdf_fill_stroke

```
void pdf_fill_stroke(int pdf_document);
```



Siehe auch pdf_closepath, pdf_stroke, pdf_fill, pdf_setgray_fill, pdf_setgray, pdf_setrgbcolor_fill, pdf_setrgbcolor.

pdf_closepath_fill_stroke

pdf_closepath_fill_stroke füllt, zeichnet und schließt den aktuellen Pfad.

pdf_closepath_

```
void pdf_closepath_fill_stroke(int pdf_document);
```



Siehe auch pdf_closepath, pdf_stroke, pdf_fill, pdf_setgray_fill, pdf_setgray, pdf_setrgbcolor_fill, pdf_setrgbcolor.

pdf_endpath

pdf_endpath beendet den aktuellen offenen Pfad.

pdf_endpath

```
void pdf_endpath(int pdf_document);
```



Siehe auch pdf_closepath.

pdf_clip

pdf_clip pdf_clip schneidet alles aus dem aktuellen Pfad aus.

◆ void **pdf_clip**(int pdf_document);

pdf_setgray_fill

pdf_setgray_fill pdf_setgray_fill setzt die Füllfarbe auf eine Graustufe.

◆ void **pdf_setgray_fill**(int pdf_document, double value);

Siehe auch pdf_setrgbcolor_fill.

pdf_setgray_stroke

pdf_setgray_stroke Die Funktion pdf_setgray_stroke setzt die Zeichenfarbe auf den Grauwert *gray_value*.

◆ void **pdf_setgray_stroke**(int pdf_document,
double gray_value);

Siehe auch pdf_setrgbcolor_stroke.

pdf_setgray

pdf_setgray pdf_setgray setzt die Zeichen- und Füllfarbe auf eine Graustufe.

◆ void **pdf_setgray**(int pdf_document, double gray value);

Siehe auch pdf_setrgbcolor_stroke, pdf_setrgbcolor_fill.

pdf_setrgbcolor_fill

pdf_setrgbcolor_fill pdf_setrgbcolor_fill setzt die Füllfarbe auf eine RGB-Farbe, wobei jeder Farbwert einzeln angegeben wird.

◆ void **pdf_setrgbcolor_fill**(int pdf_document,
double red value,
double green value,
double blue value);

Siehe auch pdf_setrgbcolor_fill.

pdf_setrgbcolor_stroke

pdf_setrgbcolor_stroke pdf_setrgbcolor_stroke setzt die Zeichenfarbe auf einen RGB-Wert, wobei jeder Farbwert einzeln angegeben wird.

◆ void **pdf_setrgbcolor_stroke**(int pdf_document,
double red value,

```
double green value,  
double blue value);
```

Siehe auch pdf_setrgbcolor_stroke.

pdf_setrgbcolor

pdf_setrgbcolor setzt Füll- und Zeichenfarbe auf einen RGB-Wert.

```
void pdf_setrgbcolor(int pdf_document,  
double red value,  
double green value,  
double blue value);
```

Siehe auch pdf_setrgbcolor_stroke, pdf_setrgbcolor_fill.

pdf_add_outline

pdf_add_outline setzt ein Lesezeichen auf die aktuelle Seite.

```
void pdf_add_outline(int pdf_document, string text);
```

Der Parameter *text* bestimmt den Inhalt des Lesezeichens.

pdf_set_transition

Die Funktion pdf_set_transition setzt einen Wert für den Übergang zwischen Seiten.

```
void pdf_set_transition(int pdf_document, int transition);
```

Verschiedene Effekte bestimmen, wie der Seitenwechsel stattfindet. Der Parameter *transition* kann sein:

- 0 für keinen sichtbaren Übergang.
- 1 für »Zwei Linien laufen über die Seite«.
- 2 für »Mehrere Linien laufen über die Seite«.
- 3 für »Ein Rechteck bringt die neue Seite hervor«.
- 4 für »Eine Linie wandert über den Bildschirm, um die neue zum Vorschein zu bringen«.
- 5 für »Die alte Seite verschwindet um die neue zum Vorschein zu bringen«.
- 6 für »Der Übergangseffekt bewegt sich von einer Ecke zur anderen«.

- 7 für »Die alte Seite wird durch die neue ersetzt (Standardwert)«.

pdf_set_duration

pdf_set_duration pdf_set_duration setzt eine Verzögerung zwischen Seiten.



```
void pdf_set_duration(int pdf_document, double duration);
```

pdf_open_gif

pdf_open_gif pdf_open_gif öffnet ein GIF-Bild.



```
int pdf_open_gif(int pdf_document, string file name);
```

Siehe auch pdf_close_image, pdf_open_jpeg, pdf_open_memory_image, pdf_execute_image, pdf_place_image, pdf_put_image.

pdf_open_memory_image

pdf_open_memory_image pdf_open_memory_image öffnet ein von PHP erzeugtes Bild und macht es für ein pdf-Dokument verfügbar.



```
int pdf_open_memory_image(int pdf_document, string int image);
```

Siehe auch pdf_close_image, pdf_open_jpeg, pdf_open_gif, pdf_execute_image, pdf_place_image, pdf_put_image.

pdf_open_jpeg

pdf_open_jpeg pdf_open_jpeg öffnet ein JPEG-Bild.



```
int pdf_open_jpeg(int pdf_document, string file name);
```

Siehe auch pdf_close_image, pdf_open_gif, pdf_open_memory_image, pdf_execute_image, pdf_place_image, pdf_put_image.

pdf_close_image

pdf_close_image pdf_close_image schließt ein Bild, das durch eine der pdf_open_xxx-Funktionen geöffnet wurde.



```
void pdf_close_image(int image);
```

Siehe auch pdf_open_jpeg, pdf_open_gif, pdf_open_memory_image.

pdf_place_image

pdf_place_image pdf_place_image platziert ein Bild auf einer Seite.

```
void pdf_place_image(int pdf_document, int image,  
                    double x-coor, double y-coor,  
                    double scale);
```



Die Funktion platziert ein Bild an den Koordinaten *x-coor*, *y-coor*. Das Bild kann gleichzeitig skaliert werden um den Faktor *scale*. Eine Skalierung von 1,0 lässt das Bild in der Originalgröße.

Siehe auch pdf_put_image.

pdf_put_image

pdf_put_image speichert ein Bild im Dokument für die spätere Verwendung.

pdf_put_image

```
void pdf_put_image(int pdf_document, int image);
```



Gespeicherte Bilder können mit der Funktion pdf_execute_image wieder sichtbar gemacht werden. Die Funktion ist sinnvoll, wenn ein und dasselbe Bild im Dokument mehrfach platziert werden soll, beispielsweise bei Icons.

Siehe auch pdf_place_image, pdf_execute_image.

pdf_execute_image

Mit der Funktion pdf_execute_image wird ein gespeichertes Bild im Dokument platziert.

pdf_execute_image

```
void pdf_execute_image(int pdf_document, int image,  
                      double x-coor, double y-coor,  
                      double scale);
```



Die Funktion platziert ein Bild an den Koordinaten *x-coor*, *y-coor*. Das Bild kann gleichzeitig skaliert werden um den Faktor *scale*. Eine Skalierung von 1,0 belässt das Bild in der Originalgröße.

Shockwave-Flash-Funktionen

Die Shockwave-Flash-Funktionen sind ab PHP4 verfügbar. Sie dienen der Erzeugung von SWF-Dateien, die im Flash-Player abgespielt werden können. Der Flash-Player ist ein Browser-Plug-In, das Sie unter www.macromedia.com kostenlos beziehen können. Die Bibliothek für PHP finden Sie unter der folgenden Adresse:

- <http://reality.sgi.com/grafica/flash>

Laden Sie die Bibliothek und kompilieren Sie dann PHP mit dem folgenden Parameter:

```
--with-swf[=<DIR>]
```

Ersetzen Sie <DIR> durch den Pfad zu der Bibliothek.

swf_openfile

swf_openfile

Die Funktion öffnet eine neue Shockwave-Datei. Diese Funktion muss immer zuerst aufgerufen werden, da die anderen Funktionen nur in eine bereits existierende Datei schreiben können.

```
void swf_openfile(string filename, float width, float height,
                  float framerate, float r, float g, float b)
```

filename ist der Name der Shockwave-Datei (kein Handle), width und height geben die Größe des Objekts in Pixeln an. framerate gibt die Wiederholrate an, die übrigen Parameter die Hintergrundfarbe im RGB-Format.

Alle Funktionen schreiben automatisch in die zuletzt geöffnete Datei. Sie können deshalb immer nur eine SWF-Datei beschreiben.

Siehe auch swf_closefile

swf_closefile

swf_closefile

Diese Funktion schließt die aktuelle, zuvor geöffnete SWF-Datei.

```
void swf_closefile(void)
```

Siehe auch swf_openfile.

swf_labelframe

swf_labelframe

Das aktuelle Frame bekommt eine Bezeichnung.

```
void swf_labelframe(string name)
```

swf_showframe

swf_showframe

Die Funktion gibt das aktuelle Frame aus.

```
void swf_showframe(void)
```

swf_setframe

swf_setframe

Die Funktion wechselt zu dem mit framenummer ausgewählten Frame.

```
void swf_setframe(int framenummer)
```

swf_getframe

Die Funktion ermittelt die Nummer des aktuellen Frames.

```
int swf_getframe(void)
```

swf_getframe**swf_mulcolor**

Setzt die »global-multiply«-Farbe auf den angegebenen Farbwert.

```
void swf_mulcolor(float r, float g, float b, float a)
```

swf_mulcolor

Die Funktion wird implizit durch `swf_placeobject`, `swf_modifyobject` und `swf_addbuttonrecord` aufgerufen.

swf_addcolor

Setzt die »global-add«-Farbe auf den angegebenen Farbwert.

```
void swf_addcolor(float r, float g, float b, float a)
```

swf_addcolor

Die Funktion wird implizit durch `swf_placeobject`, `swf_modifyobject` und `swf_addbuttonrecord` aufgerufen.

swf_placeobject

Die Funktion platziert ein Objekt auf dem Bildschirm.

```
void swf_placeobject(int objid, int depth)
```

swf_placeobject

Die Parameter für die Objekt-ID (`objid`) und die Tiefe (`depth`) müssen zwischen 1 und 65 535 liegen.

swf_modifyobject

Diese Funktion modifiziert die aktuelle Position oder Farbe des aktuellen Objekts.

```
void swf_modifyobject(int depth, int how)
```

swf_modifyobject

Der Parameter `how` entscheidet, wie `depth` interpretiert wird:

- `MOD_MATRIX`. Position
- `MOD_COLOR`. Farbe

Die beiden Konstanten können mit `|` (oder) kombiniert werden.

swf_removeobject

Diese Funktion entfernt das Objekt auf der Ebene `depth`:

swf_removeobject

	void swf_removeobject (int depth)
◆	swf_nextid
swf_nextid	Die Funktion gibt die nächste frei verfügbare Objekt-ID zurück.
◆	int swf_nextid (void)
	swf_startdoaction
swf_startdoaction	Die Funktion startet eine Beschreibung einer Aktion im aktuellen Frame. Sie müssen diese Funktion zuerst aufrufen, bevor irgendwelche Aktionen im Frame definiert werden.
◆	void swf_startdoaction (void)
	swf_enddoaction
swf_enddoaction	Die Funktion beendet den aktuell laufenden Film im aktuellen Frame.
◆	void swf_startdoaction (void)
	swf_actiongotoframe
swf_actiongotoframe	Mit dieser Funktion spielen Sie einen gespeicherten Film im ausgewählten Frame ab. Danach stoppt die Ausführung.
◆	void swf_actiongotoframe (int framenummer)
	Siehe auch <code>swf_actionplay</code>
	swf_actiongeturl
swf_actiongeturl	Die Funktion ermittelt die URL aus einem Shockwave-Film mit der Adresse <code>url</code> und dem Ziel <code>target</code> .
◆	void swf_actiongeturl (string url, string target)
	swf_actionnextframe
swf_actionnextframe	Mit dieser Funktion wird ein Frame weiter geschaltet. Die Frames werden durch einen internen Zeiger adressiert.
◆	void swf_actionnextframe (void)
	swf_actionprevframe
swf_actionprevframe	Mit dieser Funktion wird ein Frame zurück geschaltet. Die Frames werden durch einen internen Zeiger adressiert.

void **swf_actionnextframe**(void)



swf_actionplay

Die Funktion spielt einen Film aus dem aktuellen Frame ab.

swf_actionplay

void **swf_actionplay**(void)



swf_actionstop

Die Funktion stoppt den Film aus dem aktuellen Frame.

swf_actionstop

void **swf_actionstop**(void)



swf_actiontogglequality

Mit dieser Funktion wird die Qualität des laufenden Films zwischen guter und schlechter Qualität umgeschaltet.

swf_actiontogglequality

void **swf_actiontogglequality**(void)



swf_actionwaitforframe

Die Funktion überspringt Aktionen, wenn ein Frame nicht geladen wurde. Zuerst wird überprüft, ob das angegebene Frame geladen ist. Wenn das nicht der Fall ist, wird die Anzahl *skipcount* Aktionen übersprungen.

swf_actionwaitforframe

void **swf_actionwaitforframe**(int framenummer, int skipcount)



swf_actionsettarget

Mit dieser Funktion setzen Sie das Ziel für Aktionen. Damit können Sie mehrere Flash-Filme kontrollieren.

swf_actionsettarget

void **swf_actionsettarget**(string target)



swf_actiongotolabel

Diese Funktion zeigt einen mit der angegebenen Marke *label* markierten Frame an.

swf_actiongotolabel

void **swf_actiongotolabel**(string label)



swf_defineline

Zeichnet eine Linie mit den angegebenen Parametern.

swf_defineline

void **swf_defineline**(int objid, float x1, float y1,
float x2, float y2, float width)



Die Koordinaten geben Anfangs- und Endpunkt sowie die Breite an.

swf_definerect

swf_definerect Zeichnet ein Rechteck mit den angegebenen Parametern.

◆
 void **swf_definerect**(int objid, float x1, float y1,
 float x2, float y2, float width)

Die Koordinaten geben die linke obere und die rechte untere Ecke sowie die Breite der Umrandung an.

swf_startshape

swf_startshape Die Funktion startet eine komplexe Form im angegebenen Objekt.

◆
 void **swf_startshape**(int objid)

swf_definepoly

swf_definepoly Die Funktion zeichnet ein Polygon, dessen Koordinaten in einem Array übergeben werden.

◆
 void **swf_definepoly**(int objid, array coords,
 int npoints, float width)

swf_shapelinesolid

swf_shapelinesolid Die Funktion bestimmt den Stil für Linien. Die nächste gezeichnete Linie erscheint mit diesen Attributen.

◆
 void **swf_shapelinesolid** (float r, float g, float b, float a,
 float width)

Angegeben wird die Farbe im RGBA-Format und die Breite.

swf_shapefilloff

swf_shapefilloff Schaltet das Ausfüllen für Formen aus. Nach Ausführung werden Formen nicht mehr gefüllt.

◆
 void **swf_shapefilloff**(void)

Siehe auch `shapefillsolid`.

swf_shapefillsolid

swf_shapefillsolid Diese Funktion bestimmt, wie Formen gefüllt werden (RGBA-Wert).

```
void swf_shapefillsolid(float r, float g, float b, float a)
```



swf_shapefillbitmapclip

Mit dieser Funktion wird ein Bitmap bestimmt, mit dem leere Flächen einer Form gefüllt werden.

swf_shapefillbit
mapclip

```
void swf_shapefillbitmapclip(int bitmapid)
```



swf_shapefillbitmaptile

Mit dieser Funktion wird ein Bitmap bestimmt, mit dem leere Flächen einer Form im Kachelmode gefüllt werden.

swf_shapefillbit
maptile

```
void swf_shapefillbitmaptile(int bitmapid)
```



swf_shapemoveto

Diese Funktion verschiebt die Koordinaten des Zeichenwerkzeugs auf die angegebene Position.

swf_shapemoveto

```
void swf_shapemoveto(float x, float y)
```



swf_shapelineto

Zeichnet eine Linie von der aktuellen Position zu den angegebenen Koordinaten. Der Endpunkt wird dann zum aktuellen Punkt.

swf_shapelineto

```
void swf_shapelineto(float x, float y)
```



swf_shapecurveto

Die Funktion zeichnet eine Bezier-Kurve von den aktuellen Koordinaten zu den angegebenen und setzt den aktuellen Zeiger auf den Endpunkt.

swf_shapecurveto

```
void swf_shapecurveto(float x1, float y1, float x2, float y2)
```



swf_shapecurveto3

Die Funktion zeichnet eine kubische Bezier-Kurve von den aktuellen Koordinaten zu den angegebenen und setzt den aktuellen Zeiger auf den Endpunkt.

swf_
shapecurveto3

```
void swf_shapecurveto3(float x1, float y1,  
                      float x2, float y2,  
                      float x3, float y3)
```



swf_shapearc

swf_shapearc Die Funktion zeichnet einen Kreisbogen. Die Koordinaten x und y geben den Mittelpunkt des Kreises an.

◆
void **swf_shapearc**(float x, float y,
float r, float angl, float ang2)

swf_endshape

swf_endshape Die Funktion schließt die Definition der aktuell erstellten Form ab.

◆
void **swf_endshape**(void)

swf_definefont

swf_definefont Die Funktionen definiert für einen angegebenen Font eine Font-ID und setzt diesen Font als aktuellen Font. Zukünftige Ausgaben werden den Font verwenden.

◆
void **swf_definefont**(int fontid, string fontname)

swf_setfont

swf_setfont Die Funktion wählt einen zuvor mit **swf_definefont** definierten Font anhand der Font-ID aus.

◆
void **swf_setfont**(int fontid)

swf_fontsize

swf_fontsize Setzt die Größe für den aktuellen Font.

◆
void **swf_fontsize**(float size)

swf_fontslant

swf_fontslant Setzt die Schräge des aktuellen Fonts im angegebenen Winkel. Positive Werte stellen die Schrift nach rechts (kursiv), negative Werte nach links.

◆
void **swf_fontslant**(float slant)

swf_fontracking

swf_fontracking Die Funktion bestimmt den Abstand der Buchstaben im Fließtext. Der aktuelle Wert wird nicht absolut gesetzt, sondern vergrößert (positiver Wert) oder verringert (negativer Wert).

void **swf_fontracking**(float tracking)

swf_getfontinfo

Die Funktion gibt ein assoziatives Array mit Angaben über den aktuellen Font zurück.

array **swf_getfontinfo**(void)

- "Aheight": Höhe des Großbuchstaben A
- "xheight": Höhe des kleinen Buchstaben x

swf_definetext

Schreibt einen Text in das aktuelle Objekt.

void **swf_definetext**(int objid, string str, int docenter)

Wenn docenter TRUE ist, wird der Text zentriert.

swf_textwidth

Mit dieser Funktion wird die Breite des in str übergebenen Textes mit den aktuellen Font-Parametern berechnet. Der Text wird aber nicht geschrieben.

float **swf_textwidth**(string str)

swf_definebitmap

Die Funktion definiert ein Bitmap. Akzeptiert werden GIF, JPEG, RGB oder FI.

void **swf_definebitmap**(int objid, string image_name)

swf_getbitmapinfo

Die Funktion ermittelt die Attribute des angegebenen Bitmaps. Zurückgegeben wird ein assoziatives Array.

array **swf_getbitmapinfo**(int bitmapid)

- "size". Größe in Byte
- "width". Breite in Pixel
- "height". Höhe in Pixel

swf_startsymbol

swf_startsymbol Die Funktion startet die Definition eines Symbols.

◆ `void swf_startsymbol(int objid)`

Symbole sind kleine Flash-Filme, die ununterbrochen abgespielt werden.

swf_endsymbol

swf_endsymbol Mit dieser Funktion wird die Definition eines Symbols beendet.

◆ `void swf_endsymbol(void)`

swf_startbutton

swf_startbutton Startet die Definition einer Schaltfläche mit den angegebenen Parametern.

◆ `void swf_startbutton(int objid, int type)`

Möglich sind zwei Arten von Schaltflächen:

- TYPE_MENUBUTTON. Wenn die Maustaste gedrückt ist, kann der Fokus entfernt werden.
- TYPE_PUSHBUTTON. Wenn die Maustaste gedrückt ist, bleibt der Fokus auf der Schaltfläche.

swf_addbuttonrecord

swf_addbuttonrecord Diese Funktion steuert die Eigenschaften (mögliche Zustände) der aktuellen Schaltfläche.

◆ `void swf_addbuttonrecord(int states, int shapeid, int depth)`

Der Parameter state bestimmt, welche Zustände die Schaltfläche haben darf. Mehrere Konstanten können mit | (oder) kombiniert werden:

- BSHitTest. Gedrückt, während Maustaste gedrückt, aber ohne Aktion, wenn die Maustaste außerhalb der Schaltzone auslöst.
- BSDown. Gedrückt
- BSOver. Maus schwebt über der Schaltfläche.
- BSUp. Oben

shapeid ist die ID einer Form, die bestimmt, wie der Schalter aussieht. Mit depth wird der Schalter auf einem bestimmten Frame platziert.

Das folgende Beispiel zeigt, wie eine Schaltfläche erstellt wird:

```
function define_swf_button($objid, $btnImgID, $acturl, $frame) {
    swf_startButton ($objid, TYPE_MENUBUTTON);
    swf_addButtonRecord (BSDown|BSOver, $btnImgId, $frame);
    swf_onCondition (MenuEnter);
    swf_actionGetUrl ("http://$acturl", "_level1");
    swf_onCondition (MenuExit);
    swf_actionGetUrl ("", "_level1");
    swf_endButton ();
}
```



Beispiel

swf_oncondition

Die Funktion definiert eine Aktion, die ausgelöst wird, wenn das spezifizierte Ereignis eintritt.

swf_oncondition

void **swf_oncondition**(int transition)



Der Parameter transition kann je nach Einstellung des Parameters type folgende Werte annehmen:

- TYPE_MENUBUTTON
 - IdletoOverUp
 - OverUptoIdle
 - OverUptoOverDown
 - OverDowntoOverUp
 - IdletoOverDown
 - OutDowntoIdle
 - MenuEnter (IdletoOverUp | IdletoOverDown)
 - MenuExit (OverUptoIdle | OverDowntoIdle)
- TYPE_PUSHBUTTON
 - IdletoOverUp
 - OverUptoIdle
 - OverUptoOverDown
 - OverDowntoOverUp
 - OverDowntoOutDown

- OutDowntoOverDown
- OutDowntoIdle
- ButtonEnter (IdletoOverUp | OutDowntoOverDown)
- ButtonExit (OverUptoIdle | OverDowntoOutDown)

swf_endbutton

swf_endbutton Schließt die Definition einer Schaltfläche ab.

◆ void **swf_endbutton**(void)

swf_viewport

swf_viewport Mit dieser Funktion wird ein Bereich auf der Zeichenfläche definiert, in dem künftig Ausgaben erfolgen.

◆ void **swf_viewport**(double xmin, double xmax,
double ymin, double ymax)

Wenn diese Funktion nicht aufgerufen wird, ist die gesamte Zeichenfläche verwendbar.

swf_ortho

swf_ortho Die Funktion definiert eine orthografische Sicht auf den aktuellen Sichtbereich.

◆ void **swf_ortho**(double xmin, double xmax,
double ymin, double ymax,
double zmin, double zmax)

Hinweis: Der aktuelle Sichtbereich wird mit `swf_viewport` definiert.

swf_ortho2

swf_ortho2 Die Funktion definiert eine 2D-Sicht auf den Sichtbereich.

◆ void **swf_ortho2**(double xmin, double xmax,
double ymin, double ymax)

swf_perspective

swf_perspective Transformiert Koordinaten perspektivisch.

◆ void **swf_perspective**(double fovy, double aspect,
double near, double far)

fovy bestimmt den Winkel in y-Richtung, *aspect* zeigt das Verhältnis an. *near* gibt die nähere Ebene an, *far* die entfernte.

swf_polarview

Die Funktion definiert die Position des Betrachters mit Polarkoordinaten.

swf_polarview

```
void swf_polarview(double dist, double azimuth,  
                  double incidence, double twist)
```



dist bestimmt den Abstand zwischen Betrachter und Ursprung. *azimuth* ist der Winkel im x-y-Feld, *incidence* im y-z-Feld. *amount* gibt an, wie weit die Linie rotiert wird.

swf_lookat

Die Funktion bestimmt den Sichtwinkel zwischen zwei Betrachtungspunkten.

swf_lookat

```
void swf_lookat(double view_x, double view_y, double view_z,  
               double ref_x, double ref_y, double ref_z,  
               double twist)
```



swf_pushmatrix

Legt die aktuelle Transformationsmatrix auf dem Stapel ab.

swf_pushmatrix

```
void swf_pushmatrix(void);
```



swf_popmatrix

Holt die aktuelle Transformationsmatrix vom Stapel.

swf_popmatrix

```
void swf_popmatrix(void);
```



swf_scale

Skaliert die aktuelle Transformation.

swf_scale

```
void swf_scale(double x, double y, double z)
```



swf_translate

Die Funktion übersetzt die aktuelle Transformation auf die angegebene Ebene Koordinaten.

swf_translate

```
void swf_translate(double x, double y, double z)
```



swf_rotate**swf_rotate**

Die Funktion rotiert die aktuelle Transformation um den angegebenen Winkel *angle* und die Achse *axis*.



```
void swf_rotate(double angle, string axis)
```

Für *axis* können Sie "x", "y" oder "z" angeben.

swf_posround**swf_posround**

Die Funktion schaltet eine interne Rundungsfunktion ein oder aus. Wenn die Rundungsfunktion aktiviert ist, werden Texte unter Umständen exakter dargestellt, da berechnete Werte auf volle Pixel gesetzt werden.



```
void swf_posround(int round)
```

Mail-Funktionen

Die mit dem Präfix *imap* beginnenden Funktionen lassen sich auch für die Verarbeitung in normalen POP3- und SMTP-Funktionen verwenden. Wenn Sie E-Mail-Funktionen benötigen, sollten Sie das IMAP-Modul immer mit übersetzen oder die entsprechende DLL aktivieren.

mail**mail**

Die Funktion `mail` dient zum Versenden von E-Mail über einen SMTP-Server.



```
bool mail(string to, string subject, string message,  
          string [additional_headers]);
```

Die Nachricht *message* mit der Betreffzeile *subject* wird an den Empfänger *to* gesendet. Weitere Kopfzeilen der E-Mail können mit dem optionalen Parameter *additional_headers* angegeben werden. Die E-Mail-Nachricht muss zuvor mit geeigneten Funktionen bearbeitet werden, um den Anforderungen der Versendung von E-Mail zu genügen.



Beispiel

```
$message = "Line 1\nLine 2\nLine 3";  
mail("rasmus@lerdorf.on.ca", "My Subject", $message);
```



Beispiel

```
$message = "Line 1\nLine 2\nLine 3";  
$header = "From: webmaster@$SERVER_NAME\nReply-To:  
webmaster@$SERVER_NAME\nX-Mailer: PHP/" . phpversion();  
mail("nobody@aol.com", "the subject", $message, $header);
```

`imap_append`

`imap_append` fügt eine Nachricht an eine Mailbox an.

```
int imap_append(int imap_stream, string mbox,  
                string message, string flags);
```

Die Funktion gibt im Erfolgsfall `TRUE` zurück, sonst `FALSE`. Die Besonderheiten des IMAP-Servers sind zu beachten.

`imap_append`



`imap_base64`

`imap_base64` dekodiert Base64-kodierten Text.

```
string imap_base64(string text);
```

Die Funktion gibt den dekodierten Text als Zeichenkette zurück.

`imap_base64`



`imap_body`

`imap_body` liest den Nachrichtentext einer Nachricht.

```
string imap_body(int imap_stream, int msg_number, int flags);
```

Die Nachricht wird anhand der Nummer `msg_number` ausgewählt. Der Parameter `flags` wird nach folgender Liste bestimmt:

- `FT_UID`. Die Nachrichtennummer ist eine UID.
- `FT_PEEK`. Setzt nicht das Flag `\Seen` (Gelesen), wenn es nicht ohnehin schon gesetzt wurde.
- `FT_INTERNAL`. Die zurückgegebene Zeichenkette bleibt im internen Format.

`imap_body`



`imap_check`

`imap_check` prüft die aktuelle Mailbox.

```
object imap_check(int imap_stream);
```

Die Funktion gibt ein Objekt zurück, das bestimmte Informationen über eine Mailbox enthält. Diese Informationen werden als Eigenschaften des Objekts abgebildet:

- `Date` : Datum der Nachricht
- `Driver` : Treiber
- `Mailbox` : Name der Mailbox
- `Nmsgs` : Anzahl der Nachrichten in der Mailbox

`imap_check`



- Recent : Anzahl der neuen Nachrichten in der Mailbox

imap_close

imap_close imap_close schließt eine IMAP-Verbindung.

◆ `int imap_close(int imap_stream, [int flags]);`

Das optionale Flag *flag* kann auf den Wert CL_EXPUNGE gesetzt werden. Dadurch wird die Mailbox vor dem Schließen geleert.

imap_createmailbox

imap_createmailbox imap_createmailbox erzeugt eine neue Mailbox mit dem Namen *mbx*. Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

◆ `int imap_createmailbox(int imap_stream, string mbx);`

imap_delete

imap_delete imap_delete markiert eine Nachricht zum Löschen aus der Mailbox.

◆ `int imap_delete(int imap_stream, int msg_number);`

Die Funktion gibt immer TRUE zurück. Die eigentliche Löschung der Nachrichten erfolgt mit der Funktion imap_expunge.

imap_deletemailbox

imap_deletemailbox imap_deletemailbox löscht eine Mailbox mit allen darin enthaltenen Nachrichten.

◆ `int imap_deletemailbox(int imap_stream, string mbx);`

Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_expunge

imap_expunge imap_expunge löscht alle Nachrichten, die zum Löschen gekennzeichnet wurden.

◆ `int imap_expunge(int imap_stream);`

Die Funktion gibt immer TRUE zurück.

imap_fetchbody

imap_fetchbody imap_fetchbody liest einen bestimmten Teil einer Nachricht.

```
string imap_fetchbody(int imap_stream, int msg_number,
                      string part_number, flags flags);
```



Der Teil wird durch eine Zeichenkette ausgewählt, die aus Doppelpunkten bestehenden Zahlen besteht, entsprechend der IMAP-Spezifikation. Eine Dekodierung der zurückgegebenen Zeichenkette erfolgt nicht.

Der Parameter *flags* wird nach folgender Liste bestimmt:

- `FT_UID`. Die Nachrichtennummer ist eine UID.
- `FT_PEEK`. Setzt nicht das Flag `\Seen` (Gelesen) , wenn es nicht ohnehin schon gesetzt wurde.
- `FT_INTERNAL`. Die zurückgegebene Zeichenkette bleibt im internen Format.

imap_fetchstructure

`imap_fetchstructure` liest die Struktur der Nachricht mit der Nummer *msg_number*.

imap_fetchstructure

```
array imap_fetchstructure(int imap_stream, int msg_number);
```



Der Rückgabewert ist ein Objekt mit folgenden Eigenschaften:

- `type`, `encoding`, `ifsubtype`, `subtype`, `ifSyntax`, `Syntax`, `ifid`, `id`, `lines`, `bytes`, `ifparameters`

Es wird außerdem ein Array von Objekten mit dem Namen `parameters[]` zurückgegeben. Diese Objekte haben folgende Eigenschaften:

- `attribute`, `value`

Wenn eine Multipart-Nachricht vorliegt, wird zusätzlich ein Array von Objekten mit dem Namen `parts[]` erzeugt, das wiederum alle genannten Eigenschaften besitzt.

imap_header

`imap_header` liest den Kopf (Header) einer Nachricht.

imap_header

```
object imap_header(int imap_stream, int msg_number,
                   int fromlength, int subjectlength,
                   int defaulthost);
```



Die Funktion gibt ein Objekt zurück, das die verschiedenen Header-Zeilen als Eigenschaften enthält:

- `reply`
- `date`. Datum
- `subject`. Subject, Betreff
- `in_reply_to`
- `message_id`
- `newsgroups`
- `followup_to`
- `references`
- `toaddress` (voller Umfang mit bis zu 1 024 Zeichen)
- `to[]`. Ein Array mit Objekten der »To«-Zeile:
 - `personal`
 - `adl`
 - `mailbox`
 - `host`
 - `fromaddress` (voller Umfang mit bis zu 1 024 Zeichen)
- `from[]`. Ein Array mit Objekten der »To«-Zeile:
 - `personal`
 - `adl`
 - `mailbox`
 - `host`
 - `ccaddress` (voller Umfang mit bis zu 1 024 Zeichen)
- `cc[]`. Ein Array mit Objekten der »Cc«-Zeile:
 - `personal`
 - `adl`
 - `mailbox`
 - `host`
 - `bccaddress` (voller Umfang mit bis zu 1 024 Zeichen)
- `bcc[]`. Ein Array mit Objekten der »Bcc«-Zeile:
 - `personal`

- adl
- mailbox
- host
- reply_toaddress (voller Umfang mit bis zu 1 024 Zeichen)
- reply_to[]. Ein Array mit Objekten der »Reply_To«-Zeile:
 - personal
 - adl
 - mailbox
 - host
 - senderaddress (voller Umfang mit bis zu 1 024 Zeichen)
- sender[]. Ein Array mit Objekten der Sender-Zeile:
 - personal
 - adl
 - mailbox
 - host
 - return_path (voller Umfang mit bis zu 1 024 Zeichen)
- return_path[]. Ein Array mit Objekten der »«Return_Path«-Zeile:
 - personal
 - adl
 - mailbox
 - host
 - udate (Unix-Zeitstempel)
- fetchfrom (formatierte Absenderzeile)
- fetchsubject (formatiertes Betrefffeld)

imap_headers

imap_headers gibt die Kopfzeilen aller Nachrichten der Mailbox zurück.

imap_headers

```
array imap_headers(int imap_stream);
```



Die Rückgabe erfolgt in einem Array, jede Nachricht stellt ein Element dar.

imap_listmailbox

imap_listmailbox imap_listmailbox liest alle Mailboxen aus.

◆ array **imap_listmailbox**(int imap_stream, string ref, string pat);

imap_listsubscribed

imap_listsubscribed imap_listsubscribed listet alle oder ausgewählte abonnierte Mailboxen auf.

◆ array **imap_listsubscribed**(int imap_stream, string ref, string pattern);

ref und *pattern* sind Suchmuster, nach denen die Suche eingeschränkt werden kann.

imap_mail_copy

imap_mail_copy imap_mail_copy kopiert Nachrichten in die Mailbox.

◆ int **imap_mail_copy**(int imap_stream, string msglist, string mbox, int flags);

Gibt im Erfolgsfall TRUE zurück, sonst FALSE. Der Parameter *flag* ist ein Bitfeld mit folgenden Werten:

- CP_UID. Die Nachrichtennummern sind UIDs.
- CP_MOVE. Die Nachrichten sollen an der Quelle gelöscht werden (entspricht Verschieben).

imap_mail_move

imap_mail_move imap_mail_move verschiebt Nachrichten in eine Mailbox.

◆ int **imap_mail_move**(int imap_stream, string msglist, string mbox);

Gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_num_msg

imap_num_msg imap_num_msg ermittelt die Anzahl der Nachrichten in einer Mailbox.

◆ int **imap_num_msg**(int stream_id);

`imap_num_recent`

`imap_num_recent` ermittelt die Anzahl der neuen Nachrichten in einer Mailbox. `imap_num_recent`

```
int imap_num_recent(int imap_stream);
```



`imap_open`

`imap_open` öffnet eine IMAP-Verbindung zu einer Mailbox. `imap_open`

```
int imap_open(string mailbox,  
               string username, string password,  
               int flags);
```



Gibt die Nummer einer IMAP-Verbindung oder `FALSE` zurück. Mit dieser Funktion können auch Verbindungen zu POP3- oder NNTP-Servern eröffnet werden.

Der Parameter *options* ist ein Bitfeld und kann folgende Werte annehmen:

- `OP_READONLY`. Öffnet eine Mailbox nur zum Lesen.
- `OP_ANONYMOUS`. Verwendet keine `.newsrc` für Nachrichten.
- `OP_HALFOPEN`. Für IMAP- und NNTP-Verbindungen wird die Verbindung, aber keine Mailbox geöffnet.
- `CL_EXPUNGE`. Als gelöscht markierte Nachrichten werden beim späteren Schließen der Box gelöscht.

Das folgende Beispiel zeigt die Verbindung mit einem IMAP-Server:

```
$mbox = imap_open("{localhost:143}INBOX","user_id","password");
```

Für einen lokalen POP3-Server sieht das Beispiel folgendermaßen aus:

```
$mbox = imap_open("{localhost:110}INBOX","user_id","password");
```

Ein NNTP-Server benutzt einen anderen Port:

```
$nntp = imap_open("{localhost:119}comp.test","","");
```

Bei entfernten Servern wird das Wort *localhost* durch den Domain-Namen oder die IP-Adresse des Servers ersetzt.



Beispiel

imap_ping

imap_ping imap_ping prüft, ob die Verbindung noch besteht, und gibt dann TRUE zurück.

◆ `int imap_ping(int imap_stream);`

imap_renamemailbox

imap_renamemailbox imap_renamemailbox benennt eine Mailbox in einen anderen Namen um. Der alte und der neue Name muss angegeben werden.

◆ `int imap_renamemailbox(int imap_stream,
string old_mbox,
string new_mbox);`

Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_reopen

imap_reopen imap_reopen eröffnet eine Verbindung erneut zu einer neuen Mailbox.

◆ `int imap_reopen(string imap_stream, string mailbox,
string [flags]);`

Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE. Das optionale Bitfeld *mask* kann folgende Werte beinhalten:

- OP_READONLY. Öffnet eine Mailbox nur zum Lesen.
- OP_ANONYMOUS. Verwendet keine .newsrc für Nachrichten.
- OP_HALFOPEN. Für IMAP- und NNTP-Verbindungen wird die Verbindung, aber keine Mailbox geöffnet.
- CL_EXPUNGE. Als gelöscht markierte Nachrichten werden beim späteren Schließen der Box gelöscht.

imap_subscribe

imap_subscribe imap_subscribe abonniert eine Mailbox.

◆ `int imap_subscribe(int imap_stream, string mbox);`

Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_undelete

imap_undelete imap_undelete nimmt ein Löschkennzeichen wieder weg.

```
int imap_undelete(int imap_stream,int msg_number);
```



Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_unsubscribe

Die Funktion `imap_unsubscribe` hebt das Abonnement der Mailbox *mbx* wieder auf.

imap_unsubscribe

```
int imap_unsubscribe(int imap_stream, string mbx);
```



Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

imap_qprint

`imap_qprint` konvertiert eine im Quoted-Printable-Format kodierte Zeichenkette in das 8-Bit-Format.

imap_qprint

```
string imap_qprint(string string);
```



imap_8bit

`imap_8bit` konvertiert eine im 8-Bit-Format kodierte Zeichenkette in das Quoted-Printable-Format.

imap_8bit

```
string imap_8bit(string string);
```



imap_binary

`imap_binary` konvertiert eine im 8-Bit-Format kodierte Zeichenkette in das Base64-Format.

imap_binary

```
string imap_binary(string string);
```



imap_scanmailbox

`imap_scanmailbox` liest eine Liste der Mailboxen ein und kann nach Mailboxnamen suchen.

imap_scanmailbox

```
array imap_scanmailbox(int imap_stream, string string);
```



Die Funktion gibt ein Array *array* zurück, das die Namen der gefundenen Mailboxen enthält.

imap_mailboxmsginfo

`imap_mailboxmsginfo` holt allgemeine Informationen über die Mailbox *imap_stream*.

imap_mailboxmsginfo

```
array imap_mailboxmsginfo(int imap_stream);
```



Die Funktion gibt ein Array zurück, das bestimmte Informationen über eine Mailbox enthält. Diese Informationen werden als Elemente des Arrays abgebildet:

- Date : Datum der Nachricht
- Driver : Treiber
- Mailbox : Name der Mailbox
- Nmsgs : Anzahl der Nachrichten in der Mailbox
- Recent : Anzahl der neuen Nachrichten in der Mailbox
- Unread : Anzahl der nicht gelesenen Nachrichten
- Size : Größe der Mailbox

imap_rfc822_write_address

imap_rfc822_write_address

imap_rfc822_write_address gibt eine nach RFC822 formatierte Adresse zurück.



```
string imap_rfc822_write_address(string mailbox,
                                string host,
                                string personal);
```

imap_rfc822_parse_adrlist

imap_rfc822_parse_adrlist

imap_rfc822_parse_adrlist untersucht eine Adresse, in dem diese nach dem in RFC 822 definierten Verfahren analysiert wird.



```
string imap_rfc822_parse_adrlist(string address,
                                string default_host);
```

Für jede Adresse in einer Zeichenkette wird ein Array mit vier Objekten zurückgegeben. Die Objekte haben folgende Eigenschaften:

- mailbox. Der Name der Mailbox
- host. Der Hostname
- personal. Der persönlichen Name
- adl. Die Route zur Quell-Domain

imap_setflag_full

imap_setflag_full

imap_setflag_full setzt Flags an Nachrichten.

```
string imap_setflag_full(int stream, string sequence,  
                        string flag, string options);
```



imap_clearflag_full

`imap_clearflag_full` löscht Flags an Nachrichten. Siehe auch bei der Funktion `imap_setflag_full`.

imap_clearflag_full

```
string imap_clearflag_full(int stream, string sequence,  
                        string flag, string options);
```



imap_sort

`imap_sort` gibt ein Array mit Nachrichtennummern zurück, die nach bestimmten Kriterien sortiert sind.

imap_sort

```
string imap_sort(int stream, int criteria,  
                int reverse, int options);
```



Der Parameter *reverse* wird auf 1 gesetzt, wenn rückwärts sortiert werden soll. *criteria* kann eines oder mehrere der folgenden Bitwerte sein:

- SORTDATE Nachrichtendatum
- SORTARRIVAL Empfangszeit
- SORTFROM Nach der Absenderadresse
- SORTSUBJECT Nach der Betreffzeile
- SORTTO Nach der ersten To-Adresse
- SORTCC Nach der ersten Cc-Adresse
- SORTSIZE Anzahl der Nachrichten in Oktets

Die *flags* sind Bitfelder mit folgenden Werten:

- SE_UID gibt die UID anstatt der fortlaufenden Nummern zurück.
- SE_NOPREFETCH. Gesuchte Nachrichten werden nicht gescannt.

imap_fetchheader

`imap_fetchheader` gibt den Header einer Nachricht zurück.

imap_fetchheader

```
string imap_fetchheader(int imap_stream, int msgno, int flags);
```



Die Flags *flag* sind:

- FT_UID. der Parameter *msgno* ist eine UID.
- FT_INTERNAL. Der Rückgabewert wird nicht konvertiert.
- FT_PREFETCHTEXT. Der Text wird sofort gelesen.

imap_uid

imap_uid imap_uid gibt die UID für eine Nachricht mit der Nummer *msgno* zurück.



```
string imap_uid(string mailbox, int msgno);
```

LDAP-Funktionen

ldap_add

ldap_add ldap_add fügt einem LDAP-Verzeichnis Einträge hinzu.



```
int ldap_add(int link_identifizier, string dn, array entry);
```

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Die DN (distinguished name) wird durch *dn* angegeben. Die Attribute des Objekts werden im Array *entry* folgendermaßen angegeben:

```
entry["attribute1"] = value
entry["attribute2"][0] = value1
entry["attribute2"][1] = value2
```



```
<?php
// Annahme: LDAP Server ist lokal installiert
$ds=ldap_connect("localhost");
if ($ds) {

    // Zugriff auf das aktuelle Verzeichnis
    $r = ldap_bind($ds,"cn=root, o=MyCompany, c=US", "secret");

    // Daten vorbereiten
    $info["cn"]="John Jones";
    $info["sn"]="Jones";
    $info["mail"]="jonj@here.and.now";
    $info["objectclass"]="person";

    // Daten dem Verzeichnis hinzufügen
    $r = ldap_add($ds,"cn=John Jones, o=MyCompany, c=US", $info);
    ldap_close($ds);
} else {
    echo "Kann LDAP-Server nicht erreichen";
}
?>
```

ldap_mod_add

ldap_mod_add fügt aktuellen Attributen weitere Werte hinzu.

ldap_mod_add

```
int ldap_mod_add(int link_identifizier,  
                string dn, array entry);
```



Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Die Attribute werden der angegebenen *dn* hinzugefügt.

Siehe auch ldap_add.

ldap_mod_del

ldap_mod_del löscht Werte aktueller Attribute.

ldap_mod_del

```
int ldap_mod_del(int link_identifizier,  
                string dn, array entry);
```



Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.

Siehe auch ldap_del.

ldap_mod_replace

ldap_mod_replace ersetzt Werte von Attributen mit neuen Werten.

ldap_mod_replace

```
int ldap_mod_replace(int link_identifizier,  
                    string dn, array entry);
```



Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.

Siehe auch ldap_modify.

ldap_bind

ldap_bind stellt die Bindung zu einem LDAP-Verzeichnis her.

ldap_bind

```
int ldap_bind(int link_identifizier,  
             string [bind_rdn],  
             string [bind_password]);
```



Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Verwendet zum Anmelden die RDN *bind_rdn* und das Kennwort *bind_password*. Die Angaben sind optional; fehlen sie, wird anonym angemeldet.

ldap_close

ldap_close schließt die Verbindung *link_identifizier* zu einem LDAP-Server.

ldap_close

- ◆ `int ldap_close(int link_identifizier);`
Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.

ldap_connect

ldap_connect ldap_connect verbindet mit einem LDAP-Server.

- ◆ `int ldap_connect(string [hostname], int[port]);`
Diese Funktion gibt das Handle zurück, das andere Funktionen zum Zugriff auf den LDAP-Server benötigt. Im Fehlerfall wird FALSE zurückgegeben. Der Port ist optional, fehlt die Angabe, wird der Standardport 389 verwendet.

ldap_count_entries

ldap_count_entries ldap_count_entries ermittelt die Anzahl der Einträge einer vorangegangenen Suche, deren Ergebnisliste durch *result_identifizier* bezeichnet wird.

- ◆ `int ldap_count_entries(int link_identifizier,
int result_identifizier);`
Im Fehlerfall wird FALSE zurückgegeben.

ldap_delete

ldap_delete ldap_delete löscht ein durch *dn* bezeichnetes Objekt aus dem Verzeichnis.

- ◆ `int ldap_delete(int link_identifizier, string dn);`
Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE.

ldap_dn2ufn

ldap_dn2ufn ldap_dn2ufn konvertiert ein DN *dn* in ein nutzerfreundlicheres Format, indem die Typbezeichner entfernt werden.

- ◆ `string ldap_dn2ufn(string dn);`

ldap_explode_dn

ldap_explode_dn ldap_explode_dn splittet ein durch *dn* bezeichnetes DN-Objekt in seine Bestandteile auf und gibt diese als assoziatives Array zurück.

- ◆ `array ldap_explode_dn(string dn, int with_attrib);`
Um auch alle Attribute in der Form *attribut=wert* zu erhalten, muss der Parameter *with_attrib* auf 0 gesetzt werden, sonst auf 1.

ldap_first_attribute

ldap_first_attribute gibt das erste durch *result_entry_identifier* bestimmte Attribut zurück.

```
string ldap_first_attribute(int link_identifier,  
                           int result_entry_identifier,  
                           int ber_identifier);
```

Im Erfolgsfall wird das erste Attribut zurückgegeben, sonst FALSE. Der interne Zeiger auf die Attribute wird durch *ber_identifier* bestimmt.

Siehe auch ldap_next_attribute und ldap_get_attributes.

ldap_first_entry

ldap_first_entry gibt die erste ID der Ergebnisliste der Anfrage zurück.

```
int ldap_first_entry(int link_identifier,  
                    int result_identifier);
```

Im Fehlerfall wird FALSE zurückgegeben. Die Einträge können sequenziell mit ldap_first_entry und ldap_next_entry gelesen werden.

Siehe auch ldap_get_entries.

ldap_free_result

ldap_free_result gibt den von der Ergebnisliste belegten Speicher wieder frei.

```
int ldap_free_result(int result_identifier);
```

Im Erfolgsfall wird TRUE zurückgegeben, sonst FALSE. Normalerweise wird der Speicher am Ende des Skripts automatisch freigegeben. Der Einsatz der Funktion macht deshalb nur innerhalb eines komplexen Skripts Sinn.

ldap_get_attributes

ldap_get_attributes holt die Attribute einer Suchanfrage.

```
array ldap_get_attributes(int link_identifier,  
                          int result_entry_identifier);
```

Die Attribute werden vollständig in ein mehrdimensionales Array zurückgegeben. Der Zugriff auf das Array kann über folgende Notationen erfolgen:

ldap_
first_attribute



ldap_first_entry



ldap_free_result



ldap_
get_attributes



- `return_value["count"]` = Anzahl der Werte des Attributes.
- `return_value[0]` = Erster Wert.
- `return_value[n]` = n-ter Wert des Attributes.
- `return_value["attribute"]["count"]` = Anzahl der Werte des Attributes.
- `return_value["attribute"][0]` = Erster Wert des Attributes.
- `return_value["attribute"][n]` = n-ter Wert des Attributes.



Beispiel

```
// $ds ist der Link-Identifikator des Verzeichnisses
// $sr ist eine gültige Ergebnisliste der Suchanfrage
$entry = ldap_first_entry($ds, $sr);
$attrs = ldap_get_attributes($ds, $entry);
echo $attrs["count"]." Attribute dieses Eintrags:<p>";
for ($i=0; $i<$attrs["count"]; $i++) {
    echo $attrs[$i]."<br>";
}
```

Siehe auch `ldap_first_attribute` und `ldap_next_attribute`.

ldap_get_dn

ldap_get_dn

`ldap_get_dn` ermittelt den DN (distinguished name) des Eintrags.



```
string ldap_get_dn(int link_identifier,
                  int result_entry_identifier);
```

ldap_get_entries

ldap_get_entries

`ldap_get_entries` gibt alle Objekte der Ergebnisliste zurück.



```
array ldap_get_entries(int link_identifier,
                      int result_identifier);
```

Im Erfolgsfall wird ein mehrdimensionales Array zurückgegeben, sonst `FALSE`. Der Index des Arrays, der aus den Attributnamen besteht, wird in Kleinbuchstaben konvertiert. Die folgende Liste zeigt, wie Sie auf dieses Array zugreifen können:

- `return_value["count"]` = Anzahl der Einträge
- `return_value[0]` = Array mit Detailenträgen (erstes Objekt):
 - `return_value[n]["dn"]` = DN des n-ten Eintrags
 - `return_value[n]["count"]` = Anzahl der Attribute des Eintrags

- `return_value[n][k]` = k-tes Attribut des n-ten Eintrags
- `return_value[n]["attribute"]["count"]` = Anzahl der Werte des n-ten Attributes
- `return_value[n]["attribute"][k]` = k-ter Wert des n-ten Attributes

Siehe auch `ldap_first_entry` und `ldap_next_entry`.

ldap_get_values

`ldap_get_values` ruft alle Werte aus einer Ergebnisliste ab.

ldap_get_values

```
array ldap_get_values(int link_identifizier,
                    int result_entry_identifizier,
                    string attribute);
```



Im Erfolgsfall wird ein Array mit den Werten des Attributes *attribute* zurückgegeben, sonst FALSE. Der erste Index ist 0. Die Angabe des Attributes kann hardkodiert (beispielsweise "email") oder mit Hilfe der Funktion `ldap_get_attributes`.

Der Zugriff auf das Array kann folgendermaßen erfolgen:

- `return_value["count"]` = Anzahl der Werte des Attributes
- `return_value[0]` = erster Wert
- `return_value[n]` = n-ter Wert des Attributes

`$ds` ist ein gültiges Handle zum LDAP-Server, `$sr` ist ein Handle auf ein Suchergebnis einer vorangegangenen Suche, `$entry` der Eintrag der Suche. Das Beispiel sucht alle Attribute der gefundenen E-Mail-Adressen:



Beispiel

```
$values = ldap_get_values($ds, $entry, "mail");
echo $values["count"]
    . " E-Mail-Adresse für diesen Eintrag.<p>";
for ($i=0; $i < $values["count"]; $i++)
    echo $values[$i]."<br>";
```

ldap_list

`ldap_list` führt eine einfache Suche durch.

ldap_list

```
int ldap_list(int link_identifizier,
             string base_dn,
             string filter, array [attributes]);
```



Gibt im Erfolgsfall TRUE zurück, sonst FALSE. Die Suche wird im aktuellen Kontext ausgeführt, also dem momentan gültigen Verzeichnis (LDAP_SCOPE_ONELEVEL).



```
// $ds ist eine Link-ID zum Server
$basedn = "o=My Company, c=US";
$justthese = array("ou");
$sr=ldap_list($ds, $basedn, "ou=", $justthese);
$info = ldap_get_entries($ds, $sr);
for ($i=0; $i<$info["count"]; $i++) {
    echo $info[$i]["ou"][0] ;
}
```

ldap_modify

ldap_modify

ldap_modify verändert einen LDAP-Eintrag.

◆ `int ldap_modify(int link_identifizier,
string dn, array entry);`

Gibt im Erfolgsfall TRUE zurück, sonst FALSE. Für weitere Informationen zur Syntax siehe ldap_add.

ldap_next_attribute

ldap_next_attribute

ldap_next_attribute holt das nächste Attribut aus der Ergebnisliste. Dabei wird der interne Zeiger auf die Attribute weiter gesetzt.

◆ `string ldap_next_attribute(int link_identifizier,
int result_entry_identifizier,
int ber_identifizier);`

Gibt im Erfolgsfall das Attribut zurück, sonst FALSE.

Siehe auch ldap_get_attributes.

ldap_next_entry

ldap_next_entry

ldap_next_entry holt den nächsten Eintrag.

◆ `int ldap_next_entry(int link_identifizier,
int result_entry_identifizier);`

Die Funktion gibt ein Handle auf den Eintrag zurück oder FALSE, wenn keine weiteren Einträge zu finden sind.

Siehe auch ldap_get_entries.

ldap_read

ldap_read

ldap_read sucht einen Eintrag.

```
int ldap_read(int link_identifizier, string base_dn,
             string filter, array [attributes]);
```



Gibt im Erfolgsfall ein Handle auf die Ergebnisliste zurück, sonst FALSE. Die Funktion sucht im Bereich LDAP_SCOPE_BASE. Der Filter *filter* darf nicht leer sein. Wenn Sie alle Objekte suchen möchten, verwenden Sie Platzhalter:

```
$filter = "objectClass=*";
$result = $ldap_read($linkid, $basedn, $filter);
```

Als vierten, optionalen Parameter kann ein Array mit Attributen angegeben werden, die vorhanden sein müssen. Das Array erzeugen Sie mit der Funktion array:

```
$attrib = array("email","name","organisation")
```

ldap_search

ldap_search sucht im LDAP-Baum.

ldap_search

```
int ldap_search(int link_identifizier, string base_dn,
              string filter, array [attributes]);
```



Gibt im Erfolgsfall ein Handle auf die Ergebnisliste zurück, sonst FALSE. Die Funktion sucht im Bereich LDAP_SCOPE_SUBTREE. Das Filter *filter* darf nicht leer sein. Wenn Sie alle Objekte suchen möchten, verwenden Sie Platzhalter.

Mehr Informationen zu Filtern finden Sie unter:

- <http://developer.netscape.com/tech/directory>

```
// $ds ist die LinkID zum Server
// $person ist eine Person,
//z.B. "Jo" $dn = "o=My Company, c=US";
$filter="(|(sn=$person*)(givenname=$person*))";
$justthese = array( "ou", "sn", "givenname", "mail");
$sr = ldap_search($ds, $dn, $filter, $justthese);
$info = ldap_get_entries($ds, $sr);
print $info["count"]." entries returned<p>;
```



Beispiel

ldap_unbind

ldap_unbind hebt die Bindung an ein LDAP-Verzeichnis auf.

ldap_unbind

```
int ldap_unbind(int link_identifizier);
```



Gibt im Erfolgsfall TRUE zurück, sonst FALSE.

Bildfunktionen (GD)

Mit den Bildfunktionen werden dynamisch Bilder erzeugt. Die dazu nötige GD-Bibliothek von Thomas Boutell muss entsprechend eingebunden werden. Mehr Informationen und aktuelle Versionen gibt es auf der Website unter

- <http://www.boutell.com/gd>

getimagesize

getimagesize

getimagesize gibt die Größe eines GIF-, JPG- oder PNG-Bildes zurück.



```
array getimagesize(string filename, array [imageinfo]);
```

Die Größe wird in Pixel in einem Array mit vier Elementen zurückgegeben. Die Indizes haben folgende Bedeutung:

- 0: Breite in Pixel
- 1: Höhe in Pixel
- 2: Grafikformat; 1 = GIF, 2 = JPG, 3 = PNG
- 3: Zeichenkette für die Einbindung in das -Tag:

```
"height=xxx width=xxx"
```



```
<?php $size = GetImageSize("img/flag.jpg"); ?>
<IMG SRC="img/flag.jpg" <?php echo $size[3]; ?>
```

Der optionale Parameter *imageinfo* gibt weitere Bildinformationen zurück. Der APP13-Marker ist eine sehr nützliche Information über steganografische Elemente. Sie können die Funktion *iptcparse* nutzen, um damit umzugehen. Mehr Informationen finden Sie im Web unter:

- <http://www.xe.net/iptc>



```
<?php
$size = GetImageSize("testimg.jpg",&$info);
if (isset($info["APP13"])) {
    $iptc = iptcparse($info["APP13"]);
    var_dump($iptc);
}
?>
```



Diese Funktion benötigt die GD-Bibliothek nicht.

imagearc

imagearc zeichnet einen Teil einer Ellipse.

```
int imagearc(int im, int cx, int cy,  
            int w, int h, int s, int e, int col);
```

Die Funktion zeichnet einen Ellipsenbogen mit dem Zentrum cx , cy (die linke obere Ecke ist 0,0) im Bild mit dem Handle im . w und h bestimmen die Höhe und Breite der Ellipse, während s und e den Anfangs- und Endwinkel des Bogens in Grad bestimmen. col bestimmt die Zeichenfarbe.

imagearc**imagechar**

imagechar zeichnet ein Zeichen in horizontaler Richtung.

```
int imagechar(int im, int font, int x, int y,  
             string c, int col);
```

Das erste Zeichen der Zeichenkette c wird an der Position x , y gezeichnet, wobei dort die obere, linke Ecke des Zeichens steht. Das Zeichen bekommt die Farbe col . $font$ bestimmt die Nummer eines eingebauten Fonts.

Siehe auch `imeloadfont`.

imagechar**imagecharup**

imagecharup zeichnet ein Zeichen in vertikaler Richtung.

```
int imagecharup(int im, int font, int x, int y,  
               string c, int col);
```

Das erste Zeichen der Zeichenkette c wird an der Position x , y gezeichnet, wobei dort die obere linke Ecke des Zeichens steht. Das Zeichen bekommt die Farbe col . $font$ bestimmt die Nummer eines eingebauten Fonts.

Siehe auch `imeloadfont`.

imagecharup**imagecolorallocate**

imagecolorallocate weist dem Bild eine Farbe zu. Der Farbwert wird zurückgegeben.

```
int imagecolorallocate(int im, int red, int green, int blue);
```

Jedem Bild muss jede verwendete Farbe zuvor mit dieser Funktion zugewiesen werden. Die so gebildeten Farbwerte werden von

**imagecolor
allocate**

anderen Funktionen verwendet. Das Handle *im* verweist auf ein Bild.



```
$white = ImageColorAllocate($im, 255,255,255);
$black = ImageColorAllocate($im, 0,0,0);
```

imagecolortransparent

imagecolor transparent

`imagecolortransparent` definiert eine Farbe als transparent. Der Farbwert muss zuvor mit `imagecolorallocate` bestimmt werden.

◆ `int imagecolortransparent(int im, int [col]);`

Die Funktion gibt das Handle auf die Farbe zurück. Wird keine Farbe angegeben, wird die aktuelle verwendet. Der Einsatz macht nur bei Grafikformaten Sinn, die Transparenz darstellen können.

imagecopyresized

imagecopy resized

`imagecopyresized` kopiert und skaliert einen Bildausschnitt von einem Bild in ein anderes.

◆ `int imagecopyresized(int dst_im, intsrc_im,
int dstX, int dstY,
int srcX, int srcY,
int dstW, int dstH,
int srcW, int srcH);`

Die Parameter für die Skalierung haben folgende Bedeutung:

- *dst_im*. ID des Zieles
- *src_im*. ID der Quelle
- *dstY, dstX, dstH, dstW*. Zielbereich
- *srcY, srcX, srcH, srcW*. Quellbereich

Wenn sich die Größe des Quell- und Zielbereiches unterscheiden, wird der Bildausschnitt entsprechend gestreckt oder gestaucht. Die linke, obere Ecke hat die Koordinaten 0,0. Wenn Ziel und Quelle im gleichen Bild liegen und die Bereiche sich überlappen, ist das Ergebnis unbestimmt.

imagecreate

imagecreate

`imagecreate` erzeugt ein neues Bild und gibt ein Handle auf dieses Bild zurück.

◆ `int imagecreate(int x_size, int y_size);`

x_size und *y_size* bestimmen die Größe des Bildes.

imagecreatefromgif

imagecreatefromgif erzeugt ein neues Bild auf der Basis eines vorhandenen.

```
int imagecreatefromgif(string filename)
```

Das Bild *filename* kann als lokaler Pfad oder vollständige URL eingelesen werden. Im Fehlerfall wird eine leere Zeichenkette zurück- und eine Fehlermeldung ausgegeben. Das Beispiel zeigt den Umgang damit. (Vielen Dank an *vic@zmysys.com.*)

```
function LoadGif($imgname) {
    $im = @imagecreatefromgif($imgname);
    /* Versuch zu öffnen */
    if ($im == "") { /* Fehler? */
        $im = ImageCreate(150,30); /* Leeres Bild */
        $bgc = ImageColorAllocate($im,255,255,255);
        $tc = ImageColorAllocate($im,0,0,0);
        ImageFilledRectangle($im,0,0,150,30,$bgc);
        ImageString($im,1,5,5,"Fehler beim laden",$tc);
        /* Ausgabe der Fehlermeldung als Bild */
    }
    return $im;
}
```

imagecreate
fromgif



imagecreatefromjpeg

imagecreatefromjpeg erzeugt ein neues Bild auf der Basis eines vorhandenen.

```
int imagecreatefromjpeg(string filename)
```

imagecreate
fromjpeg



imagecreatefrompng

imagecreatefrompng erzeugt ein neues PNG-Bild auf der Basis eines vorhandenen.

```
int imagecreatefrompng(string filename)
```

imagecreate
frompng



imagedashedline

imagedashedline zeichnet eine gestrichelte Linie.

```
int imagedashedline(int im, int x1, int y1,
                    int x2, int y2, int col);
```

imagedashedline



Die Linie verläuft von *x1*, *y1* bis *x2*, *y2* mit der Farbe *col*.

Siehe auch `imagine`.

imagedestroy

imagedestroy `imagedestroy` zerstört ein im Speicher bearbeitetes Bild und gibt den Speicher frei.

◆ `int imagedestroy(int im);`

imagefill

imagefill `imagefill` füllt einen Bereich.

◆ `int imagefill(int im, int x, int y, int col);`

Die Füllung beginnt am Punkt x, y und mit der Farbe col .

imagefilledpolygon

imagefilledpolygon `imagefilledpolygon` zeichnet ein gefülltes Polygon (Vieleck) mit einer bestimmten Farbe.

◆ `int imagefilledpolygon(int im, array points, int num_points,
int col);`

Das Polygon wird als Array übergeben, das folgenden Aufbau hat:
 $points[0] = x_0, points[1] = y_0, points[2] = x_1, points[3] = y_1$ usw.
 num_points gibt die Anzahl der Punkte an.

imagefilledrectangle

imagefilledrectangle `imagefilledrectangle` zeichnet ein gefülltes Rechteck in einer bestimmten Farbe.

◆ `int imagefilledrectangle(int im, int x1, int y1,
int x2, int y2, int col);`

Das Rechteck wird durch die linke obere Ecke x_1, y_1 und die rechte untere Ecke x_2, y_2 bestimmt. $0,0$ ist die linke obere Ecke des Bildes. col bestimmt die Füllfarbe.

imagefilltoborder

imagefilltoborder `imagefilltoborder` füllt einen Bereich mit einer bestimmten Farbe. Der Füllvorgang beginnt an der bestimmten Stelle x, y .

◆ `int imagefilltoborder(int im, int x, int y, int border, int col);`

Der zu füllende Bereich wird durch eine Randfarbe bestimmt, die mit $border$ spezifiziert wird.

imagefontheight

imagefontheight ermittelt die Größe eines Fonts in Pixel.

```
int imagefontheight(int font);
```

Siehe auch imagefontwidth und imageloadfont.

imagefontheight



imagefontwidth

imagefontwidth ermittelt die Fontbreite in Pixel.

```
int imagefontwidth(int font);
```

Siehe auch imagefontheight und imageloadfont.

imagefontwidth



imagegif

imagegif gibt das fertige GIF-Bild an den Browser oder in eine Datei aus.

```
int imagegif(int im, string [filename]);
```

Das Dateiformat ist GIF87a, wenn keine Farbe transparent ist, sonst ist es GIF89a. Der Dateiname ist optional, ohne Angabe wird das Bild sofort an den Browser gesendet. Die HTTP-Header müssen zuvor selbst erzeugt werden. Wird ein Dateiname angegeben, wird das Bild gespeichert.

imagegif



imagejpeg

imagejpeg gibt das fertige JPEG-Bild an den Browser oder in eine Datei aus.

```
int imagejpeg(int im, string [filename]);
```

Siehe auch Hinweise bei imagegif.

imagejpeg



imagepng

imagepng gibt das fertige PNG-Bild an den Browser oder in eine Datei aus.

```
int imagepng(int im, string [filename]);
```

Siehe auch Hinweise bei imagegif.

imagepng



imageinterlace

imageinterlace imageinterlace schaltet die Bildart interlaced oder nicht interlaced ein und aus.

◆ `int imageinterlace(int im, int [interlace]);`

Die Funktion gibt den aktuellen Zustand zurück, wenn der zweite Parameter nicht angegeben wurde. Der Parameter *interlace* ist 1 für interlaced oder 0 für normale Bilder.

imageline

imageline imageline zeichnet eine Linie.

◆ `int imageline(int im, int x1, int y1, int x2, int y2, int col);`

Die Linie verläuft von *x1, y1* nach *x2, y2* in der Farbe *col*.

Siehe auch `imagecreate` und `imagecolorallocate`.

imageloadfont

imageloadfont imageloadfont lädt einen neuen Font.

◆ `int imageloadfont(string file);`

Die Funktion lädt einen nutzerdefinierten Bitmap-Font und gibt eine ID-Nummer zurück. Die Nummer ist größer als 5, damit kein Konflikt mit den eingebauten Fonts auftritt. Die Struktur der Fonts selbst hängt von der Maschine ab, auf der PHP läuft. Die folgende Tabelle zeigt den grundsätzlichen Aufbau der Bitmap-Fonts.

Tabelle B.4: Font-Syntax

Byte	Datenformat	Bedeutung
0 - 3	Zahl	Anzahl der Zeichen im Font
4 - 7	Zahl	Wert des ersten Zeichens (meist 32 für Leerzeichen)
8 - 11	Zahl	Breite jedes Zeichens in Pixel
12 - 15	Zahl	Höhe jedes Zeichens in Pixel
ab 16	Zeichen	Array der Zeichendaten; jedes folgende Byte beschreibt ein Pixel eines Zeichens, von links oben nach rechts unten. Die Gesamtzahl berechnet sich wie folgt: Zeichenzahl * Breite * Höhe

Siehe auch `ImageFontWidth` und `ImageFontHeight`.

imagepolygon

imagepolygon zeichnet ein Polygon.

```
int imagepolygon(int im, array points, int num_points, int col);
```

Das Polygon wird durch ein Array beschrieben, das folgenden Aufbau hat: points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1 usw.. *num_points* gibt die Gesamtzahl der Punkte an.

Siehe auch imagecreate.

imagepolygon



imagerectangle

imagerectangle zeichnet ein Rechteck.

```
int imagerectangle(int im, int x1, int y1,  
                  int x2, int y2, int col);
```

Das Rechteck startet von links oben *x1, y1* und endet rechts unten bei *x2, y2*. 0,0 ist die linke obere Ecke des Bildes.

imagerectangle



imagesetpixel

imagesetpixel setzt einen einzelnen Bildpunkt.

```
int imagesetpixel(int im, int x, int y, int col);
```

Der Punkt erscheint an den Koordinaten *x, y* und in der Farbe *col*.

Siehe auch imagecreate und imagecolorallocate.

imagesetpixel



imagestring

imagestring zeichnet eine Zeichenkette horizontal.

```
int imagestring(int im, int font, int x, int y,  
               string s, int col);
```

Die Zeichenkette *s* wird im Bild *im* beginnend an der linken oberen Ecke *x, y* gezeichnet. Benutzt wird die Farbe *col* und der Font *font*.

Siehe auch imageloadfont.

imagestring



imagestringup

imagestringup zeichnet eine Zeichenkette horizontal.

```
int imagestringup(int im, int font, int x, int y,  
                  string s, int col);
```

imagestringup



Die Zeichenkette *s* wird im Bild *im* beginnend an der linken oberen Ecke *x, y* gezeichnet. Benutzt wird die Farbe *col* und der Font *font*.

Siehe auch `imageloadfont`.

imagesx

imagesx

`imagesx` ermittelt die Breite eines Bildes in Pixel.



```
int imagesx(int im);
```

Siehe auch `imagecreate` und `imagesy`.

imagesy

imagesy

`imagesy` ermittelt die Höhe eines Bildes in Pixel.



```
int imagesy(int im);
```

Siehe auch `imagecreate` und `imagesx`.

imagettfbbox

imagettfbbox

`imagettfbbox` gibt ein Array mit den Abmessungen eines TRUETYPE-Textes zurück.



```
array imagettfbbox(int size, int angle, string fontfile,  
                  string text);
```

Die Parameter haben folgende Bedeutung:

- *text*. Die zu bemessende Zeichenkette.
- *size*. Die Größe des Fonts.
- *fontfile*. Die TTF-Datei (TrueType-Font). Dies kann ein Dateiname oder eine URL sein.
- *angle*. Der Winkel, in dem der Text gezeichnet wird.

Die Funktion `ImageTTFBBox` gibt ein Array mit acht Elementen zurück, die folgende Daten enthalten:

- 0: untere linke Ecke, X Position
- 1: untere linke Ecke, Y Position
- 2: untere rechte Ecke, X Position
- 3: untere rechte Ecke, Y Position
- 4: obere rechte Ecke, X Position

- 5: obere rechte Ecke, Y Position
- 6: obere linke Ecke, X Position
- 7: obere linke Ecke, Y Position

Die Koordinaten sind absolut, also unabhängig vom Winkel, um den der Text gedreht erscheint.

Diese Funktion benötigt neben der GD- auch die Freetype-Bibliothek. Mehr Informationen zu Freetype finden Sie unter:

- <http://www.freetype.org>

Siehe auch ImageTTFText.

imagettftext

imagettftext zeichnet einen Text mit einem TrueType-Font.

```
array imagettftext(int im, int size, int angle,
                  int x, int y, int col,
                  string fontfile, string text);
```

Die Funktion zeichnet einen Text *text* an die Stelle *x*, *y* und in der Farbe *col*. Der Text kann mit dem Winkel *angle* gedreht werden und wird im Font *fontfile* dargestellt. Die Koordinaten stellen die linke untere Ecke (die Grundlinie) dar, im Gegensatz zur Funktion ImageString. Der Winkel zählt von 0 (links nach rechts) entgegen dem Uhrzeigersinn. *text* darf UTF-8 Zeichensequenzen enthalten, beispielsweise in der Form `{`.

Die Funktion gibt ein Array mit den Abmaßen der Textbox zurück, ähnlich der Funktion ImageTTFBBox.

```
<?php
header("Content-type: image/gif");
$im = imagecreate(400,30);
$black = ImageColorAllocate($im, 0,0,0);
$white = ImageColorAllocate($im, 255,255,255);
ImageTTFText($im, 20, 0, 10, 20, $white,
             "/path/arial.ttf",
             "Testing... Omega: &#937;");
ImageGif($im);
ImageDestroy($im);
?>
```

Diese Funktion benötigt neben der GD- auch die Freetype-Bibliothek. Mehr Informationen zu Freetype finden Sie unter:

- <http://www.freetype.org>

imagettftext



Siehe auch `imagettfbbox`.

imagecolorat

imagecolorat `imagecolorat` ermittelt den Farbindex eines Bildpunktes.

◆ `int imagecolorat(int im, int x,inty);`

Siehe auch `imagecolorset` und `imagecolorsforindex`.

imagecolorclosest

imagecolorclosest `imagecolorclosest` ermittelt den zu einem Bildpunkt passenden Index des in der Farbpalette nächstliegenden Wertes.

◆ `int imagecolorclosest(int im, int red, int green, int blue);`

Der Abstand wird im dreidimensionalen RGB-Farbwürfel gemessen.

Siehe auch `imagecolorexact`.

imagecolorexact

imagecolorexact `imagecolorexact` ermittelt den exakten Index der Farbe.

◆ `int imagecolorexact(int im, int red, int green, int blue);`

Der Farbwert einer Farbe in der Palette wird ermittelt. Wenn der Farbwert in der Palette nicht existiert, wird -1 zurückgegeben.

Siehe auch `imagecolorclosest`.

imagecolorresolve

imagecolorresolve `imagecolorresolve` ermittelt den exakten Index der Farbe oder, wenn dies nicht erfolgreich ist, die nächstliegende Farbe.

◆ `int imagecolorresolve(int im, int red, int green, int blue);`

Siehe auch `imagecolorclosest`.

imagecolorset

imagecolorset `imagecolorset` setzt eine Farbe auf einen Index der Palette.

◆ `bool imagecolorset(int im, int index,
int red, int green, int blue);`

Durch Setzen einer Farbe auf einen vorhandenen Index kann die Farbe eines Bildteiles global umgeschaltet werden. Dies ist deutlich schneller als beispielsweise eine der Füllfunktionen.

Siehe auch `imagecolorat`.

imagecolorsforindex

`imagecolorsforindex` ermittelt die Farbe eines Indizes der aktuellen Palette des Bildes.

imagecolorsforindex

```
array imagecolorsforindex(int im, int index);
```



Die Funktion gibt ein assoziatives Array mit den Schlüsseln "red", "green" und "blue" zurück.

Siehe auch `imagecolorat` und `imagecolorexact`.

imagecolorstotal

`imagecolorstotal` ermittelt die Anzahl der Farben der aktuellen Palette des Bildes *im*.

imagecolorstotal

```
int imagecolorstotal(int im);
```



Siehe auch `imagecolorat` und `imagecolorsforindex`.

imagepsloadfont

`imagepsloadfont` lädt einen Type-1-Font.

imagepsloadfont

```
int imagepsloadfont(string filename);
```



Die Funktion gibt ein Handle auf den Font zurück oder FALSE, wenn der Versuch misslang.

Siehe auch `imagepsfreefont`.

imagepsfreefont

`imagepsfreefont` gibt Speicher frei, der von einem Type-1-Font benutzt wird.

imagepsfreefont

```
void imagepsfreefont(int fontindex);
```



Siehe auch `imagepsloadfont`.

imagepsencodfont

imageps encodfont

imagepsencodfont ändert den Codevektor eines Fonts. Die Funktion lädt den Vektor aus einer Datei, um Zeichen größer als 127 in einer Postscriptdatei zu verwenden.



```
int imagepsencodfont(string encodingfile);
```

imagepstext

imagepstext

imagepstext zeichnet eine Zeichenkette mit Type-1-Schrift.



```
array imagepstext(int image, string text, int font, int size,
                  int foreground, int background,
                  int x, int y, int space, int tightness,
                  float angle, int antialias_steps);
```

Die Parameter haben folgende Bedeutung:

- *size* bestimmt die Größe in Pixel.
- *foreground* und *background* bestimmen die Vorder- und Hintergrundfarbe. Die Hintergrundfarbe wird benötigt, um die Farbüberblendung für die Weichzeichnung (Antialiasing) zu steuern.
- *x*, *y* bestimmen die linke untere Ecke des ersten Zeichens.
- *space* bestimmt den Zeichenzwischenraum; der Wert kann auch negativ sein.
- *tightness* bestimmt die Breite eines Leerzeichens.
- *angle* ist der Winkel, um den der Text gedreht wird.
- *antialias_steps* bestimmt die Anzahl der Farben, die für eine Weichzeichnung am Rand verwendet werden. Die Werte dürfen zwischen 4 und 16 liegen.



Beispiel

Die Parameter *space* und *tightness* werden in 1/1000-stel der Breite des Buchstaben »m« gemessen (Einheit em).

Die Parameter *space*, *tightness*, *angle* und *antialias_steps* sind optional.

Die Funktion gibt ein Array zurück, das die Abmaße des erzeugten Textes enthält. Die Indizes des Arrays haben folgende Bedeutung:

- [0]: linke untere X-Koordinate

- [1]: linke untere Y-Koordinate
- [2]: untere rechte X-Koordinate
- [3]: untere rechte Y-Koordinate

Siehe auch `imagepsbbox`.

imagepsbbox

`imagepsbbox` ermittelt die Maße eines Textes, die in Anspruch genommen werden, wenn der Text erzeugt würde.

imagepsbbox

```
array imagepsbbox(string text, int font, int size,  
                  int space, int width, float angle);
```



Die Parameter haben folgende Bedeutung:

- *size* bestimmt die Größe in Pixel.
- *space* bestimmt den Zeichenzwischenraum; der Wert kann auch negativ sein.
- *tightness* bestimmt die Breite eines Leerzeichens.
- *angle* ist der Winkel, um den der Text gedreht wird.

Die Parameter *space* und *tightness* werden in 1/1000-stel der Breite des Buchstaben »m« gemessen (Einheit em). Die Parameter *space*, *tightness* und *angle* sind optional.



Hinweis

Die Funktion gibt ein Array zurück, das die Abmaße des erzeugten Textes enthält. Die Indizes des Arrays haben folgende Bedeutung:

- [0]: linke untere X-Koordinate
- [1]: linke untere Y-Koordinate
- [2]: untere rechte X-Koordinate
- [3]: untere rechte Y-Koordinate

Siehe auch `imagepsbbox`.

iptcparse

`iptcparse` zerlegt einen binären IPTC-Block in einzelne Tags.

iptcparse

```
array iptcparse(string iptcblock);
```



Die Funktion gibt ein Array mit IPTC-Markierungen zurück, wobei *gmarker* den Schlüssel bildet und die Elemente die Werte enthalten. Werden keine Markierungen gefunden, wird FALSE zurückgegeben.



Mehr Informationen zu IPTC finden Sie unter folgender Adresse:

- <http://www.xe.net/iptc/>

Siehe auch `getImageSize` für ein Anwendungsbeispiel.

Hyperwave-Funktionen (hw)

Informationen zu Hyperwave erhalten Sie unter der folgenden Adresse:

- <http://www.hyperwave.com>

hw_array2objrec

hw_array2objrec Konvertiert die Daten aus einem Objekt-Array in einen Objekt-Datensatz, der als Zeichenkette zurückgegeben wird.

◆ `string hw_array2objrec(array object_array)`

hw_children

hw_children Gibt ein Array mit Objekt-IDs zurück. Jede ID ist einem Teil des Kind *objectID* zugeordnet. Das Array enthält sowohl Dokumente als auch Kollektionen.

◆ `array hw_children(int connection, int objectID)`

hw_childrenobj

hw_childrenobj Gibt ein Array mit Objekt-Arrays zurück. Jede ID ist einem Teil des Kind *objectID* zugeordnet. Das Array enthält sowohl Dokumente als auch Kollektionen.

◆ `array hw_childrenobj(int connection, int objectID)`

hw_close

hw_close Schließt die Verbindung zu einem Hyperwave-Server. Die Funktion gibt FALSE zurück, wenn keine Verbindung bestand, sonst TRUE.

◆ `bool hw_close(int connection)`

hw_connect

hw_connect öffnet eine Verbindung zu einem Hyperwave-Server.

```
int hw_connect(string host, int port,
               [string username], [string password]);
```

Siehe auch hw_pconnect.

hw_connect**hw_cp**

hw_cp kopiert Objekte.

```
int hw_cp(int connection, array object_id_array,
           int destination id);
```

Die Funktion gibt die Anzahl der kopierten Objekte zurück.

Siehe auch hw_mv.

hw_cp**hw_deleteobject**

hw_deleteobject löscht Objekte.

```
int hw_deleteobject(int connection, int object_to_delete);
```

Siehe auch hw_mv.

hw_deleteobject**hw_docbyanchor**

hw_docbyanchor gibt die Objekt-ID eines durch einen Anker adressierten Dokuments zurück.

```
int hw_docbyanchor(int connection, int anchorID);
```

hw_docbyanchor**hw_docbyanchorobj**

hw_docbyanchorobj gibt ein Datensatzobjekt des Dokuments zurück, auf das *anchorID* zeigt.

```
string hw_docbyanchorobj(int connection, int anchorID);
```

hw_docbyanchorobj**hw_documentattributes**

hw_documentattributes ermittelt das Datensatzobjekt eines Dokuments und gibt es als Zeichenkette zurück.

```
string hw_documentattributes(int hw_document);
```

Siehe auch hw_DocumentBodyTag, hw_DocumentSize.

hw_documentattributes

hw_documentbodytag

hw_documentbodytag hw_documentbodytag gibt das Body-Tag des Dokuments *hw_document* zurück.

- ◆ `string hw_documentbodytag(int hw_document);`
Siehe auch `hw_DocumentAttributes`, `hw_DocumentSize`.

hw_documentcontent

hw_documentContent Die Funktionen `hw_documentcontent` gibt den Inhalt des Dokuments *hw_document* zurück.

- ◆ `string hw_documentcontent(int hw_document);`
Siehe auch `hw_DocumentAttributes`, `hw_DocumentSize`, `hw_DocumentSetContent`.

hw_documentsetcontent

hw_documentsetcontent `hw_documentsetcontent` setzt oder ersetzt den Inhalt eines Dokuments *hw_document*.

- ◆ `string hw_documentsetcontent(int hw_document, string content);`
Siehe auch `hw_DocumentAttributes`, `hw_DocumentSize`, `hw_DocumentContent`.

hw_documentsize

hw_documentsize `hw_documentsize` ermittelt die Größe des Dokuments *hw_document* in Bytes.

- ◆ `int hw_documentsize(int hw_document);`
Siehe auch `hw_DocumentBodyTag`, `hw_DocumentAttributes`.

hw_errormsg

hw_errormsg `hw_errormsg` gibt die letzte Fehlermeldung zurück.

- ◆ `string hw_errormsg(int connection);`
Wenn kein Fehler vorliegt, wird »No Error« zurückgegeben.

hw_edittest

hw_edittest `hw_edittest` lädt ein Text-Dokument auf den Hyperwave-Server.

- ◆ `int hw_edittest(int connection, int hw_document);`

Siehe auch `hw_PipeDocument`, `hw_FreeDocument`, `hw_DocumentBodyTag`, `hw_DocumentSize`, `hw_OutputDocument`, `hw_GetText`.

hw_error

`hw_error` gibt die letzte Fehlernummer zurück oder 0, wenn keine Fehler auftraten.

```
int hw_error(int connection);
```

Siehe auch `hw_ErrorMsg`.

hw_free_document

`hw_free_document` gibt den von einem Dokument belegten Speicher wieder frei.

```
int hw_free_document(int hw_document);
```

hw_getparents

`hw_getparents` gibt ein Array von übergeordneten Objekten zu der angegebenen Objekt-ID *objectID* zurück.

```
array hw_getparentsobj(int connection, int objectID);
```

hw_getparentsobj

`hw_getparentsobj` gibt ein Array von Datensätzen mit Informationen übergeordneter Objekte zurück.

```
array hw_getparentsobj(int connection, int objectID);
```

hw_getchildcoll

`hw_getchildcoll` gibt die Kollektion der untergeordneten Objekte in einem Array zurück.

```
array hw_getchildcoll(int connection, int objectID);
```

Das Array enthält die IDs, nicht jedoch die Dokumente.

Siehe auch `hw_getchildren`, `hw_getchilddoccoll`.

hw_getchildcollobj

`hw_getchildcollobj` ermittelt die Datensätze der untergeordneten Objekte.

```
array hw_getchildcollobj(int connection, int objectID);
```

Siehe auch `hw_childrenobj`, `hw_getchilddoccollobj`.

hw_getremote

hw_getremote `hw_getremote` holt ein Dokument von einem anderen Server, üblicherweise über FTP, HTTP und einige Datenbanken.

◆ `int hw_getremote(int connection, int objectID);`

Siehe auch `hw_getremotechildren`.

hw_getremotechildren

hw_getremotechildren `hw_getremotechildren` holt untergeordnete Komponenten eines Dokuments auf einem anderen Server.

◆ `int hw_getremotechildren(int connection, string object record);`

Siehe auch `hw_getremote`.

hw_getsrcbydestobj

hw_getsrcbydestobj `hw_getsrcbydestobj` gibt einen Anker zurück, der auf ein Datensatzobjekt zeigt.

◆ `array hw_getsrcbydestobj(int connection, int objectID);`

Siehe auch `hw_getanchors`.

hw_getobject

hw_getobject `hw_getobject` gibt das Datensatzobjekt selbst zurück.

◆ `array hw_getobject(int connection, int objectID);`

Siehe auch `hw_getandlock`.

hw_getandlock

hw_getandlock `hw_getandlock` gibt ein Datensatzobjekt zurück und verriegelt den Zugriff durch andere.

◆ `string hw_getandlock(int connection, int objectID);`

Siehe auch `hw_unlock`, `hw_getobject`.

hw_gettext

hw_gettext `hw_gettext` gibt das durch `objectID` bezeichnete Textdokument zurück.

```
int hw_gettext(int connection, int objectID, int rootID);
```



Der optionale Parameter *rootID* bestimmt die Behandlung untergeordneter Knoten. Damit wird die Abbildung aus dem Web ermittelter Seiten auf das lokale Dateisystem gesteuert.

Siehe auch `hw_pipedocument`, `hw_freedocument`, `hw_documentbodytag`, `hw_documentsize`, `hw_outputdocument`.

hw_getobjectbyquery

`hw_getobjectbyquery` gibt ein Objekt auf gefundene Dokumente nach einer Suchanfrage zurück.

**hw_getobject
byquery**

```
array hw_getobjectbyquery(int connection, string query,  
                           int max_hits);
```



Die maximale Anzahl der Treffer wird mit *max_hits* angegeben, -1 steht für unbegrenzt.

Siehe auch `hw_getobjectbyqueryobj`.

hw_getobjectbyqueryobj

`hw_getobjectbyqueryobj` gibt ein Array von Objekten auf gefundene Dokumente nach einer Suchanfrage zurück.

**hw_getobject
byqueryobj**

```
array hw_getobjectbyqueryobj(int connection, string query,  
                              int max_hits);
```



Die maximale Anzahl der Treffer wird mit *max_hits* angegeben, -1 steht für unbegrenzt.

Siehe auch `hw_getobjectbyquery`.

hw_getobjectbyquerycoll

`hw_getobjectbyquerycoll` gibt eine Kollektion von Dokumentenobjekten in einem Array zurück.

**hw_getobject
byquerycoll**

```
array hw_getobjectbyquerycoll(int connection, int objectID,  
                               string query, int max_hits);
```



Die maximale Anzahl der Treffer wird mit *max_hits* angegeben, -1 steht für unbegrenzt.

Siehe auch `hw_getobjectbyquerycollobj`.

hw_getobjectbyquerycollobj

hw_getobjectbyquerycollobj hw_getobjectbyquerycollobj gibt eine Kollektion von Dokumentenobjekten in einem Array zurück.

◆ array **hw_getobjectbyquerycollobj**(int connection, int objectID, string query, int max_hits);

Die maximale Anzahl der Treffer wird mit *max_hits* angegeben, -1 steht für unbegrenzt.

Siehe auch hw_getobjectbyquerycoll.

hw_getchilddoccoll

hw_getchilddoccoll hw_getchilddoccoll gibt ein Array von Objekt-IDs auf untergeordnete Dokumente zurück.

◆ array **hw_getchilddoccoll**(int connection, int objectID);

Siehe auch hw_getchildren, hw_getchildcoll.

hw_getchilddoccollobj

hw_getchilddoccollobj hw_getchilddoccollobj gibt ein Array von Objekten auf untergeordnete Dokumente zurück.

◆ array **hw_getchilddoccollobj**(int connection, int objectID);

Siehe auch hw_childrenobj, hw_getchildcollobj.

hw_getanchors

hw_getanchors hw_getanchors gibt die Objekt-IDs der Anker im Dokument zurück.

◆ array **hw_getanchors**(int connection, int objectID);

hw_getanchorsobj

hw_getanchorsobj hw_getanchorsobj gibt die Datensatzobjekte der Anker im Dokument in einem Array zurück.

◆ array **hw_getanchorsobj**(int connection, int objectID);

hw_mv

hw_mv hw_mv verschiebt Objekte.

◆ int **hw_mv**(int connection, array object id array, int source id, int destination id);

Die Funktion gibt die Anzahl der verschobenen Objekte zurück.

Siehe auch `hw_cp`, `hw_deleteobject`.

hw_identify

`hw_identify` identifiziert sich als Nutzer mit den angegebenen Daten.

```
int hw_identify(string username, string password);
```

Siehe auch `hw_connect`.

hw_incollections

`hw_incollections` prüft auf Objekt-IDs in einer Kollektion.

```
array hw_incollections(int connection, array object_id_array,
                      array collection_id_array,
                      int return_collections);
```

hw_info

`hw_info` ermittelt Informationen über die aktuelle Verbindung.

```
string hw_info(int connection);
```

Die zurückgegebene Zeichenkette hat folgenden Aufbau:

- <Serverstring>, <Host>, <Port>, <Username>, <Port of Client>, <Byte swapping>

hw_inscoll

`hw_inscoll` fügt eine Kollektion ein.

```
int hw_inscoll(int connection, int objectID, array object_array);
```

hw_insdoc

`hw_insdoc` fügt ein Dokument ein.

```
int hw_insdoc(int connection, int parentID,
              string object_record, string text);
```

Siehe auch `hw_insertdocument`, `hw_inscoll`.

hw_insertdocument

`hw_insertdocument` lädt ein Dokument auf den Hyperwave-Server *connection*.

◆ `int hw_insertdocument(int connection, int parent_id,
int hw_document);`

Siehe auch `hw_pipedocument`.

hw_insertobject

hw_insertobject `hw_insertobject` fügt ein Datensatzobjekt ein.

◆ `int hw_insertobject(int connection, string object rec,
string parameter);`

Siehe auch `hw_pipedocument`, `hw_insertdocument`, `hw_insdoc`, `hw_inscoll`.

hw_modifyobject

hw_modifyobject `hw_modifyobject` verändert ein Datensatzobjekt.

◆ `int hw_modifyobject(int connection, int object_to_change,
array remove, array add, int mode);`



```
// $connect ist eine Hyperwave-Verbindung
// $objid ist die ID des zu ändernden Objekts
$remarr = array("Name" => "books");
$addarr = array("Name" => "articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
```

Um ein Wertepaar zu löschen, wird als dritter Parameter lediglich ein leeres Array übergeben.



```
// Neuen Wert hinzufügen
// $connect ist eine Hyperwave-Verbindung
// $objid ist die ID des zu ändernden Objekts
$remarr = array("Name" => 0);
$addarr = array("Name" => "articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
```



```
$remarr = array("Title" => "en:Books");
$addarr = array("Title" => "en:Articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
```



```
$remarr = array("Title" => array("en" => "Books"));
$addarr = array("Title" =>
    array("en" => "Articles", "ge"=>"Artikel"));
$hw_modifyobject($connect, $objid, $remarr, $addarr);
```



```
$remarr = array("Title" => "");
$addarr = array("Title" => "en:Articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
```

Die Funktion gibt im Erfolgsfall TRUE zurück, sonst FALSE.

hw_new_document

hw_new_document erzeugt ein neues Dokument. Die Funktion übergibt das Dokument noch nicht an den Hyperwave-Server.

```
int hw_new_document(string object_record, string document_data,
                   int document_size);
```

Die Funktion übergibt das Dokument noch nicht an den Hyperwave-Server.

Siehe auch hw_FreeDocument, hw_DocumentSize, hw_DocumentBodyTag, hw_OutputDocument, hw_InsertDocument.

hw_new_document



hw_objrec2array

hw_objrec2array konvertiert ein Datensatzobjekt in ein Array.

```
array hw_objrec2array(string object_record);
```

hw_objrec2array



hw_outputdocument

hw_outputdocument zeigt ein Dokument an.

```
int hw_outputdocument(int hw_document);
```

hw_output



hw_pconnect

hw_pconnect stellt eine persistente (ständige) Verbindung mit einem Hyperwave-Server her.

```
int hw_pconnect(string host, int port,
                string username, string password);
```

hw_pconnect



Die Funktion gibt ein Handle für die Verbindung zurück.

Siehe auch hw_connect.

hw_pipedocument

hw_pipedocument lädt ein Dokument auf den Hyperwave-Server.

```
int hw_pipedocument(int connection, int objectID);
```

hw_pipe



Enthält das Dokument Anker, auch die daran verknüpften Dokumente werden übertragen.

Siehe auch hw_gettext für mehr Informationen über Links und Anker sowie hw_freedocument, hw_documentsize, hw_documentbodytag, hw_outputdocument.

hw_root

hw_root hw_root gibt die ID des Root-Objekts zurück.

◆ `int hw_root();`

hw_unlock

hw_unlock hw_unlock gibt ein Dokument wieder frei.

◆ `int hw_unlock(int connection, int objectID);`

Siehe auch `hw_getandlock`.

hw_who

hw_who hw_who gibt eine Liste der aktuell verbundenen Nutzer an.

◆ `array hw_who(int connection);`

Gibt ein Array zurück, das für jeden Nutzer ein weiteres Array enthält, das die folgenden Informationen beinhaltet:

- `id`, `name`, `system`, `onSinceDate`, `onSinceTime`, `TotalTime` und `self`.

Der Eintrag `self` ist 1, wenn die Abfrage von diesem Nutzer erfolgte.

hw_username

hw_username hw_username gibt den Namen des Nutzers der aktuellen Verbindung zurück.

◆ `string hw_getusername(int connection);`

A.8 Datenbankfunktionen

Dieser Teil der Referenz enthält nur die Funktionen der wichtigsten Datenbanksysteme. Es gibt weitere Funktionen, auch für exotische Programme. Konsultieren Sie ggf. die Online-Dokumentation auf der CD.

DBM-Funktionen

dbmopen

dbmopen öffnet eine dbm-Datenbank.

```
int dbmopen(string filename, string flags);
```

Das erste Argument ist der Pfad zur Datenbank, der Parameter *flags* gibt an, wie die Datenbank geöffnet werden soll:

- "r". Nur zum Lesen öffnen.
- "n". Neu anlegen und zum Schreiben und Lesen öffnen.
- "c". Neu anlegen und zum Schreiben und Lesen öffnen. Bestehende Datenbanken gleichen Namens bleiben erhalten.
- "w". Lesen und Schreiben.

Siehe auch: <ftp://prep.ai.mit.edu/pub/gnu>.

dbmclose

dbmclose schließt eine Datenbank.

```
bool dbmclose(int dbm_identifizier);
```

dbmexists

Die Funktion dbmexists ermittelt, ob zu einem Schlüssel ein Wert in der Datenbank existiert.

```
bool dbmexists(int dbm_identifizier, string key);
```

Die Funktion gibt TRUE zurück, wenn der Wert existiert.

dbmfetch

dbmfetch ermittelt einen Wert zu einem Schlüssel.

```
string dbmfetch(int dbm_identifizier, string key);
```

dbmopen



dbmclose



dbmexists



dbmfetch



Gibt aus der Datenbank *dbm_identifier* den Wert des Schlüssels *key* zurück.

dbminsert

dbminsert

dbminsert fügt den Wert zu einem Schlüssel ein.



```
int dbminsert(int dbm_identifier,
              string key,
              string value);
```

Schreibt einen Wert mit dem Schlüssel *key* in die Datenbank. Gibt -1 zurück, wenn die Datenbank nur zum Lesen geöffnet war. 0 wird im Erfolgsfall zurückgegeben und 1, wenn der Schlüssel bereits existierte.

Um Werte zu ersetzen, verwenden Sie `dbmreplace`.

dbmreplace

dbmreplace

dbmreplace ersetzt einen Wert.



```
bool dbmreplace(int dbm_identifier,
                string key,
                string value);
```

Existiert der Schlüssel noch nicht, wird er hinzugefügt.

dbmdelete

dbmdelete

dbmdelete löscht einen Wert.



```
bool dbmdelete(int dbm_identifier, string key);
```

Wenn der Schlüssel nicht existiert, wird FALSE zurückgegeben.

dbmfirstkey

dbmfirstkey

dbmfirstkey ermittelt den ersten Schlüssel in der Datenbank.



```
string dbmfirstkey(int dbm_identifier);
```

Die Reihenfolge der Datensätze ist nicht vorhersagbar.

dbmnextkey

dbmnextkey

dbmnextkey ermittelt den nächsten Schlüssel.



```
string dbmnextkey(int dbm_identifier,
                  string key);
```

Zusammen mit der Funktion `dbmfirstkey` können alle Daten aus der Datenbank gelesen werden.

```
$key = dbmfirstkey($dbm_id);
while ($key) {
    echo "$key = " . dbmfetch($dbm_id, $key) . "\n";
    $key = dbmnextkey($dbm_id, $key);
}
```



Beispiel

dblist

`dblist` beschreibt die genutzte Bibliothek.

```
string dblist(void);
```

dblist



dBase-Funktionen

dbase_create

Die Funktion `dbase_create` erzeugt eine neue dBase-Datenbank.

```
int dbase_create(string filename, array fields);
```

dbase_create



Der Parameter *fields* ist ein Array aus Arrays. Jedes Array beschreibt ein Feld der Datenbank mit Name, Feldtyp, Länge und Genauigkeit. Der Feldtyp wird als Zeichen angegeben:

- L. *Boolean*. Länge und Genauigkeit entfallen.
- M. *Memo*. Länge und Genauigkeit entfallen. Eine direkte Unterstützung für Memo-Felder gibt es in PHP nicht.
- D. *Date* (Format: YYYYMMDD). Länge und Genauigkeit entfallen.
- N. *Number*. Länge und Genauigkeit müssen angegeben werden. Die Genauigkeit bezeichnet dabei die Anzahl der Dezimalstellen.
- C. *String*. Zeichenkette.

```
// Name der Datenbank
$dbname = "/tmp/test.dbf";
$def =
array(
array("date", "D"),
array("name", "C", 50),
array("age", "N", 3, 0),
array("email", "C", 128),
array("ismember", "L")
);
```



Beispiel

```
// Datenbank erzeugen
if (!dbase_create($dbname, $def))
print "<strong>Error!</strong>";
```

dbase_open

dbase_open Die Funktion `dbase_open` öffnet eine bestehende dBase-Datenbank.

◆ `int dbase_open(string filename, int flags);`

Der Parameter *flags* kann folgendermaßen angegeben werden:

- 0. Read-Only, Nur-Lese-Zugriff.
- 1. Write-Only, Nur Schreib-Zugriff.
- 2. Read-Write, Lesen und Schreiben.

Die Funktion gibt ein Handle zurück, wenn das Öffnen erfolgreich war, sonst `false`.

dbase_close

dbase_close Die Funktion `dbase_close` schließt eine Datenbank wieder.

◆ `bool dbase_close(int dbase_identifizier);`

dbase_identifizier ist der von `dbase_open` vergebene Handle.

dbase_pack

dbase_pack Die Funktion `dbase_pack` packt eine dBase-Datenbank. Unter Packen versteht dBase das endgültige Löschen von zum Löschen markierten Datensätzen.

◆ `bool dbase_pack(int dbase_identifizier);`

dbase_identifizier ist der von `dbase_open` vergebene Handle.

dbase_add_record

dbase_add_record `dbase_add_record` fügt einer dBase-Datenbank einen neuen Datensatz hinzu.

◆ `bool dbase_add_record(int dbase_identifizier,
array record);`

Das Array *record* wird dem Handle *identifizier* als neuer Datensatz hinzugefügt. Wenn die Anzahl der Elemente des Arrays nicht mit der Anzahl der Felder in der Datenbank übereinstimmt, wird `FALSE` zurückgegeben und der Prozess schlägt fehl.

`dbase_replace_record`

Die Funktion `dbase_replace_record` ersetzt einen Datensatz in einer dBase-Datenbank.

`dbase_replace_record`

```
bool dbase_replace_record(int dbase_identifizier,  
                           array record,  
                           int dbase_record_number);
```



Ersetzt die Daten des durch *record_number* ausgewählten Datensatzes mit den Daten des Arrays *record*. Wenn die Anzahl der Elemente des Arrays nicht mit der Anzahl der Felder in der Datenbank übereinstimmt, wird FALSE zurückgegeben und der Prozess schlägt fehl. Der Parameter *dbase_record_number* darf zwischen 1 und Maximalanzahl der Datensätze liegen. Die Maximalzahl der Datensätze können Sie mit der Funktion *dbase_numrecords* ermitteln.

`dbase_delete_record`

Die Funktion `dbase_delete_record` löscht einen Datensatz aus der dBase-Datenbank.

`dbase_delete_record`

```
bool dbase_delete_record(int dbase_identifizier, int record);
```



Die Funktion markiert den Datensatz *dbase_identifizier* zum Löschen. Um die Daten physisch zu entfernen, müssen Sie `dbase_pack` aufrufen.

`dbase_get_record`

Die Funktion `dbase_get_record` holt die Daten eines Datensatzes aus der dBase-Datenbank.

`dbase_get_record`

```
array dbase_get_record(int dbase_identifizier, int record);
```



Gibt die Daten des Datensatzes *record* der Datenbank *dbase_identifizier* in einem Array zurück. Das Array ist mit 1 beginnend indiziert und enthält ein assoziatives Element ["delete"], das gelöschte, aber noch nicht gepackte Datensätze kennzeichnet. Feldtypen werden in PHP-Datentypen konvertiert. Datumsfelder werden als Zeichenketten zurückgegeben.

`dbase_numfields`

Die Funktion `dbase_numfields` gibt die Anzahl der Felder in der Datenbank zurück.

`dbase_numfields`

```
int dbase_numfields(int dbase_identifizier);
```



Die Anzahl der Felder kann zur Adressierung von Arrays genutzt werden. Beachten Sie, dass das erste Feld mit 0 indiziert wird, die Anzahl also zwischen 0 und `dbase_numfields($db)-1` liegt. Im Gegensatz dazu wird der erste Datensatz mit 1 adressiert.



```
$rec = dbase_get_record($db, $recno);
$nf = dbase_numfields($db);
for ($i=0; $i < $nf; $i++) {
    print $rec[$i]."<br>\n";
}
```

dbase_numrecords

dbase_numrecords

Die Funktion `dbase_numrecords` gibt die Anzahl der Datensätze in einer dBase-Datenbank zurück.



```
int dbase_numrecords(int dbase_identifizier);
```

Die Datensätze zählen mit 1 beginnend.

filePro-Funktionen

filepro

filepro

Die Funktion `filepro` liest und prüft die Map-Datei einer FilePro-Datenbank.



```
bool filepro(string directory);
```

Die Funktion speichert die Anzahl der Felder und die Feldinformationen in der Map-Datei. Die Datenbank wird dabei nicht verriegelt. Sie sollten sicherstellen, dass während der Ausführung keine weiteren Zugriffe auf das Datenverzeichnis erfolgen.

filepro_fieldname

filepro_fieldname

Die Funktion `filepro_fieldname` ermittelt den Namen eines Feldes.



```
string filepro_fieldname(int field_number);
```

Der Feldname wird zu einer gegebenen Feldnummer `field_number` ermittelt.

filepro_fieldtype

filepro_fieldtype

`filepro_fieldtype` ermittelt den Datentyp eines Felds.



```
string filepro_fieldtype(int field_number);
```

filepro_fieldwidth

filepro_fieldwidth gibt die Breite eines Felds an.

```
int filepro_fieldwidth(int field_number);
```

filepro_fieldwidth**filepro_retrieve**

Die Funktion filepro_retrieve holt eine Spalte aus der Datenbank.

```
string filepro_retrieve(int row_number, int field_number);
```

filepro_retrieve**filepro_fieldcount**

filepro_fieldcount zählt die Anzahl der Felder (Spalten) der geöffneten filePro-Datenbank.

```
int filepro_fieldcount(void);
```

Siehe auch filepro.

filepro_fieldcount**filepro_rowcount**

filepro_rowcount gibt die Anzahl der Datensätze der geöffneten Datenbank zurück.

```
int filepro_rowcount(void);
```

Siehe auch filepro.

filepro_rowcount**Informix-Funktionen**

Der Informix-Treiber für Online (ODS) 7.x, SE 7.x und Universal Server (IUS) 9.x ist in »functions/ifx.ec« und »functions/php3_ifx.h« implementiert. ODS 7.x unterstützt alle Funktionen. IUS 9.x unterstützt SLOB und CLOB nicht (PHP 3.0.11).

Setzen Sie zuerst die INFORMIXDIR-Variablen. Das Konfigurationsskript erkennt automatisch alle Bibliotheken, wenn es mit

```
Configure -with_informix=yes"
```

gestartet wird. Sie können die automatische Erkennung unterdrücken, indem »IFX_LIBDIR«, »IFX_LIBS« und »IFX_INCDIR« im Umgebungsbereich angegeben wird.

Konfiguration**ifx_connect**

ifx_connect öffnet eine Informix-Server-Verbindung.

ifx_connect

◆

```
int ifx_connect(string [database],
               string [userid],
               string [password]);
```

Die Funktion gibt ein Handle zurück, wenn die Verbindung erfolgreich war, sonst FALSE. Die Argumente sind optional. Wenn keine Argumente angegeben werden, entnimmt PHP die folgenden Werte der Konfigurationsdatei PHP.INI:

- ifx.default_host. Hostname. Wenn kein Name angegeben wurde, versucht der Informix-Server die Umgebungsvariable \$INFORMIXSERVER zu lesen.
- ifx.default_user. Nutzernamen.
- ifx.default_password. Kennwort.

Beim erneuten Aufruf wird keine neue Verbindung gestartet, sondern eine mit den gleichen Parametern bestehende Verbindung genutzt. Zum Schließen der Verbindung wird die Funktion ifx_close verwendet.

Siehe auch ifx_pconnect und ifx_close.



Beispiel

```
$conn_id = ifx_connect (mydb@ol_srv1, "imyself", "mypassword");
```

ifx_pconnect

ifx_pconnect

ifx_pconnect öffnet eine persistente Datenbankverbindung zu einem Informix-Server.

◆

```
int ifx_pconnect(string [database],
                string [userid],
                string [password]);
```

Die Funktion gibt ein Handle zurück, wenn die Verbindung erfolgreich war, sonst FALSE. Die Argumente sind optional. Die Anwendung entspricht ifx_connect.

Die Funktion öffnet eine persistente Verbindung, das heißt, die Verbindung wird mit dem Ende des Skripts nicht unterbrochen. Wenn in einem anderen Skript erneut ein Aufruf der Funktion mit identischen Parametern erfolgt, dann wird die bestehende Verbindung genutzt. Diese Eigenschaft hat die Funktion nur, wenn PHP als Modul im Apache oder IIS installiert wurde, nicht als CGI-Programm.

Siehe auch unter ifx_connect.

ifx_close

ifx_close schließt eine Verbindung zu einer Informix-Datenbank.

```
int ifx_close(int [link_identifizier]);
```

Die Funktion gibt immer TRUE zurück. Die Anwendung ist nur innerhalb eines Skripts sinnvoll, denn am Ende des Skripts wird die Verbindung automatisch beendet.

ifx_close schließt keine persistenten Verbindungen, die mit ifx_pconnect geöffnet wurden.

Siehe auch: ifx_connect und ifx_pconnect.

```
$conn_id = ifx_connect (mydb@ol_srv,
                      "itsme",
                      "mypassword");
... code arbeitet mit Informix ...
ifx_close($conn_id);
```

ifx_close**ifx_query**

ifx_query sendet eine SQL-Abfrage an den Informix-Server.

```
int ifx_query(string query,
             int [link_identifizier],
             int [cursor_type],
             mixed [blobidarray]);
```

ifx_query

Die Funktion gibt ein Handle auf eine Ergebnisliste zurück. Wenn die Abfrage misslingt, wird FALSE zurückgegeben. Anschließend kann die Anzahl bearbeiteter Datensätze mit der Funktion ifx_affected_rows ermittelt werden.

Wenn die Verbindung geschlossen ist, wird implizit ifx_open aufgerufen. Wenn eine Verbindung besteht, kann der Parameter link_identifizier entfallen – es wird dann die zuletzt geöffnete Verbindung genutzt.

Der optionale Parameter cursor_type kann folgende Werte annehmen:

- IFX_SCROLL
- IFX_HOLD

blobidarray enthält die ID eines BLOB-Handle, die entsprechenden Spalten in der SQL-Anweisung sind dann durch ?-Zeichen zu ersetzen.



```
ifx_textasvarchar(1); // use "text mode" for blobs
$res_id = ifx_query("select * from orders", $conn_id);
if (! $res_id) {
    printf("Can't select orders : %s\n<br>%s<br>\n",
        ifx_error());
    ifx_errormsg();
    die;
}
ifx_htmltbl_result($res_id, "border=\"1\"");
ifx_free_result($res_id);
```



```
// create blob id's for a byte and text column
$textid = ifx_create_blob(0, 0,
    "Text column in memory");
$byteid = ifx_create_blob(1, 0,
    "Byte column in memory");
// store blob id's in a blobid array
$blobidarray[] = $textid;
$blobidarray[] = $byteid;
// launch query
$query = "insert into catalog (stock_num, manu_code,
    cat_descr,cat_picture)
    values(1,'HR0',?,?)";
$res_id = ifx_query($query, $conn_id, $blobidarray);
if (! $res_id) {
    ... error ...
}
// free result id
ifx_free_result($res_id);
```

ifx_prepare

ifx_prepare

ifx_prepare bereitet eine SQL-Anweisung zur Ausführung vor.



```
int ifx_prepare(string query, int conn_id, int[cursor_def],
    mixed blobidarray);
```

Gibt ein Handle zurück, das mit ifx_do genutzt werden kann.

Siehe auch ifx_do, ifx_query

ifx_do

ifx_do

Die Funktion ifx_do führt eine vorbereitete SQL-Anweisung aus.



```
int ifx_do(int result_id);
```

Gibt im Erfolgsfall TRUE zurück, sonst FALSE.

Die Funktion setzt den tatsächlichen Wert für ifx_affected_rows.

Siehe auch ifx_prepare, ifx_affected_rows.

ifx_error

ifx_error gibt den letzten Fehlercode zurück.

```
string ifx_error(void);
```

Der Informix-Fehlercode (SQLSTATE & SQLCODE) wird folgendermaßen formatiert:

```
x [SQLSTATE = aa bbb SQLCODE=cccc]
```

Wobei x folgende Bedeutung hat:

- Leerzeichen: Kein Fehler
- E : Fehler
- N : Keine Daten vorhanden
- W : Warnung
- ? : undefiniert

Siehe auch ifx_errormsg.

ifx_error**ifx_errormsg**

ifx_errormsg gibt die letzte Fehlermeldung zurück.

```
string ifx_errormsg(int [errorcode]);
```

Die Funktion kann mit dem Parameter *errorcode* aufgerufen werden, dann wird der Text des angegebenen Fehlers ausgegeben.

Siehe auch: ifx_error.

```
printf("%s\n<br>", ifx_errormsg(-201));
```

ifx_errormsg

Beispiel

ifx_affected_rows

ifx_affected_rows gibt die Anzahl der bearbeiteten Datensätze der letzten SQL-Anweisung zurück.

```
int ifx_affected_rows(int result_id);
```

result_id ist ein gültiges Handle, das von ifx_query oder ifx_prepare zurückgegeben wurde.

Siehe auch: ifx_num_rows

```
$rid = ifx_prepare ("select * from emp
                  where name like " .
                  $name, $connid);
if (! $rid) {
```

ifx_affected_rows

Beispiel

```

    ... error ...
}
$rowcount = ifx_affected_rows($rid);
if ($rowcount > 1000) {
    printf ("Too many rows in result set (%d)\n<br>",
           $rowcount);
    die ("Please restrict your query<br>\n");
}

```

ifx_getsqlca

ifx_getsqlca

Die Funktion `ifx_getsqlca` holt den Inhalt aus `sqlca.sqlerrd[0..5]` nach einer Abfrage.



```
array ifx_getsqlca(int result_id);
```

`result_id` ist ein gültiges Handle, das von `ifx_query` oder `ifx_prepare` zurückgegeben wurde. Zurückgegeben wird ein assoziatives Array mit den Werten `sqlca.sqlerrd[0]` bis `sqlca.sqlerrd[5]`.



Beispiel

```

/*assume the first column of 'sometable' is a serial*/
$qid = ifx_query("insert into sometable
                values(0,
                       '2nd column',
                       'another col-umn'
                ", $connid);
if (! $qid) {
    ... error ...
}
$sqlca = ifx_getsqlca ($qid);
$serial_value = $sqlca["sqlerrd1"];
echo "The serial value of the inserted row is : " .
     $serial_value<br>\n";

```

ifx_fetch_row

ifx_fetch_row

`ifx_fetch_row` holt einen Datensatz als assoziatives Array.



```
array ifx_fetch_row(int result_id,
                   mixed [position]);
```

BLOB-Spalten werden als ID-Wert (integer) zurückgegeben, die mit der Funktion `ifx_get_blob` ausgelesen werden können.

`[position]` steuert die Art des Dateicursors:

- "NEXT"
- "PREVIOUS"
- "CURRENT"
- "FIRST"

- "LAST"
- Eine Zahl springt direkt zu der angegebenen Position.

```

$rid = ifx_prepare ("select * from emp
                  where name like " . $name,
                  $connid, IFX_SCROLL);

if (! $rid) {
    ... error ...
}
$rowcount = ifx_affected_rows($rid);
if ($rowcount > 1000) {
    printf ("Too many rows in result set (%d)\n<br>",
           $rowcount);
    die ("Please restrict your query<br>\n");
}
if (! ifx_do ($rid)) {
    ... error ...
}
$row = ifx_fetch_row ($rid, "NEXT");
while (is_array($row)) {
    for(reset($row); $fieldname=key($row); next($row)) {
        $fieldvalue = $row[$fieldname];
        printf ("%s = %s,", $fieldname, $fieldvalue);
    }
    printf("\n<br>");
    $row = ifx_fetch_row ($rid, "NEXT");
}
ifx_free_result ($rid);

```



ifx_htmltbl_result

Die Funktion `ifx_htmltbl_result` formatiert eine Ergebnisliste als **ifx_htmltbl_result** HTML-Tabelle.

```
int ifx_htmltbl_result(int result_id, string [html_options]);
```



Die Funktion gibt die Anzahl der Zeilen zurück. *html_options* kann Argumente für das HTML-Tag `<TABLE>` enthalten.

```

$rid = ifx_prepare ("select * from emp
                  where name like " . $name,
                  $connid, IFX_SCROLL);

if (! $rid) {
    ... error ...
}
$rowcount = ifx_affected_rows ($rid);
if ($rowcount > 1000) {
    printf ("Too many rows in result set (%d)\n<br>",
           $rowcount);
    die ("Please restrict your query<br>\n");
}

```



```

if (! ifx_do($rid) {
    ... error ...
}
ifx_htmltbl_result ($rid, "border=\"2\"");
ifx_free_result($rid);

```

ifx_fieldtypes

ifx_fieldtypes

ifx_fieldtypes listet alle Felder auf.



```
array ifx_fieldtypes(int result_id);
```

Gibt ein assoziatives Array zurück, das als Schlüssel Feldnamen und als Werte Feldtypenbezeichner enthält.



```

$types = ifx_fieldtypes ($resultid);
if (! isset ($types)) {
    ... error ...
}
for ($i = 0; $i < count($types); $i++){
    $fname = key($types);
    printf("%s :\t type = %s\n", $fname,
        $types[$fname]);
    next($types);
}

```

ifx_fieldproperties

ifx_fieldproperties

ifx_fieldproperties gibt eine Liste mit den Feldeigenschaften zurück.



```
array ifx_fieldproperties(int result_id);
```

Gibt ein assoziatives Array mit Feldnamen als Schlüssel und Feldeigenschaften als Werte zurück. Im Fehlerfall wird FALSE zurückgegeben. Die Eigenschaften werden mit folgender Syntax ausgegeben:

```
"SQLTYPE;length;precision;scale;ISNULLABLE"
```

Die Bestandteile haben folgende Bedeutung:

- SQLTYPE. Der Informix-Type, beispielsweise SQLVCHAR.
- *length; precision; scale*. Länge, Genauigkeit, Skalierung.
- ISNULLABLE. Y oder N.



```

$properties = ifx_fieldtypes ($resultid);
if (! isset($properties)) {
    ... error ...
}
for ($i = 0; $i < count($properties); $i++) {

```

```

    $fname = key ($properties);
    printf ("%s:\t type = %s\n", $fname,
           $properties[$fname]);
    next ($properties);
}

```

ifx_num_fields

ifx_num_fields gibt die Anzahl der Spalten einer Ergebnisliste zurück.

ifx_num_fields

```
int ifx_num_fields(int result_id);
```



Die Funktion kann nach dem Vorbereiten oder Ausführen einer SQL-Abfrage genutzt werden.

ifx_num_rows

ifx_num_rows gibt die Anzahl der Datensätze an, die in der Ergebnisliste stehen.

ifx_num_rows

```
int ifx_num_rows(int result_id);
```



Nach der Ausführung von ifx_query oder ifx_do kann die Anzahl der bearbeiteten Datensätze ermittelt werden.

ifx_free_result

ifx_free_result gibt den von Ergebnislisten benötigten Speicher frei.

ifx_free_result

```
int ifx_free_result(int result_id);
```



ifx_create_char

ifx_create_char erzeugt ein Zeichen-Objekt.

ifx_create_char

```
int ifx_create_char(string param);
```



Das Argument enthält die Zeichenkette, die dem Objekt zugewiesen werden soll. Die Funktion gibt die ID des Objekts zurück.

ifx_free_char

ifx_free_char löscht ein Zeichen-Objekt.

ifx_free_char

```
int ifx_free_char(int bid);
```



Das Argument ist die ID des Zeichenobjekts.

ifx_update_char

ifx_update_char ifx_update_char erneuert den Inhalt eines Zeichen-Objekts.

- ◆ `int ifx_update_char(int bid, string content);`
Siehe auch ifx_free_char, ifx_create_char.

ifx_get_char

ifx_get_char ifx_get_char gibt den Inhalt eines Zeichen-Objekts zurück.

- ◆ `int ifx_get_char(int bid);`

ifx_create_blob

ifx_create_blob ifx_create_blob erzeugt ein BLOB-Objekt.

- ◆ `int ifx_create_blob(int type, int mode, string param);`
Erzeugt ein BLOB-Objekt vom Type *type* im Modus *mode* und mit den zusätzlichen Parametern *param*:

- *type*:
 - 1 = TEXT
 - 0 = BYTE
- *mode*:
 - 0 = BLOB-Objekt wird im Speicher gehalten
 - 1 = BLOB-Objekt wird als Datei abgelegt
- *param*:
 - Wenn *mode* 0 ist: Zeiger auf den Inhalt
 - Wenn *mode* 1 ist: Zeiger auf die Datei

ifx_copy_blob

ifx_copy_blob ifx_copy_blob verdoppelt das BLOB-Objekt.

- ◆ `int ifx_copy_blob(int bid);`

ifx_free_blob

ifx_free_blob ifx_free_blob löscht das BLOB-Objekt.

- ◆ `int ifx_free_blob(int bid);`

ifx_get_blob

ifx_get_blob gibt den Inhalt eines BLOB-Objekts zurück.

```
string int ifx_get_blob(int bid);
```

ifx_get_blob**ifx_update_blob**

ifx_update_blob erneuert den Inhalt eines BLOB-Objekts.

```
void ifx_update_blob(int bid, string content);
```

ifx_update_blob**ifx_blobinfile_mode**

ifx_blobinfile_mode setzt einen Standardmodus für alle Abfragen, die BLOB-Objekte beinhalten.

```
void ifx_blobinfile_mode(int mode);
```

ifx_blobinfile_mode

Mode kann folgendermaßen angegeben werden:

- "0" hält BLOB-Objekte im Speicher
- "1" speichert BLOB-Objekt als Datei

ifx_textasvarchar

ifx_textasvarchar setzt den Standardmodus für BLOB-Objekte vom Typ »Text«.

```
void ifx_textasvarchar(int mode);
```

ifx_textasvarchar

mode kann folgendermaßen angegeben werden:

- "0" hält BLOB-Objekte im Speicher.
- "1" speichert BLOB-Objekt als Datei.

ifx_byteasvarchar

ifx_byteasvarchar setzt den Standardmodus für BLOB-Objekte vom Typ »Byte«.

```
void ifx_byteasvarchar(int mode);
```

ifx_byteasvarchar

mode kann folgendermaßen angegeben werden:

- "0" hält BLOB-Objekte im Speicher.
- "1" speichert BLOB-Objekt als Datei.

ifx_nullformat

ifx_nullformat ifx_nullformat setzt den Standardwert für die Datensatzabfrage.



```
void ifx_nullformat(int mode);
```

mode kann folgendermaßen angegeben werden:

- "0" gibt für NULL-Werte eine leere Zeichenkette zurück.
- "1" gibt NULL zurück.

ifxus_create_slob

ifxus_create_slob ifxus_create_slob erzeugt ein SLOB-Objekt und öffnet es.



```
int ifxus_create_slob(int mode);
```

mode kann folgende Werte annehmen:

- 1 = LO_RDONLY
- 2 = LO_WRONLY
- 4 = LO_APPEND
- 8 = LO_RDWR
- 16 = LO_BUFFER
- 32 = LO_NOBUFFER

ifx_free_slob

ifx_free_slob ifx_free_slob gibt ein SLOB-Objekt frei.



```
int ifxus_free_slob(int bid);
```

ifxus_close_slob

ifxus_close_slob ifxus_close_slob löscht ein SLOB-Objekt.



```
int ifxus_close_slob(int bid);
```

ifxus_open_slob

ifxus_open_slob ifxus_open_slob öffnet ein SLOB-Objekt.



```
int ifxus_open_slob(long bid, int mode);
```

Mode kann folgende Werte annehmen:

- 1 = LO_RDONLY

- 2 = LO_WRONLY
- 4 = LO_APPEND
- 8 = LO_RDWR
- 16 = LO_BUFFER
- 32 = LO_NOBUFFER

ifxus_tell_slob

ifxus_tell_slob gibt die aktuelle Dateiposition zurück.

```
int ifxus_tell_slob(long bid);
```

ifxus_tell_slob



ifxus_seek_slob

ifxus_seek_slob setzt die Position des Zeigers in der Datei.

```
int ifxus_seek_slob(long bid, int mode, long offset);
```

ifxus_seek_slob



Mode kann folgende Werte annehmen:

- 1 = LO_RDONLY
- 2 = LO_WRONLY
- 4 = LO_APPEND
- 8 = LO_RDWR
- 16 = LO_BUFFER
- 32 = LO_NOBUFFER

ifxus_read_slob

ifxus_read_slob liest Bytes aus einem SLOB-Objekt.

```
int ifxus_read_slob(long bid, long nbytes);
```

Die Funktion liest *nbytes* Bytes aus dem SLOB-Objekt.

ifxus_read_slob



ifxus_write_slob

ifxus_write_slob schreibt eine Zeichenkette in ein SLOB-Objekt.

```
int ifxus_write_slob(long bid, string content);
```

ifxus_write_slob



Der Inhalt der Zeichenkette *content* wird in das SLOB-Objekt geschrieben, das durch *bid* ausgewählt wird.

MS-SQL-Funktionen

mssql_affected_rows

mssql_affected_rows

Die Funktion `mssql_affected_rows` gibt die Anzahl der Datensätze zurück, die von der letzten Aktion betroffen waren.



```
int mssql_affected_rows(int [link_identifizier]);
```

Die SQL-Befehle `INSERT`, `UPDATE` oder `DELETE` können unterschiedliche Auswirkungen haben. Wie viele Datensätze tatsächlich bearbeitet wurden, kann mit dieser Funktion ermittelt werden. Diese Funktion kann nicht nach `SELECT` eingesetzt werden, da `SELECT` keine Daten verändert. Wenn Sie die Anzahl der Datensätze ermitteln möchten, die `SELECT` zurückgibt, verwenden Sie die Funktion `mssql_num_rows`.

mssql_close

mssql_close

Schließt eine Datenbank.



```
int mssql_close(int [link_identifizier]);
```

Die Funktion gibt `TRUE` zurück, wenn der Vorgang erfolgreich war, sonst `FALSE`. Wird der *link_identifizier* nicht angegeben, wird die zuletzt geöffnete Verbindung geschlossen. Wenn nicht-persistente Verbindungen genutzt werden, werden alle Datenbankverbindungen am Ende des Skripts automatisch geschlossen, der Aufruf von `mssql_close` ist dann nicht notwendig. Persistente Verbindungen können mit dieser Funktion nicht geschlossen werden.

Siehe auch `mssql_connect`, `mssql_pconnect`.

mssql_connect

mssql_connect

`mssql_connect` öffnet eine Datenbankverbindung.



```
int mssql_connect(string servername,  
                 string username,  
                 string password);
```

Die Funktion gibt ein Datenbankhandle zurück (*link_identifizier*) oder `FALSE`, wenn der Vorgang nicht beendet werden konnte. Wenn die Funktion mehrfach aufgerufen wird, wird erneut das Datenbankhandle zurückgegeben, aber keine weitere Verbindung geöffnet. Verbindungen werden automatisch geschlossen, wenn das Skript beendet ist. Innerhalb des Skripts werden Verbindungen mit `mssql_close` geschlossen.

Siehe auch: `mssql_pconnect`, `mssql_close`.

mssql_data_seek

Die Funktion `mssql_data_seek` bewegt den internen Datenbankcursor.

mssql_data_seek

```
int mssql_data_seek(int result_identifizier,  
                   int row_number);
```



Die Funktion gibt im Erfolgsfall `TRUE` zurück, sonst `FALSE`. Nach dem der Datensatzcursor gesetzt wurde, gibt die Funktion `mssql_fetch_row` diesen Datensatz zurück.

mssql_fetch_array

Die Funktion `mssql_fetch_array` liest einen Datensatz komplett aus und schreibt die Daten in ein Array.

mssql_fetch_array

```
int mssql_fetch_array(int result);
```



Wenn keine Datensätze mehr gelesen werden können, wird `FALSE` zurückgegeben. Die Funktion gibt sowohl ein indiziertes Array mit Feldnummern als auch ein assoziatives Array mit Feldnamen als Schlüssel zurück. Die Funktionalität ist gegenüber `mssql_fetch_row` eingeschränkt, deshalb ist die Funktion `mssql_fetch_array` bei einer großen Anzahl Datensätze schneller.

Siehe auch: `mssql_fetch_row`.

mssql_fetch_field

`mssql_fetch_field` ermittelt Informationen zu Feldern.

mssql_fetch_field

```
object mssql_fetch_field(int result,  
                        int field_offset);
```



`mssql_fetch_field` ermittelt Daten aus einzelnen Feldern. Wenn der Offset `field_offset` nicht angegeben wurde, wird das nächste ungelesene Feld ermittelt.

Die Eigenschaften des zurückgegebenen Objekts sind:

- *name*. Spaltenname. Basiert die Spalte auf einer Funktion, wird der Spaltenname als `computed#N` angegeben, wobei `N` eine fortlaufende Nummer ist.
- *column_source*. Datenquelle, normalerweise der Name der Tabelle, aus der die Daten stammen.

- *max_length*. Maximale Länge der Spalte.
- *Numeric*. Diese Eigenschaft ist 1, wenn die Spalte numerisch ist.

Siehe auch `mssql_field_seek`.

mssql_fetch_object

mssql_fetch_object

Die Funktion `mssql_fetch_object` holt einen Datensatz und legt ihn als Objekt ab.



```
int mssql_fetch_object(int result);
```

Die Funktion entspricht `mssql_fetch_array`, allerdings wird ein Objekt und kein Array zurückgegeben. Die Möglichkeit, Felder über numerische Indizes anzusprechen, besteht nicht.

Siehe auch `mssql_fetch_array`, `mssql_fetch_row`.

mssql_fetch_row

mssql_fetch_row

Die Funktion `mssql_fetch_row` holt einen Datensatz als indiziertes Array.



```
array mssql_fetch_row(int result);
```

Die Funktion gibt `FALSE` zurück, wenn keine Daten mehr gelesen werden können. Jede Spalte ist ein Element des Arrays, beginnend mit dem Index 0. Aufeinanderfolgende Aufrufe der Funktion lesen jeweils den darauffolgenden Datensatz, der Datensatzcursor wird automatisch weiter bewegt.

Siehe auch: `mssql_fetch_array`, `mssql_fetch_object`, `mssql_data_seek`, `mssql_fetch_lengths` und `mssql_result`.

mssql_field_seek

mssql_field_seek

`mssql_field_seek` setzt den Feldoffset.



```
int mssql_field_seek(int result, int field_offset);
```

Setzt den Datensatzcursor auf das spezifizierte Feld.

Siehe auch `mssql_fetch_field`.

mssql_free_result

mssql_free_result gibt den Speicher frei, der für die Ablage der Ergebnislisten verbraucht wurde. **mssql_free_result**

```
int mssql_free_result(int result);
```



Der Speicher wird am Ende eines Skripts automatisch gelöscht. Der Aufruf ist nur notwendig, wenn es bei sehr großen Skripten zu Speicherproblemen kommt.

mssql_num_fields

mssql_num_fields gibt die Anzahl der Felder in einer Ergebnisliste zurück. **mssql_num_fields**

```
int mssql_num_fields(int result);
```



Siehe auch: mssql_db_query, mssql_query, mssql_fetch_field, mssql_num_rows.

mssql_num_rows

mssql_num_rows gibt die Anzahl der Datensätze der letzten Ergebnisliste zurück. **mssql_num_rows**

```
int mssql_num_rows(string result);
```



Siehe auch mssql_db_query, mssql_query und mssql_fetch_row.

mssql_pconnect

Die Funktion mssql_pconnect öffnet eine persistente (ständige) Datenbankverbindung. **mssql_pconnect**

```
int mssql_pconnect(string servername,  
                  string username,  
                  string password);
```



Diese Funktion gibt FALSE zurück, wenn die Anforderung nicht erfolgreich war, sonst wird das Datenbankhandle zurückgegeben. Die Funktion arbeitet wie mssql_connect, mit zwei Unterschieden: Wenn eine Verbindung zu einer Datenbank erfolgt, wird zuerst eine passende persistente Verbindung ermittelt; wenn diese vorhanden ist, wird sie auch verwendet. Schlägt dieser Versuch fehl, wird eine normale Verbindung geöffnet. Persistente Verbindungen können nicht geschlossen werden.

mssql_query

mssql_query mssql_query sendet eine SQL-Abfrage an den SQL Server.

◆
`int mssql_query(string query,
int link_identifizier);`

Die Funktion gibt ein Handle für die Ergebnisliste zurück. Wenn die Ausführung misslingt, wird FALSE zurückgegeben.

Wird der Parameter *link_identifizier* nicht angegeben, wird die zuletzt geöffnete Verbindung genutzt.

Siehe auch `mssql_db_query`, `mssql_select_db` und `mssql_connect`.

mssql_result

mssql_result mssql_result holt ein Feld aus einer Ergebnisliste.

◆
`int mssql_result(int result, int i, mixed field);`

Der Parameter *field* kann der Feldindex, der Feldname oder die komplette Bezeichnung *tabelle.name* sein. Wenn in einer Abfrage ein Alias genutzt wird, muss hier der Alias auch verwendet werden.

Es ist schneller, den gesamten Datensatz zu lesen, wenn mehr als ein Feld benötigt wird. Außerdem ist die Verwendung eines numerischen Index schneller als die eines Feldnamens. Schneller sind generell die folgenden Funktionen:

- `mssql_fetch_row`, `mssql_fetch_array` und `mssql_fetch_object`.

mssql_select_db

mssql_select_db Mit der Funktion `mssql_select_db` wird eine Datenbank ausgewählt.

◆
`int mssql_select_db(string database_name,
int link_identifizier);`

Im Erfolgsfall wird TRUE zurückgegeben. Wird der Parameter *link_identifizier* nicht angegeben, wird die zuletzt geöffnete Verbindung genutzt. Die Verwendung entspricht dem SQL-Befehl USE.

Siehe auch `mssql_connect`, `mssql_pconnect` und `mssql_query`.

mSQL-Funktionen

msql

Die Funktion `msql` sendet eine SQL-Anweisung an die Datenbank.

```
int msql(string database, string query, int link_identifizier);
```

Wenn die Anweisung misslingt, wird `FALSE` zurückgegeben, ansonsten ein Handle auf die Ergebnisliste. Der Parameter *link_identifizier* ist optional; wenn er nicht angegeben wird, öffnet PHP die letzte oder aktuelle Verbindung.

msql



msql_affected_rows

`msql_affected_rows` gibt die Anzahl der von der letzten Anweisung behandelten Datensätze zurück.

```
int msql_affected_rows(int query_identifizier);
```

Die Funktion arbeitet mit allen SQL-Befehlen.

Siehe auch `msql_query`.

msql_affected_rows



msql_close

`msql_close` schließt eine mSQL-Verbindung.

```
int msql_close(int link_identifizier);
```

Wird der Parameter nicht angegeben, wird die letzte Verbindung geschlossen. Persistente Verbindungen werden nicht geschlossen.

Siehe auch `msql_connect` und `msql_pconnect`.

msql_close



msql_connect

`msql_connect` öffnet eine mSQL-Verbindung.

```
int msql_connect(string hostname);
```

Gibt eine positive Nummer als Verbindungshandle zurück. Im Fehlerfall wird `FALSE` zurückgegeben. Wird kein Servername angegeben, wird *localhost* angenommen.

Offene Verbindungen werden am Ende eines Skripts automatisch geschlossen. Innerhalb eines Skripts wird zum Schließen `msql_close` eingesetzt.

Siehe auch `msql_pconnect`, `msql_close`.

msql_connect



mysql_create_db

mysql_create_db mysql_create_db erzeugt eine Datenbank.

◆ `int mysql_create_db(string database name,
int [link_identifizier]);`

Siehe auch mysql_drop_db.

mysql_createdb

mysql_createdb mysql_createdb erzeugt eine Datenbank.

◆ `int mysql_createdb(string database name,
int[link_identifizier]);`

Diese Funktion ist mit mysql_create_db identisch.

mysql_data_seek

mysql_data_seek mysql_data_seek bewegt den internen Datensatzcursor.

◆ `int mysql_data_seek(int query_identifizier,
int row_number);`

Der nächste Aufruf der Funktion mysql_fetch_row gibt den Datensatz zurück, der durch den Zeiger angesprochen wurde.

Siehe auch mysql_fetch_row.

mysql_dbname

mysql_dbname Die Funktion mysql_dbname gibt den Namen der aktuellen Datenbank zurück.

◆ `string mysql_dbname(int query_identifizier, int i);`

mysql_dbname gibt den Namen der Datenbank zurück, in Position *i* des Ergebnislistencursors der Funktion mysql_listdbs.

mysql_drop_db

mysql_drop_db mysql_drop_db löscht eine mSQL-Datenbank.

◆ `int mysql_drop_db(string database_name,
int link_identifizier);`

mysql_drop_db versucht, die gesamte Datenbank mit allen Daten zu entfernen.

Siehe auch mysql_create_db.

msql_dropdb

msql_dropdb löscht eine Datenbank.

msql_dropdb

Eine andere Schreibweise für msql_drop_db.

msql_error

msql_error gibt die Fehlermeldung der letzten Aktion zurück.

msql_error

```
string msql_error();
```



msql_fetch_array

msql_fetch_array holt einen Datensatz in ein Array.

msql_fetch_array

```
int msql_fetch_array(int query_identifizier);
```



Die Funktion überträgt die Daten in ein Array, das sowohl indiziert als auch assoziativ angesprochen werden kann. Die Schlüssel des assoziativen Arrays sind die Spaltennamen. Die Funktion ist nicht langsamer als msql_fetch_row.

Siehe auch msql_fetch_row.

msql_fetch_field

msql_fetch_field liest ein Feld aus der Ergebnisliste aus.

msql_fetch_field

```
object msql_fetch_field(int query_identifizier,  
                        int field_offset);
```



Die Funktion gibt ein Objekt mit den Feldinformationen zurück. Wird kein Offset *field_offset* angegeben, wird das nächste ungelesene Feld zurückgegeben.

Die Eigenschaften des Objekts sind:

- `name`. Spaltenname. Wenn die Spalte das Ergebnis einer Berechnung ist, wird als Name `computed#N` gebildet, wobei N eine fortlaufende Nummer ist.
- `column_source`. Name der Tabelle, die als Datenquelle diente.
- `max_length`. Maximale Länge (Breite) einer Spalte.
- `numeric`. Diese Eigenschaft ist 1, wenn es sich um einen numerischen Datentyp handelt.
- `not_null`. Ist 1, wenn NULL nicht erlaubt ist.

- `primary_key`. Ist 1, wenn die Spalte ein Primärschlüssel ist.
- `unique`. Ist 1, wenn die Spalte das Attribut UNIQUE trägt.
- `type`. Der Datentyp der Spalte.

mysql_fetch_object

mysql_fetch_object mysql_fetch_object holt einen Datensatz der angegebenen Ergebnisliste in ein Objekt.

◆ `int mysql_fetch_object(int query_identifizier);`

Die Funktion arbeitet wie `mysql_fetch_array`. Die Feldnamen sind die Namen der Eigenschaften. Der Zugriff über Indizes ist nicht möglich. Es gibt keine signifikanten Geschwindigkeitsunterschiede zu den Funktionen `mysql_fetch_array` und `mysql_fetch_row`.

Siehe auch `mysql_fetch_array` und `mysql_fetch_row`.

mysql_fetch_row

mysql_fetch_row mysql_fetch_row liest einen Datensatz in ein indiziertes Array.

◆ `array mysql_fetch_row(int query_identifizier);`

Die Funktion verwendet nur ein indiziertes Array. Wenn keine Daten zu holen sind, wird FALSE zurückgegeben. Die Funktion setzt außerdem den internen Datensatzzeiger weiter.

mysql_fieldname

mysql_fieldname mysql_fieldname gibt den Namen eines Felds zurück.

◆ `string mysql_fieldname(int query_identifizier, int field);`

Das Feld wird durch seinen Index qualifiziert.

mysql_field_seek

mysql_field_seek mysql_field_seek setzt den Datensatzcursor.

◆ `int mysql_field_seek(int query_identifizier,
int field_offset);`

Der nächste Aufruf der Funktion `mysql_fetch_field` liest den ausgewählten Datensatz.

Siehe auch `mysql_fetch_field`.

`msql_fieldtable`

`msql_fieldtable` ermittelt den Namen der verwendeten Tabelle zu einem Feld.

```
int msql_fieldtable(int query_identifizier, int field);
```

Gibt den Namen der Tabelle zurück, aus der das bezeichnete Feld stammt.

`msql_fieldtable`

`msql_fieldtype`

`msql_fieldtype` gibt den Feldtyp zurück.

```
string msql_fieldtype(int query_identifizier, int i);
```

Der Datentyp kann »int«, »string« oder »real« sein.

`msql_fieldtype`

`msql_fieldflags`

`msql_fieldflags` gibt die Feldeigenschaften zurück.

```
string msql_fieldflags(int query_identifizier, int i);
```

Die Feldeigenschaften können »not null«, »primary key«, beides oder eine leere Zeichenkette "" sein.

`msql_fieldflags`

`msql_fieldlen`

`msql_fieldlen` gibt die Länge eines Felds zurück.

```
int msql_fieldlen(int query_identifizier, int i);
```

`msql_fieldlen` returns the length of the specified field.

`msql_fieldlen`

`msql_free_result`

`msql_free_result` gibt den Speicher frei, der durch die aktuelle Ergebnisliste belegt wurde.

```
int msql_free_result(int query_identifizier);
```

Die Anwendung ist normalerweise nicht notwendig, da der Speicher am Ende eines Skripts automatisch freigegeben wird.

`msql_free_result`

`msql_freeresult`

Siehe `msql_free_result`.

`msql_freeresult`

mysql_list_fields

mysql_list_fields mysql_list_fields gibt Informationen über Felder zurück.

◆ `int mysql_list_fields(string database, string tablename);`

Die Funktion gibt ein Handle über eine Tabelle zurück. Das Handle kann mit den Funktionen `mysql_fieldflags`, `mysql_fieldlen`, `mysql_fieldname` und `mysql_fieldtype` genutzt werden.

-1 wird zurückgegeben, wenn ein Fehler auftritt, außerdem wird die Fehlermeldung in `$phperrmsg` abgelegt.

mysql_listfields

mysql_listfields mysql_listfields gibt Informationen über Felder zurück.

Siehe `mysql_list_fields`.

mysql_list_dbs

mysql_list_dbs mysql_list_dbs zeigt auf alle Datenbanken.

◆ `int mysql_list_dbs(void);`

`mysql_list_dbs` gibt ein Handle zurück, der auf alle Datenbanken zeigt. Mit Hilfe der Funktion `mysql_dbname` und dem Handle kann der Name der Datenbanken ermittelt werden.

mysql_listdbs

mysql_listdbs mysql_listdbs zeigt auf alle Datenbanken.

Siehe `mysql_list_dbs`.

mysql_list_tables

mysql_list_tables mysql_list_tables zeigt Tabelleninformationen an.

◆ `int mysql_list_tables(string database);`

`mysql_list_tables` gibt ein Handle zurück, aus dem mit Hilfe der Funktion `mysql_tablename` der Name der Tabelle ermittelt werden kann.

mysql_listtables

mysql_listtables mysql_listtables zeigt Tabelleninformationen an.

Siehe `msql_list_tables`.

msql_num_fields

`msql_num_fields` gibt die Anzahl der Felder in der Ergebnisliste zurück.

```
int msql_num_fields(int query_identifizier);
```



msql_num_fields

msql_num_rows

`msql_num_rows` gibt die Anzahl der Reihen in einer Ergebnisliste zurück.

```
int msql_num_rows(int query_identifizier);
```



msql_num_rows

msql_numfields

Siehe `msql_num_fields`.

msql_numfields

msql_numrows

Siehe `msql_num_rows`.

msql_numrows

msql_pconnect

`msql_pconnect` öffnet eine persistente (ständige) Verbindung zu einer Datenbank.

```
int msql_pconnect(string hostname);
```



msql_pconnect

Im Fehlerfall wird FALSE zurückgegeben, sonst das Verbindungshandle.

`msql_pconnect` arbeitet ähnlich wie `msql_connect`.

Die Verbindung wird nicht geschlossen, wenn das Skript endet. Wenn in einem anderen Skript eine Verbindung mit denselben Daten angefordert wird, versucht die Funktion zuerst eine persistente Verbindung wiederzubeleben, bevor eine neue geöffnet wird. Verbindungen dieser Funktion können nicht mit `msql_close` geschlossen werden.

msql_query

`msql_query` sendet eine SQL-Anweisung an den Server.

```
int msql_query(string query, int link_identifizier);
```



msql_query

Der Parameter *link_identifier* ist optional, ohne Angabe wird die aktuelle oder zuletzt geöffnete Verbindung genutzt.

Siehe auch `mysql`, `mysql_select_db`, `mysql_connect`.

mysql_regcase

mysql_regcase `mysql_regcase` testet einen regulären Ausdruck

◆ Siehe `sql_regcase`.

mysql_result

mysql_result `mysql_result` holt die Ergebnisse einer Abfrage in die Ergebnisliste.

◆

```
int mysql_result(int query_identifier,
                 int i,
                 mixed field);
```

Der Parameter *field* kann der Feldindex, der Feldname oder die komplette Bezeichnung *tabelle.name* sein. Wenn in einer Abfrage ein Alias genutzt wird, muss hier der Alias auch verwendet werden.

Es ist schneller, den gesamten Datensatz zu lesen, wenn mehr als ein Feld benötigt wird. Außerdem ist die Verwendung eines numerischen Index schneller als die eines Feldnamens. Schneller sind generell die Funktionen `mysql_fetch_row`, `mysql_fetch_array` und `mysql_fetch_object`.

mysql_select_db

mysql_select_db `mysql_select_db` wählt eine Datenbank aus.

◆

```
int mysql_select_db(string database_name,
                   int link_identifier);
```

Im Erfolgsfall wird `TRUE` zurückgegeben. Wird der Parameter *link_identifier* nicht angegeben, wird die zuletzt geöffnete Verbindung genutzt. Die Verwendung entspricht dem SQL-Befehl `USE`.

Siehe auch `mysql_connect`, `mysql_pconnect` und `mysql_query`.

mysql_selectdb

mysql_selectdb Siehe `mysql_select_db`.

mysql_tablename

mysql_tablename `mysql_tablename` ermittelt den Namen einer Tabelle zu einem Feld.

```
string msql_tablename(int query_Identifier,
                    int field);

<?php
msql_connect ("localhost");
$result = msql_list_tables("wisconsin");
$i = 0;
while ($i < msql_numrows($result)) {
    $tb_names[$i] = msql_tablename($result, $i);
    echo $tb_names[$i] . "<BR>";
    $i++;
}
?>
```



Beispiel

MySQL-Funktionen

MySQL wurde aufgrund der herausragenden Bedeutung in Kapitel 6 und 7 ausführlich beschrieben. Mehr Informationen sind außerdem unter <http://www.mysql.com> zu finden.

mysql_affected_rows

mysql_affected_rows gibt die Anzahl der von der letzten Operation betroffenen Reihen zurück.

mysql_affected_rows

```
int mysql_affected_rows(int [link_Identifier]);
```



Diese Funktion arbeitet nur für INSERT, UPDATE oder DELETE. Bei DELETE wird die WHERE-Bedingung zwar ausgeführt, die Funktion gibt aber 0 zurück.

Siehe auch mysql_num_rows.

mysql_close

mysql_close schließt eine MySQL-Verbindung.

mysql_close

```
int mysql_close(int [link_Identifier]);
```



Die Verbindung wird am Ende eines Skripts automatisch geschlossen.

Siehe auch mysql_connect und mysql_pconnect.

mysql_connect

mysql_connect öffnet eine MySQL-Verbindung.

mysql_connect

```
int mysql_connect(string [hostname] [:port]
                [, string username] [, string password] );
```



Siehe auch `mysql_pconnect` und `mysql_close`.

mysql_create_db

mysql_create_db `mysql_create_db` erzeugt eine neue Datenbank.

◆ `int mysql_create_db(string database name,
int [link_identifizier]);`

mysql_data_seek

mysql_data_seek `mysql_data_seek` bewegt den Zeiger auf die Ergebnisliste.

◆ `int mysql_data_seek(int result_identifizier, int row_number);`

mysql_db_query

mysql_db_query `mysql_db_query` sendet eine SQL-Abfrage an die Datenbank.

◆ `int mysql_db_query(string database, string query,
int [link_identifizier]);`

Die Funktion gibt einen von anderen Funktionen verwendetem Zeiger auf die Ergebnisliste der Abfrage zurück.

mysql_drop_db

mysql_drop_db `mysql_drop_db` löscht eine Datenbank.

◆ `int mysql_drop_db(string database_name, int [link_identifizier]);`

Die Funktion löscht die Datenbank ohne Rücksicht darauf, ob Daten enthalten sind oder nicht.

Siehe auch `mysql_create_db`.

mysql_errno

mysql_errno `mysql_errno` gibt die letzte Fehlernummer aus.

◆ `int mysql_errno(int [link_identifizier]);`

In PHP selbst werden im Fehlerfall keine Warnungen erzeugt. Eintypischer Einsatz ist die folgende Abfragefunktion:



```
<?php
function myquery($query, $conn, $debug = FALSE) {
    if ($debug) echo "<b>SQL: </b>";
    $result = @mysql_query($query, $conn) or
        die("<b>Fehler-Nr. : </b>: " . mysql_errno($conn) .
            "<br><b>Meldung: </b>: " . mysql_error($conn));
```

```
    return $result;
?>
```

Siehe auch `mysql_error`.

mysql_error

`mysql_error` gibt den letzten Fehlertext aus.

```
string mysql_error(int [link_identifizier] );
```

Siehe auch `mysql_errno`.

mysql_error



mysql_fetch_array

`mysql_fetch_array` holt einen Datensatz der Ergebnisliste in ein assoziatives Array.

```
array mysql_fetch_array(int result,int [result_typ]);
```

`mysql_fetch_array` ist eine erweiterte Version von `mysql_fetch_row`. Haben Spalten keine eindeutigen Namen, müssen in der Abfrage Aliase benutzt werden:

```
SELECT t1.f1 AS foo t2.f1 AS bar FROM t1, t2
```

Der Parameter *result_type* kann Folgendes sein:

- `MYSQL_ASSOC`. Erzeugt ein assoziatives Array.
- `MYSQL_NUM`. Erzeugt ein numerisches Array.
- `MYSQL_BOTH`. Erzeugt beides.

```
<?php
mysql_connect($host, $user, $password);
$result = mysql_db_query("mydbase","SELECT * FROM cust_table");
while($arrRow = mysql_fetch_array($result)) {
    echo $arrRow["custID"];
    echo $arrRow["name"];
}
mysql_free_result($result);
?>
```



Beispiel

mysql_fetch_field

`mysql_fetch_field` holt Informationen über eine Spalte der Ergebnisliste in ein Objekt.

```
object mysql_fetch_field(int result, int [field_offset] );
```

Die Eigenschaften des Objekts sind:

mysql_fetch_field



- name – Spaltenname.
- table – Tabellename, dem das Feld zugehört.
- max_length – Maximale Länge.
- not_null – 1, wenn NULL nicht erlaubt ist.
- primary_key – 1, wenn die Spalte einen Primärschlüssel hat.
- unique_key – 1, wenn die Spalte UNIQUE ist.
- multiple_key – 1, wenn die Spalte nicht UNIQUE ist.
- numeric – 1, wenn die Spalte einen numerischen Datentyp hat.
- blob – 1, wenn die Spalte ein BLOB-Typ ist.
- type – der Datentyp
- unsigned – 1, wenn die Spalte kein Vorzeichen haben darf.
- zerofill – 1, wenn standardmäßig 0 eingetragen wird.

Siehe auch `mysql_field_seek`.

mysql_fetch_lengths

mysql_fetch_lengths

`mysql_fetch_lengths` gibt den maximal benötigten Platz für die Ergebnisse in Form eines Arrays zurück.



```
array mysql_fetch_lengths(int result);
```

Das Array wird mit 0 beginnend indiziert.

Siehe auch `mysql_fetch_row`.

mysql_fetch_object

mysql_fetch_object

`mysql_fetch_object` holt einen Datensatz aus der Ergebnisliste in ein Objekt.



```
object mysql_fetch_object(int result, int [result_typ]);
```

Es gibt keinen signifikanten Geschwindigkeitsunterschied zu der Funktion `mysql_fetch_array`.



```
<?php
mysql_connect($host, $user, $password);
$result = mysql_db_query("mydbase","SELECT * FROM cust_table");
while($objRow = mysql_fetch_object($result)) {
    echo $objRow->custID;
    echo $objRow->name;
}
```

```
}  
mysql_free_result($result);  
?>
```

Siehe auch `mysql_fetch_array` und `mysql_fetch_row`.

mysql_fetch_row

`mysql_fetch_row` holt einen Datensatz aus der Ergebnisliste als ein numerisches Array.

```
array mysql_fetch_row(int result);
```

Der Index des Arrays beginnt mit 0.

Siehe auch: `mysql_fetch_array`, `mysql_fetch_object`, `mysql_data_seek`, `mysql_fetch_lengths` und `mysql_result`.

mysql_field_name

`mysql_field_name` ermittelt den Namen eines Feldes.

```
string mysql_field_name(int result,int field_index);
```

Das Argument *field_index* ist der Index des Felds.

```
mysql_field_name($result,2);
```

mysql_field_seek

`mysql_field_seek` setzt den Zeiger auf die Ergebnisliste.

```
int mysql_field_seek(int result, int field_offset);
```

Hiermit wird der nächste von `mysql_fetch_field` geholte Datensatz adressiert.

Siehe auch `mysql_fetch_field`.

mysql_field_table

`mysql_field_table` ermittelt den Namen einer Tabelle, in der das benannte Feld ist.

```
string mysql_field_table(int result, int field_offset);
```

mysql_field_type

`mysql_field_type` ermittelt den Datentyp einer Spalte.

```
string mysql_field_type(int result,int field_offset);
```

mysql_fetch_row



mysql_field_name



mysql_field_seek



mysql_field_table



mysql_field_type



Die Rückgabewerte können eine der folgenden Zeichenketten sein (Auswahl):

- "int"
- "real"
- "string"
- "blob"



Beispiel

```
<?php
mysql_connect("localhost:3306");
mysql_select_db("customers");
$result = mysql_query("SELECT * FROM cust_table");
$fields = mysql_num_fields($result);
$rows = mysql_num_rows($result);
$i = 0;
$table = mysql_field_table($result, $i);
echo "Die Tabelle '". $table. "' hat ". $fields.
    " Felder und ". $rows. " Datensätze <br>";
echo "Es folgt eine Liste der Felder:<br>";
while ($i < $fields) {
    $strType = mysql_field_type ($result, $i);
    $strName = mysql_field_name ($result, $i);
    $intLen = mysql_field_len ($result, $i);
    $strFlags = mysql_field_flags ($result, $i);
    echo "$strType &nbsp; $strName &nbsp; ";
    echo "$intLen &nbsp; $strFlags <br>";
    $i++;
}
mysql_close();
?>
```

mysql_field_flags

mysql_field_flags mysql_field_flags ermittelt die Optionen eines Felds.



```
string mysql_field_flags(int result, int field_offset);
```

Der Rückgabewert kann eine oder mehrere der folgenden Zeichenketten beinhalten:

- "not_null"
- "primary_key"
- "unique_key"
- "multiple_key"
- "blob"

- "unsigned"
- "zerofill"
- "binary"
- "enum"
- "auto_increment"
- "timestamp"

mysql_field_len

mysql_field_len ermittelt die Länge eines Felds.

```
int mysql_field_len(int result,intfield_offset);
```

mysql_field_len



mysql_free_result

mysql_free_result gibt Speicher für Ergebnislisten wieder frei.

```
int mysql_free_result(int result);
```

mysql_free_result



Der Speicher wird am Ende eines Skripts immer freigegeben, die Anwendung macht also nur innerhalb eines Skripts Sinn.

mysql_insert_id

mysql_insert_id Gibt die ID zurück, welche die letzte INSERT-Anweisung erzeugt hat.

```
int mysql_insert_id(int [link_identifizier]);
```

mysql_insert_id



Der Wert kann nur ermittelt werden, wenn eine Spalte mit der Eigenschaft AUTO_INCREMENT vorhanden ist.

mysql_list_fields

mysql_list_fields zeigt Daten der Felder der Tabelle an.

```
int mysql_list_fields(string database_name,  
                    string table_name,  
                    int [link_identifizier]);
```

mysql_list_fields



Die Funktion gibt nur einen Zeiger auf die Ergebnisse der Abfrage zurück. Die Auswertung kann mit den Funktionen mysql_field_flags, mysql_field_len, mysql_field_name und mysql_field_type erfolgen.

Fehler werden als -1 zurückgegeben und Fehlermeldungen in \$phperrmsg platziert und ausgegeben. Die Ausgabe kann mit dem @-Operator vor der Funktion unterdrückt werden.

mysql_list_dbs

mysql_list_dbs mysql_list_dbs zeigt alle Datenbanken an.

◆ `int mysql_list_dbs(int [link_identifizier]);`

Die Funktion gibt nur einen Zeiger auf die Ergebnisse der Abfrage zurück. Die Auswertung kann mit der Funktion mysql_tablename erfolgen.

mysql_list_tables

mysql_list_tables mysql_list_tables zeigt alle Tabellen einer Datenbank an.

◆ `int mysql_list_tables(string database, int [link_identifizier]);`

Die Funktion gibt nur einen Zeiger auf die Ergebnisse der Abfrage zurück. Die Auswertung kann mit mysql_tablename erfolgen.

mysql_num_fields

mysql_num_fields mysql_num_fields ergibt die Anzahl der Felder der Ergebnisliste.

◆ `int mysql_num_fields(int result);`

Siehe auch mysql_db_query, mysql_query, mysql_fetch_field, mysql_num_rows.

mysql_num_rows

mysql_num_rows mysql_num_rows Anzahl der Datensätze in der Ergebnisliste.

◆ `int mysql_num_rows(int result);`

Siehe auch mysql_db_query, mysql_query und mysql_fetch_row.

mysql_pconnect

mysql_pconnect mysql_pconnect öffnet eine persistente Verbindung.

◆ `int mysql_pconnect(string [hostname] [:port] ,
string [username], string [password]);`

Persistente Verbindungen sind nur möglich, wenn PHP als Modul im Webserver arbeitet, nicht als CGI. mysql_close hat keine Wirkung auf persistente Verbindungen. Die erneute Nutzung einer Verbindung erfolgt durch die Angabe identischer Parameter.

mysql_query

mysql_query sendet eine SQL-Abfrage an MySQL.

```
int mysql_query(string query,int[link_identifier]);
```

INSERT, UPDATE und DELETE geben TRUE zurück, wenn die Anweisung erfolgreich war. SELECT gibt einen Zeiger auf die Ergebnisliste zurück. Von dieser Liste bockierter Speicher kann mit mysql_free_result freigemacht werden.

Siehe auch mysql_db_query, mysql_select_db und mysql_connect.

mysql_query



mysql_result

mysql_result gibt den Inhalt genau eines Felds zurück.

```
int mysql_result(int result, int row, mixed field);
```

Bessere (schnellere) Alternativen sind die Funktionen mysql_fetch_row, mysql_fetch_array und mysql_fetch_object.

mysql_result



mysql_select_db

mysql_select_db wählt eine Datenbank als Standard aus.

```
int mysql_select_db(string database_name,  
int [link_identifier] );
```

Die Anwendung entspricht der Anweisung USE in SQL.

Siehe auch mysql_connect, mysql_pconnect und mysql_query.

mysql_select_db



mysql_tablename

mysql_tablename ermittelt den Namen einer Tabelle zu einem Feld *i*.

```
string mysql_tablename(int result, int i);
```

mysql_tablename benötigt als Parameter den Ergebniszeiger, den die Funktion mysql_list_tables zurückgibt. mysql_num_rows zeigt die Anzahl der Tabellen an.

mysql_tablename



```
<?php  
mysql_connect ("localhost");  
$result = mysql_listtables ("mydbase");  
$i = 0;  
while ($i < mysql_num_rows ($result)) {  
    $tb_names[$i] = mysql_tablename ($result, $i);  
    echo $tb_names[$i] . "<br>";  
    $i++;  
}
```



Beispiel

```
}
?>
```

ODBC-Funktionen

odbc_autocommit

odbc_autocommit odbc_autocommit schaltet die Behandlung der automatischen Bestätigung von Transaktionen um.



```
int odbc_autocommit(int connection_id, int [OnOff]);
```

Ohne den Parameter *OnOff* wird der aktuelle Auto-Commit-Status zurückgegeben (TRUE wenn *On*, FALSE wenn *Off*). Der Standardwert ist *On*. Das Abschalten entspricht dem Start einer neuen Transaktion. Normalerweise wird jede Anweisung als eine Transaktion angesehen und nach der erfolgreichen Abarbeitung bestätigt (commit).

Siehe auch odbc_commit und odbc_rollback.

odbc_binmode

odbc_binmode odbc_binmode behandelt Spalten mit binären Daten.



```
int odbc_binmode(int result_id, int mode);
```

Der Parameter *mode* bestimmt die betroffenen Datentypen (BINARY, VARBINARY, LONGVARBINARY):

- ODBC_BINMODE_PASSTHRU: BINARY-Daten unverändert weitergeben.
- ODBC_BINMODE_RETURN: Alles unverändert zurückgeben.
- ODBC_BINMODE_CONVERT: In Zeichenkette verwandeln.

Bei der Umwandlung in Zeichenketten wird jedes Byte (8 Bit) in zwei Bytes ASCII-Code verwandelt. Diese Zeichen sind die ASCII-Hex-Darstellung des Binärwertes der Nibble: 00000001 entspricht »01« und 11111111 entspricht »FF«.

Tabelle B.5:
LONGVARBINARY
Behandlung

<i>binmode</i>	<i>longreadlen</i>	Ergebnis
ODBC_BINMODE_PASSTHRU	0	Durchleiten
ODBC_BINMODE_RETURN	0	Durchleiten
ODBC_BINMODE_CONVERT	0	Durchleiten
ODBC_BINMODE_PASSTHRU	0	Durchleiten

<code>ODBC_BINMODE_PASSTHRU</code>	<code>>0</code>	Durchleiten
<code>ODBC_BINMODE_RETURN</code>	<code>>0</code>	Unverändert
<code>ODBC_BINMODE_CONVERT</code>	<code>>0</code>	Zeichen

Wenn `odbc_fetch_into` verwendet wird, bedeutet »durchleiten«, dass für leere Spalten eine leere Zeichenkette zurückgegeben wird. Wenn der Parameter `result_id` gleich 0 ist, gelten die Einstellungen ab sofort als Standardwerte.

Der Standardwert für `longreadlen` ist 4 096 und für `binmode` `ODBC_BINMODE_RETURN`. Die Behandlung von großen Binärspalten betrifft außerdem `odbc_longreadlen`.



odbc_close

`odbc_close` schließt eine Verbindung zu einem über ODBC angeschlossenen Datenbankserver.

```
void odbc_close(int connection_id);
```

Die Funktion schlägt fehl, wenn die Verbindung noch offene Transaktionen führt.

odbc_close



odbc_close_all

`odbc_close_all` schließt alle offenen Verbindungen.

```
void odbc_close_all(void);
```

Die Funktion schlägt fehl, wenn die Verbindung noch offene Transaktionen führt.

odbc_close_all



odbc_commit

`odbc_commit` bestätigt eine Transaktion.

```
int odbc_commit(int connection_id);
```

Wenn mehreren Transaktionen für die angegebene Verbindung offen waren, werden alle bestätigt. Die Funktion gibt im Erfolgsfall `TRUE` zurück, sonst `FALSE`.

odbc_commit



odbc_connect

`odbc_connect` öffnet die Verbindung zu einer Datenquelle.

```
int odbc_connect(string dsn, string user,  
                 string password, int [cursor_type]);
```

odbc_connect



Die Funktion gibt die von anderen Funktionen genutzte *connection_id* zurück. Es können mehrere Verbindungen parallel geöffnet werden. Der Parameter *cursor_type* ist optional. Folgende Angaben sind möglich:

- SQL_CUR_USE_IF_NEEDED
- SQL_CUR_USE_ODBC
- SQL_CUR_USE_DRIVER
- SQL_CUR_DEFAULT

SQL_CUR_USE_ODBC können Sie einsetzen, wenn verschiedene Fehlermeldungen auftreten, die sich auf Cursor beziehen, vor allem, wenn die verwendete Datenbank Cursor selbst nicht optimal unterstützt.

Für persistente Verbindungen siehe auch `odbc_pconnect`.

odbc_cursor

odbc_cursor `odbc_cursor` ermittelt den Namen eines Cursors für eine Ergebnisliste *result_id*.

◆ `string odbc_cursor(int result_id);`

odbc_do

odbc_do `odbc_do` ist ein Synonym für die Funktion `odbc_exec`.

◆ `string odbc_do(int conn_id, string query);`

odbc_exec

odbc_exec `odbc_exec` verarbeitet eine SQL-Anweisung und sendet sie zur Ausführung an den SQL-Server.

◆ `int odbc_exec(int connection_id, string query_string);`

Im Fehlerfall wird `FALSE` zurückgegeben, sonst die von anderen Funktionen benötigte *result_id* für die empfangene Ergebnisliste.

Siehe auch `odbc_prepare` und `odbc_execute` für die wiederholte Ausführung von SQL-Anweisungen.

odbc_execute

odbc_execute `odbc_execute` führt eine vorbereitete Anweisung aus.

◆ `int odbc_execute(int result_id, array [parameters_array]);`

Der Befehl führt eine mit `odbc_prepare` vorbereitete Anweisung aus. Im Fehlerfall wird `FALSE` zurückgegeben, sonst `TRUE`. Wurde die Anweisung mit Parametern vorbereitet, kann optional das Array `parameters_array` mit den gewünschten Werten angegeben werden.

`odbc_fetch_into`

`odbc_fetch_into` holt einen Datensatz aus der Ergebnisliste in ein nummeriertes Array.

`odbc_fetch_into`

```
int odbc_fetch_into(int result_id, int [rownumber],  
                    array result_array);
```



Die Funktion gibt die Anzahl der Spalten der Ergebnisliste zurück, das Array `result_array` selbst muss als Referenz übergeben werden (mit dem `&`-Operator). Der Index des Arrays beginnt immer mit 0.

`odbc_fetch_row`

`odbc_fetch_row` holt einen Datensatz.

`odbc_fetch_row`

```
int odbc_fetch_row(int result_id, int [row_number]);
```



Die Funktion gibt `TRUE` zurück, wenn noch Datensätze vorhanden sind, sonst `FALSE`. Die Funktion wird unmittelbar nach `odbc_do` oder `odbc_exec` ausgeführt und bestimmt, welche Daten von einem anschließenden Aufruf von `odbc_fetch_row` übernommen werden. Die Angabe der Reihe `row_number` ist optional. Jeder Aufruf setzt den Datensatzzeiger eine Reihe weiter. Um alle Datensätze zu holen, wird der Startwert für `row_number` auf 1 gesetzt.

`odbc_field_name`

`odbc_field_name` ermittelt den Spaltennamen für das Feld `field_number`.

`odbc_field_name`

```
string odbc_fieldname(int result_id, int field_number);
```



Die Feldindizes beginnen bei dieser Funktion mit 1. Bei einem Fehler wird `FALSE` zurückgegeben.

`odbc_field_type`

`odbc_field_type` ermittelt den Datentyp für das Feld `field_number`.

`odbc_field_type`

```
string odbc_field_type(int result_id, int field_number);
```



Die Feldindizes beginnen bei dieser Funktion mit 1. Bei einem Fehler wird `FALSE` zurückgegeben.

odbc_field_len

odbc_field_len odbc_field_len ermittelt die Länge in Bytes für das Feld *field_number*.

◆ `int odbc_field_len(int result_id, int field_number);`

Die Feldindizes beginnen bei dieser Funktion mit 1. Bei einem Fehler wird FALSE zurückgegeben.

odbc_free_result

odbc_free_result odbc_free_result gibt Speicher frei, der von der Ergebnisliste *result_id* benötigt wurde.

◆ `int odbc_free_result(int result_id);`

Die Funktion gibt immer TRUE zurück. Am Ende eines Skripts wird der Speicher immer geräumt. Der Aufruf ist also nur notwendig, wenn Speicher während der Laufzeit eines Skripts benötigt wird.

Wenn Auto-Commit ausgeschaltet ist und eine offene Transaktion vorhanden ist, wird diese zurückgesetzt (rollback), wenn Sie die Funktion odbc_free_result aufrufen.

odbc_longreadlen

odbc_longreadlen odbc_longreadlen behandelt Spalten mit LONG-Datentypen (Beitritt: LONG, LONGVARBINARY).

◆ `int odbc_longreadlen(int result_id, int length);`

Der Parameter *length* gibt an, wie viele Bytes behandelt werden sollen. Wenn er auf 0 gesetzt wird, werden die Daten unverändert in voller Länge weitergegeben (pass thru).

Achten Sie auf die mit odbc_binmode vorgenommenen Einstellungen!

odbc_num_fields

odbc_num_fields odbc_num_fields gibt die Anzahl der Spalten in einer Ergebnisliste *result_id* zurück.

◆ `int odbc_num_fields(int result_id);`

Fehler werden als -1 zurückgegeben.

odbc_pconnect

odbc_pconnect öffnet eine persistente Verbindung.

```
int odbc_pconnect(string dsn, string user,
                 string password, int [cursor_type]);
```

Die Funktion entspricht odbc_connect, eine Verbindung wird aber am Ende eines Skripts nicht geschlossen. Dies funktioniert nur, wenn PHP als Modul im Webserver arbeitet, nicht als CGI. Die Funktion erkennt die Verbindung in einem anderen Skript anhand der identischen Parameter wieder.

odbc_pconnect**odbc_prepare**

odbc_prepare bereitet eine SQL-Anweisung zur Ausführung mit odbc_execute vor.

```
int odbc_prepare(int connection_id,
                string query_string);
```

Im Fehlerfall wird FALSE zurückgegeben.

odbc_prepare**odbc_num_rows**

odbc_num_rows ergibt die Anzahl der Reihen in der Ergebnisliste *result_id*.

```
int odbc_num_rows(int result_id);
```

Die Funktion gibt die Anzahl der Reihen einer mit SELECT erzeugten Ergebnisliste zurück oder die Anzahl der bearbeiteten Reihen der letzten INSERT-, UPDATE- oder DELETE-Anweisung.

odbc_num_rows**odbc_result**

odbc_result gibt den Inhalt eines Felds der Ergebnisliste zurück.

```
string odbc_result(int result_id, mixed field);
```

field kann die Nummer oder der Name eines Felds sein.

```
$item_3 = odbc_result($Query_ID, 3 );
$item_val = odbc_result($Query_ID, "val");
```

Im Fehlerfall wird -1 zurückgegeben, beispielsweise wenn ein Feld nicht vorhanden ist. Die Feldindizes beginnen bei 1. Beachten Sie die Wirkung der Funktionen odbc_binmode und odbc_longreadlen.

odbc_result

Beispiel

odbc_result_all

odbc_result_all odbc_result_all gibt Ergebnisse als HTML-Tabelle zurück.

◆ `int odbc_result_all(int result_id, string [format]);`

Der Rückgabewert gibt die Anzahl der erzeugten Reihen an. Im Fehlerfall wird FALSE zurückgegeben. Die Ausgabe erfolgt direkt an den Browser und kann nicht in PHP-Variablen umgeleitet werden. Die Tabelle kann global mit dem optionalen Parameter *format* beeinflusst werden. Am sinnvollsten sind die Style-Anweisungen (CSS).

odbc_rollback

odbc_rollback odbc_rollback macht die innerhalb einer Transaktion vorgenommenen Änderungen an der Datenbank rückgängig.

◆ `int odbc_rollback(int connection_id);`

Im Fehlerfall wird FALSE zurückgegeben, sonst TRUE.

odbc_setoption

odbc_setoption Die Funktion odbc_setoption stellt verschiedene Parameter ein.

◆ `int odbc_setoption(int id, int function, int option, int param);`

Mit dieser Funktion lassen sich spezifische Einstellungen am ODBC-Treiber vornehmen, die konkrete Anwendung hängt also von der Datenbank, dem Treiber und dem Betriebssystem ab. *function* ist die betroffene ODBC-Funktion, *option* die zu ändernde Option und *param* der für die Einstellung benötigte Parameter.



Beispiel

```
# Option 102 von SQLSetConnectOption() ist SQL_AUTOCOMMIT
# 1 von SQL_AUTOCOMMIT ist SQL_AUTOCOMMIT_ON
odbc_setoption ($conn, 1, 102, 1);
# 2. Option 0 von SQLSetStmtOption() ist SQL_QUERY_TIMEOUT
# Diese Anweisung setzt den Timeout-Wert auf 30 Sekunden
$result = odbc_prepare ($conn, $sql);
odbc_setoption ($result, 2, 0, 30);
odbc_execute ($result);
```

Oracle 7- und Oracle 8-Funktionen

Die folgenden Funktionen gelten für Oracle 7 und 8. Verwendet wird das modernere Oracle8-Call-Interface OCI8.

ocidefinebyname

ocidefinebyname verwendet nutzerdefinierte PHP-Variablen, um Ergebnisse von SELECT-Abfragen auszuwerten. **ocidefinebyname**

```
int ocidefinebyname(int stmt, string Column-Name,  
                    mixed &variable, int [type]);
```

Beachten Sie, dass Oracle Spaltennamen in Großbuchstaben erwartet. Die übrigen Oracle-Funktionen wandeln dies automatisch um, bei dieser Funktion jedoch nicht.

Siehe auch OCINewDescriptor und OCIBindByName.

```
<?php  
$conn = ocilogon("oraclehost", "mydbase_onoracle");  
$stmt = ociparse($conn,"SELECT custID, name FROM cust_table");  
# Definitionen müssen vor der Ausführung erfolgen:  
ocidefinebyname($stmt,"EMPNO",&$custID);  
ocidefinebyname($stmt,"ENAME",&$name);  
ociexecute($stmt);  
while (ocifetch($stmt)) {  
    echo "Kundennummern:" . $custID . "<br>";  
    echo "Name:" . $name . "<br>";  
}  
ocifreestatement($stmt);  
ocilogoff($conn);  
?>
```



Beispiel

ocibindbyname

ocibindbyname bindet eine PHP-Variable an einen Platzhalter.

```
int ocibindbyname(int stmt, string ph_name,  
                  mixed &variable,  
                  int length, int [type]);
```

Die Funktion bindet die Variable *variable* an den Platzhalter *ph_name*. Die maximale Länge kann mit *length* beschränkt werden; wenn *-1* angegeben ist, wird die aktuelle Länge der Variablen übernommen.

Der optionale Parameter Type teilt den Datentyp mit:

- OCI_B_FILE. Binary-File, Binärdatei.
- OCI_B_CFILE. Character-File, Text-Datei.
- OCI_B_CLOB. Character-LOB, Textobjekt (LOB = Large Object)
- OCI_B_BLOB. Binary-LOB, Binärobjekt.



Beispiel

- OCI_B_ROWID. ROWID

Das folgende Beispiel fügt drei Datensätze ein und nutzt die ROWID für den nächsten Vorgang:

```
<?php
$conn = OCILogon("oraclserver","mydbase_onoracle");
$stmt = OCIParse($conn,"INSERT INTO cust_table (custID,name) " .
    "VALUES (:custUID, :name) " .
    "returning ROWID into :rowid");
$data = array(12683 => "Joerg",
    10999 => "Andy",
    12587 => "Frank");
$rowid = OCINewDescriptor($conn,OCI_D_ROWID);
ocibindbyname($stmt,":CustID", &$CustID, 32);
ocibindbyname($stmt,":name", &$name, 32);
ocibindbyname($stmt,":rid",&$rowid,-1,OCI_B_ROWID);
$update = OCIParse($conn,"UPDATE cust_table SET plz = :plz
    WHERE ROWID = :rid");
ocibindbyname($update,":rowid",&$rowid,-1,OCI_B_ROWID);
ocibindbyname($update,":plz",&$sal,32);
$sal = 14187;
while (list($empno,$ename) = each($data)) {
    ociexecute($stmt);
    ociexecute($update);
}
$rowid->free();
ocifreestatement($update);
ocifreestatement($stmt);
$stmt = OCIParse($conn,"SELECT * FROM cust_table WHERE cutID IN
    (12683, 10999, 12587)");
ociexecute($stmt);
while (ocifetchinto($stmt,&$arr,OCI_ASSOC)) {
    var_dump($arr);
}
OCIFreeStatement($stmt);
$stmt = ociparse($conn,"DELETE FROM cust_table WHERE cutID IN
    (12683, 10999, 12587)");
ociexecute($stmt);
ocifreestatement($stmt);
ocilogout($conn);
?>
```

ocilogon

ocilogon

ocilogon eröffnet eine Verbindung zu Oracle.



```
int ocilogon(string username, string password,
    string [OCACLE_SID]);
```

Der zurückgegebene Wert ist die von anderen Funktionen benötigte *connection_id*.

ocilogoff

ocilogoff schließt eine Verbindung.

```
int ocilogoff(int connection);
```

ocilogoff**ociexecute**

ociexecute führt eine vorbereitete Anweisung aus.

```
int ociexecute(int statement, int [mode]);
```

ociexecute

Die Funktion führt eine bereits vorbereitete Anweisung aus. Beim Vorbereiten wird die Anweisungszeichenfolge vorverarbeitet, sodass die spätere Ausführung schneller abläuft. Dies optimiert das Laufzeitverhalten kritischer Skripten.

Der optionale Parameter *mode* kann folgendermaßen definiert werden:

- OCI_COMMIT_ON_SUCCESS (Standardwert) bestätigt Transaktionen automatisch am Ende des Befehls.
- OCI_DEFAULT wird gesetzt, wenn Transaktionen explizit bestätigt (committed) werden müssen.

ocicommit

ocicommit bestätigt offene, aber bereits abgearbeitete Transaktionen und macht die Daten in der Datenbank damit gültig.

```
int OCICommit(int connection);
```

ocicommit**ocirollback**

ocirollback macht den von der letzten offenen Transaktion bewachten Vorgang rückgängig.

```
int ocirollback(int connection);
```

ocirollback**ocinumrows**

ocinumrows ergibt die Anzahl der betroffenen Datensätze.

```
int ocinumrows(int statement);
```

ocinumrows

Diese Funktion kann nur mit UPDATE eingesetzt werden, nicht aber mit SELECT.

ociresult

ociresult ociresult gibt den Wert eines Felds zurück.

◆ `mixed ociresult(int statement, mixed column);`

Die Funktion gibt alles als Zeichenkette zurück, außer bei abstrakten Datentypen. Das Auslesen aus der Ergebnisliste erfolgt dann mit `ocifetch`.

ocifetch

ocifetch `ocifetch` holt den nächsten Datensatz in einen internen Ergebnis-puffer.

◆ `mixed ocifetch(int statement);`

ocifetchinto

ocifetchinto `ocifetchinto` holt den nächsten Datensatz in ein Array.

◆ `int ocifetchinto(array &result, int[mode]);`

Das Array wird als referenzierte Variable übergeben. Es werden nur Spalten übergeben, deren Werte nicht NULL sind. Das Array ist numerisch indiziert und beginnt mit dem Index 1.

Der Parameter *mode* kann folgende Werte annehmen:

- `OCI_ASSOC` gibt ein assoziatives Array mit Feldnamen als Indizes zurück.
- `OCI_NUM`. Gibt ein numerisches Array mit dem Startindex 1 zurück, dies ist der Standardwert.
- `OCI_RETURN_NULLS`. Gibt auch Spalten mit NULL zurück.
- `OCI_RETURN_LOBS` gibt bei Binärwerten die Werte und nicht nur den Zeiger zurück.

Sie können mehrere Konstanten kombinieren, indem sie addiert werden.

ocicolumnisnull

ocicolumnisnull `ocicolumnisnull` prüft, ob eine Ergebnisspalte der Anweisung *stmt* NULL ist.

◆ `int ocicolumnisnull(int stmt, mixed column);`

ocicolumnsize

ocicolumnsize gibt die Größe einer Ergebnisspalte zurück.

```
int ocicolumnsize(int stmt, mixed column);
```

Der Spaltenindex beginnt mit 1.

ocicolumnsize**ora_bind**

ora_bind bindet eine PHP-Variable an einen Oracle-Platzhalter.

```
int ora_bind(int cursor, string PHP variable name,
            string SQL parameter name, int length,
            int [type]);
```

Im Erfolgsfall wird TRUE zurückgegeben. Die Funktionen ora_error und ora_errorcode enthalten mögliche Fehlermeldungen. Der Platzhalter wird in der SQL-Anweisung in der Form »:name« angegeben. Der optionale Parameter *type* gibt an, in welcher Richtung Werte übernommen werden:

- 0, ORA_BIND_INOUT. Beide Richtungen (Standard)
- 1, ORA_BIND_IN. Eingabeparameter
- 2, ORA_BIND_OUT. Ausgabeparameter

Die Funktion muss nach ora_parse und vor ora_exec aufgerufen werden.

```
<?php
ora_parse($curs, "declare tmp INTEGER;
                begin tmp := :in; :out := tmp; :x := 7.77;
                end;");
ora_bind($curs, "result", ":x", $len, 2);
ora_bind($curs, "input", ":in", 5, 1);
ora_bind($curs, "output", ":out", 5, 2);
$input = 333;
ora_exec($curs);
echo "Ergebnis: $result<br>OUT: $output<br>IN: $input";
?>
```



Beispiel

ora_close

ora_close schließt einen Oracle-Cursor.

```
int ora_close(int cursor);
```

Siehe auch ora_open.

ora_close

ora_columnname

ora_columnname ora_columnname holt den Namen einer Ergebnisspalte.

◆ string **ora_columnname**(int cursor,intcolumn);

ora_columntype

ora_columntype ora_columntype ermittelt den Typ einer Ergebnisspalte.

◆ string **ora_columntype**(int cursor,intcolumn);

Der Typ wird als Zeichenkette zurückgegeben und kann folgende Werte annehmen:

- "VARCHAR2"
- "VARCHAR"
- "CHAR"
- "NUMBER"
- "LONG"
- "LONG RAW"
- "ROWID"
- "DATE"
- "CURSOR"

ora_commit

ora_commit ora_commit bestätigt eine Oracle-Transaktion.

◆ int **ora_commit**(int conn);

ora_commitoff

ora_commitoff ora_commitoff schaltet die automatische Bestätigung von Transaktionen (Auto-Commit) aus.

◆ int **ora_commitoff**(int conn);

ora_commiton

ora_commiton ora_commiton schaltet die automatische Bestätigung von Transaktionen (Auto-Commit) ein.

◆ int **ora_commiton**(int conn);

ora_error

ora_error holt die letzte Oracle-Fehlermeldung.

```
string ora_error(int cursor_or_connection);
```

Gibt eine Fehlermeldung der Form XXX- NNNNN zurück, wobei XXX die Fehlerquelle angibt und NNNNN die Meldung identifiziert.

ora_error**ora_errorcode**

ora_errorcode holt den letzte Oracle-Fehlercode.

```
int ora_errorcode(int cursor_or_connection);
```

ora_errorcode**ora_exec**

ora_exec führt eine vorbereitete Anweisung aus.

```
int ora_exec(int cursor);
```

ora_exec**ora_fetch**

ora_fetch holt einen Datensatz in den Ergebnisbuffer.

```
int ora_fetch(int cursor);
```

ora_fetch**ora_getcolumn**

ora_getcolumn holt Daten aus einem Ergebnisbuffer.

```
mixed ora_getcolumn(int cursor, mixed column);
```

ora_getcolumn

Fehlerzustände sollten mit ora_errorcode erkannt werden, da sich bestimmte legale Werte als FALSE darstellen (NULL, leere Zeichenketten, die Zahl 0, die Zeichenkette »0«).

ora_logoff

ora_logoff schließt eine Oracle-Verbindung.

```
int ora_logoff(int connection);
```

ora_logoff**ora_logon**

ora_logon öffnet eine Oracle-Verbindung.

```
int ora_logon(string user, string password);
```

```
$conn = ora_Logon("user @TNSNAME", "pass");
```

ora_logon

Beispiel

ora_open

ora_open ora_open öffnet einen Oracle-Cursor.

◆ `int ora_open(int connection);`

ora_parse

ora_parse ora_parse bereitet eine Anweisung vor.

◆ `int ora_parse(int cursor_ind, string sql_statement,
int defer);`

ora_rollback

ora_rollback ora_rollback macht eine Transaktion rückgängig.

◆ `int ora_rollback(int connection);`

PostgreSQL-Funktionen

PostgreSQL ist eine Open Source-Datenbank. Die aktuellste Version finden Sie unter <http://www.postgres.org>.

Status prüfen

Der folgende Ablauf zeigt, wie Sie die Verbindung zur PostgreSQL-Datenbank herstellen:

```
$ postmaster & pg_connect("", "", "", "", "dbname");
OK
$ postmaster -i & pg_connect("", "", "", "", "dbname");
OK
$ postmaster & pg_connect("localhost", "", "", "", "dbname");
Unable to connect to PostgreSQL server: connectDB() failed: Is the
postmaster running and accepting TCP/IP (with -i) connection at
'localhost' on port '5432'? in /path/to/file.php3 on line 20.
$ postmaster -i & pg_connect("localhost", "", "", "", "dbname");
OK
```

Transaktionen

```
$conn = pg_connect("host=localhost port=3390 dbname=chris");
```

Große Objekte müssen in Transaktionsblöcke eingeschlossen werden. Transaktionsblöcke werden mit folgenden Parametern in der Funktion pg_Exec gesteuert:

- "begin". Start des Blocks
- "end". Ende des Blocks

- "commit". Bestätigen und Absenden
- "abort". Abbrechen
- "rollback". Rückgängig machen.

```
<?php
$database = pg_Connect ("", "", "", "", "pg_server");
pg_exec ($database, "begin");
$oid = pg_locreate ($database);
echo "ID: $oid <br>";
$handle = pg_loopen ($database, $oid, "w");
echo "Handle: $handle <br>";
pg_lowrite ($handle, "testwert");
pg_loclose ($handle);
pg_exec ($database, "commit")
pg_exec ($database, "end")
?>
```



pg_close

pg_close schließt eine Verbindung *connection*.

```
bool pg_close(int connection);
```

pg_close



pg_cmdtuples

pg_cmdtuples gibt die Anzahl der von einer Operation betroffenen Datensätze zurück.

```
int pg_cmdtuples(int result_id);
```

pg_cmdtuples



Von der Ausführung von INSERT, UPDATE oder DELETE ist eine bestimmte Anzahl Datensätze betroffen; diese Anzahl wird mit der Funktion ermittelt. War die Operation erfolglos, wird 0 zurückgegeben.

```
<?php
$result = pg_exec($conn,
    "INSERT INTO cust_table VALUES ('name')");
$cmdtuples = pg_cmdtuples($result);
echo $cmdtuples . " Datensätze betroffen.<p>";
?>
```



pg_connect

pg_connect öffnet eine normale Verbindung zur Datenbank.

```
int pg_connect(string host, string port,
    [string options],
    [string tty], string dbname);
```

pg_connect



Gibt das bei anderen Funktionen mit *conn* oder *connection* bezeichnete Handle zurück. Im Fehlerfall wird FALSE zurückgegeben. Mehrere Verbindungen können parallel bestehen.



```
$conn = pg_connect("dbname=mypg_dbase port=3390");
```

Siehe auch `pg_pconnect`.

pg_dbname

pg_dbname `pg_dbname` gibt den Namen der Datenbank zurück, die von der angegebenen Verbindung *connection* angesprochen wird.

◆ `string pg_dbname(int connection);`

pg_errormessage

pg_errormessage `pg_errormessage` gibt Fehlermeldungen zurück.

◆ `string pg_errormessage(int connection);`

Nach einem Fehler gibt diese Funktion die Fehlermeldung des Servers zurück. Damit besteht Zugriff auf die originalen Fehlermeldungen des PostgreSQL-Servers.

pg_exec

pg_exec `pg_exec` führt eine SQL-Abfrage aus.

◆ `int pg_exec(int connection, string query);`

Die Funktion sendet die Abfrage an den Server, übernimmt die Ergebnisse und gibt die ID der Ergebnisliste (bei anderen Funktionen als *result_id* bezeichnet) zurück. *connection* muss ein gültiges Handle für eine Verbindung zu einem PostgreSQL-Server sein.

Fehler können über die Funktion `pg_errormessage` ausgewertet werden.

pg_fetch_array

pg_fetch_array `pg_fetch_array` liest einen Datensatz einer Ergebnisliste in ein assoziatives Array ein.

◆ `array pg_fetch_array(int result, int row);`

Gibt ein Array zurück, das dem Datensatz *row* der Ergebnisliste *result_id* entspricht. Die Feldnamen ergeben die Indizes des Arrays. Sind keine Reihen mehr vorhanden, wird FALSE zurückgegeben.

Siehe auch `pg_fetch_row`.

```
<?php
$conn = pg_pconnect("", "", "", "", "publisher");
if (!$conn) {
    echo "Ein Fehler trat auf!<br>";
    exit;
}
$result = pg_Exec ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Ein Fehler trat auf!<br>";
    exit;
}
$arr = pg_fetch_array ($result, 0); echo $arr[0] . " <- array\n";
$arr = pg_fetch_array ($result, 1);
echo $arr["author"] . " (Musterausgabe)<br>";
?>
```



pg_fetch_object

`pg_fetch_object` holt eine Reihe der Ergebnisliste als Objekt.

object `pg_fetch_object`(int result_id, int row);

Gibt ein Objekt zurück, das dem Datensatz *row* der Ergebnisliste *result_id* entspricht. Die Feldnamen ergeben die Eigenschaften des Objekts. Sind keine Reihen mehr vorhanden, wird `FALSE` zurückgegeben. Die Funktion ist nicht langsamer als `pg_fetch_array` oder `pg_fetch_row`.

Siehe auch `pg_fetch_array` und `pg_fetch_row`.

```
<?php
$database = "verlag";
$db_conn = pg_connect ("localhost", "3390", "", "", $database);
if (!$db_conn) {
    echo "Fehler bei der Verbindung zu: $database";
}
exit;
endif;
$query = pg_exec ($db_conn,
    "SELECT * FROM publisher ORDER BY author");
$row = 0; # postgres braucht einen Reihenzähler
while ($data = pg_fetch_object ($query, $row)):
    echo $data->autor . " (";
    echo $data->jahr . "): ";
    echo $data->titel . "<BR>";
    $row++;
endwhile;
?>
<PRE>
<?php
```



pg_fetch_object



```

$fields[] = array("autor", "Author");
$fields[] = array("jahr", " Year");
$fields[] = array("titel", " Title");
$row= 0; # postgres braucht einen Reihenzähler
while ($data = pg_fetch_object ($qu, $row)) {
    echo "<br>";
    reset ($fields);
    while (list (,$item) = each ($fields)) {
        echo $item[1] . ": " . $data->$item[0] . "<br>";
    }
    $row++;
}
echo "<p></PRE>";
<?php
pg_freeResult ($qu);
pg_close ($db_conn);
?>

```

pg_fetch_row

pg_fetch_row

pg_fetch_row holt einen Datensatz der angegebenen Ergebnisliste und überträgt sie in ein indiziertes Array.



array **pg_fetch_row**(int result_id, int row);

Der Datensatz *row* der Ergebnisliste *result_id* wird in ein indiziertes Array übertragen. Die Feldnummern ergeben die Indizes mit dem Offset 0. Wenn keine Reihen mehr vorhanden sind, wird FALSE zurückgeben. Fortlaufende Aufrufe setzen den internen Datensatzzeiger der Ergebnisliste weiter.

Siehe auch pg_fetch_array, pg_fetch_object, pg_result.



Beispiel

```

<?php
$conn = pg_pconnect("", "", "", "", "publisher");
if (!$conn) {
    echo "Fehler aufgetreten.<br>";
    exit;
}
$result = pg_Exec($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Fehler aufgetreten. <br>";
    exit;
}
$row = pg_fetch_row($result, 0);
echo $row[0] . " <- row\n";
$row = pg_fetch_row($result, 1);
echo $row[0] . " <- row\n";
$row = pg_fetch_row($result, 2);
echo $row[1] . " <- row\n";
?>

```

pg_fieldisnull

pg_fieldisnull prüft, ob ein Feldinhalt NULL ist.

```
int pg_fieldisnull(int result_id, int row,  
                  mixed field);
```

Gibt 1 (TRUE) zurück, wenn das durch *row* und *field* identifizierte Feld der Ergebnisliste *result_id* NULL ist, sonst 0 (FALSE).

pg_fieldisnull**pg_fieldname**

pg_fieldname gibt einen Spaltennamen zurück.

```
string pg_fieldname(int result_id, int field_number);
```

Die Spalten zählen von 0 beginnend in der Ergebnisliste *result_id*. Im Fehlerfall wird -1 zurückgegeben.

pg_fieldname**pg_fieldnum**

pg_fieldnum gibt eine Spaltennummer zurück.

```
int pg_fieldnum(int result_id, string field_name);
```

Die Spalten zählen von 0 beginnend in der Ergebnisliste *result_id*. Im Fehlerfall wird -1 zurückgegeben.

pg_fieldnum**pg_fieldprtl**

pg_fieldprtl gibt die Länge eines Felds beim Ausdruck des Inhalts zurück.

```
int pg_fieldprtl(int result_id, int row_number,  
                string field_name);
```

Die Funktion ermittelt die für den Ausdruck (oder Bildschirmausgabe) erzeugte Stellenzahl eines Feldinhalts. Reihe *row_number* und Feldname *field_name* müssen spezifiziert werden. Die Reihe beginnt mit 0. Im Fehlerfall wird -1 zurückgegeben.

pg_fieldprtl**pg_fieldsize**

pg_fieldsize gibt den internen Platzverbrauch eines Felds zurück.

```
int pg_fieldsize(int result_id, string field_name);
```

Die Funktion gibt den internen Speicherverbrauch eines Felds in Bytes zurück. Der Rückgabewert -1 markiert Felder mit variabler Länge.

pg_fieldsize

pg_fieldtype

pg_fieldtype pg_FieldType gibt den Namen des Datentyps eines Felds zurück.

◆ `string pg_fieldtype(int result_id, int field_number);`

Die Nummerierung *field_number* der Felder einer Ergebnisliste *result_id* beginnt mit 0.

pg_freeresult

pg_freeresult pg_FreeResult gibt Speicher frei.

◆ `int pg_freeresult(int result_id);`

Mit dieser Funktion geben Sie Speicher während der Laufzeit eines Skripts frei. Am Ende des Skripts wird der Speicher automatisch freigegeben. Die Anwendung ist nur sinnvoll, wenn sehr viele Daten innerhalb des Skripts Speicher benötigen.

pg_getlastoid

pg_getlastoid pg_getlastoid gibt die letzte *oid* zurück, dies ist die Adresse eines großen Objekts.

◆ `int pg_getlastoid(int result_id);`

Die Anwendung ist sinnvoll, wenn das Objekt mit INSERT in die Datenbank eingefügt wurde und nun die *oid* benötigt wird. Im Fehlerfall wird -1 zurückgegeben.

pg_host

pg_host pg_host gibt den Namen des PostgreSQL-Hosts zurück.

◆ `string pg_host(int connection_id);`

pg_loclose

pg_loclose pg_loclose schließt ein großes Objekt, das durch *fd* adressiert wird.

◆ `void pg_loclose(int fd);`

pg_locreate

pg_locreate pg_locreate erzeugt ein großes Objekt.

◆ `int pg_locreate(int conn);`

Die Funktion erzeugt ein großes Objekt und gibt die *oid* zurück (Object Identifier). *conn* gibt eine bestehende Verbindung zu PostgreSQL an. Die PostgreSQL-Zugriffsmodi *INV_READ*, *INV_WRITE* und *INV_ARCHIVE* werden nicht unterstützt. Standardmäßig wird das Objekt zum Lesen und Schreiben erzeugt.

pg_loopen

`pg_loopen` öffnet ein großes Objekt und gibt einen Zeiger darauf zurück, mit dem anderen Funktionen der Zugriff gewährt wird.

```
int pg_loopen(int conn, int objoid, string mode);
```

Die Funktion öffnet ein »Inversion Large Object«. Der Zeiger enthält Informationen über die Verbindung. Sie dürfen die Verbindung nicht schließen, bevor nicht das Objekt geschlossen wurde. *objoid* gibt eine gültige *oid* des Objekts an. *mode* kann einen der folgenden Werte annehmen:

- "w". Nur Schreiben
- "r". Nur Lesen
- "rw". Lesen und Schreiben

pg_loread

`pg_loread` liest ein großes Objekt.

```
string pg_loread(int fd, int len);
```

Die Funktion liest von einem großen, durch *fd* adressierten Objekt *len* Zeichen ein und gibt diese zurück.

pg_loreadall

`pg_loreadall` liest ein großes Objekt komplett ein.

```
void pg_loreadall(int fd);
```

Die Funktion liest ein großes, durch *fd* adressiertes Objekt ein und sendet es direkt, zusammen mit den erforderlichen Headerzeilen, an den Browser. Typischerweise lassen sich damit Bilder aus der Datenbank auslesen und darstellen.

pg_lounlink

`pg_lounlink` löscht ein großes Objekt.

```
void pg_lounlink(int conn, int lobjid);
```

pg_loopen



pg_loread



pg_loreadall



pg_lounlink



lobjid identifiziert das große Objekt.

pg_lowrite

pg_lowrite pg_lowrite schreibt ein großes Objekt.

◆ `int pg_lowrite(int fd, string buf);`

Die Funktion schreibt ein großes Objekt (lo = large object) in die Datenbank, dabei ist *fd* das Filehandle, das mit `pg_loopen` erzeugt wurde, *buf* die Variable, die das große Objekt enthält.

pg_numfields

pg_numfields pg_numfields gibt die Anzahl der Felder in einer Ergebnisliste zurück.

◆ `int pg_numfields(int result_id);`

Die *result_id* wird von einem vorangegangenen `pg_Exec` erzeugt. Im Fehlerfall wird -1 zurückgegeben.

pg_numrows

pg_numrows pg_numrows gibt die Anzahl der von der letzten Abfrageoperation betroffenen Datensätze zurück.

◆ `int pg_numrows(int result_id);`

Die *result_id* wird von einem vorangegangenen `pg_Exec` erzeugt. Im Fehlerfall wird -1 zurückgegeben.

pg_options

pg_options pg_options gibt die von der aktuellen oder angegebenen Verbindung genutzten Einstellungen als Zeichenkette zurück.

◆ `string pg_options(int connection_id);`

pg_pconnect

pg_pconnect pg_pconnect stellt eine persistente (ständige) Verbindung her.

◆ `int pg_pconnect(string host, string port,
string options, string tty,
string dbname);`

Die Funktion gibt im Erfolgsfall ein Verbindungshandle zurück, andernfalls FALSE. Persistente Verbindungen sind nur möglich, wenn PHP als Modul im Webserver läuft. Die Argumente müssen

in Anführungszeichen gesetzt werden. *options* und *tty* ist optional. Sie können mehrere Verbindungen parallel öffnen.

Siehe auch `pg_Connect`.

```
$conn = pg_pconnect("dbname=my_dbase port=3390");
```



pg_port

`pg_port` gibt die Portnummer der aktuellen Verbindung zum PostgreSQL-Server zurück.

```
int pg_port(int connection_id);
```

pg_port

pg_result

`pg_result` gibt Ergebnisse zurück.

```
mixed pg_result(int result_id, int row_number, mixed fieldname);
```

pg_result

`pg_Result` gibt Ergebnisse des durch *result_id* angegebenen Ergebnissatzes zurück. Die *result_id* wird durch `pg_Exec` erzeugt. Die Parameter *row_number* und *fieldname* spezifizieren das Feld in der Ergebnisliste, die Nummerierung beginnt bei 0, alternativ kann auch der Name angegeben werden. Die Funktion gibt nur die Basistypen `integer`, `double` und `string` zurück, alle anderen internen PostgreSQL-Formate werden entsprechend konvertiert.

pg_tty

`pg_tty` gibt den Namen des tty-Terminals zurück.

```
string pg_tty(int connection_id);
```

pg_tty

Für das PostgreSQL-Verbindungshandle *connection_id* wird der serverseitige Debugger-Port zurückgegeben.

Sybase-Funktionen

sybase_affected_rows

`sybase_affected_rows` ergibt die Anzahl der bearbeiteten Datensätze der letzten SQL-Anweisung.

```
int sybase_affected_rows(int [link_identifizier]);
```

sybase_affected_rows

`sybase_affected_rows` gibt die Anzahl der von INSERT, UPDATE oder DELETE behandelten Reihen zurück.

Wird *link_identifier* nicht angegeben, wird die aktuell oder zuletzt geöffnete Verbindung verwendet. Um die Anzahl der Reihen einer SELECT-Abfrage zu ermitteln, verwenden Sie die Funktion `sybase_num_rows`.

sybase_close

sybase_close `sybase_close` schließt eine Verbindung zu einer Sybase-Datenbank.

◆ `int sybase_close(int link_identifier);`

Gibt TRUE oder (im Fehlerfall) FALSE zurück.

Verbindungen werden am Ende eines Skripts automatisch geschlossen, die Anwendung ist nur innerhalb des Skripts sinnvoll. Persistente Verbindungen können mit dieser Funktion nicht geschlossen werden.

Siehe auch: `sybase_connect`, `sybase_pconnect`.

sybase_connect

sybase_connect `sybase_connect` öffnet eine reguläre Verbindung zu einer Sybase-Datenbank.

◆ `int sybase_connect(string servername,
string username,
string password);`

Der Servername muss in der Datei »interfaces« definiert sein. Wenn ein erneuter Aufruf der Funktion bei geöffneter Verbindung erfolgt, wird keine neue Verbindung etabliert, wenn die Parameter identisch zu einer bestehenden Verbindung sind. Verbindungen werden am Ende des Skripts automatisch geschlossen. Innerhalb des Skripts verwenden Sie `sybase_close`.

Siehe auch `sybase_pconnect` und `sybase_close`.

sybase_data_seek

sybase_data_seek `sybase_data_seek` bewegt den internen Datensatzcursor.

◆ `int sybase_data_seek(int result_identifier, int row_number);`

Der Cursor wird auf den bezeichneten Datensatz gesetzt. Der nächste Aufruf von `sybase_fetch_row` gibt den durch den Cursor bezeichneten Datensatz zurück.

Siehe auch `sybase_data_seek`.

sybase_fetch_array

Die Funktion `sybase_fetch_array` liest den aktuellen Datensatz aus der Ergebnisliste aus und übergibt ihn in einem Array.

sybase_fetch_array

```
int sybase_fetch_array(int result);
```



Die Funktion gibt `FALSE` zurück, wenn keine Daten mehr vorhanden sind. `sybase_fetch_array` ist eine erweiterte Version der Funktion `sybase_fetch_row`.

Die Elemente des Arrays sind sowohl über Indizes als auch über assoziative Schlüssel erreichbar. Die Schlüsselnamen entsprechen den Spaltennamen. Die Funktion ist nicht langsamer als `sybase_fetch_row`.

sybase_fetch_field

Die Funktion `sybase_fetch_field` liest den Wert eines bestimmten Felds anhand des Offsets (Feldnummer).

sybase_fetch_field

```
object sybase_fetch_field(int result,  
                          int field_offset);
```



Die Funktion gibt ein Objekt mit den Feldinformationen zurück. Wird kein Offset `field_offset` angegeben, wird das nächste ungelesene Feld zurückgegeben.

Die Eigenschaften des Objekts sind:

- *name*. Spaltenname. Wenn die Spalte das Ergebnis einer Berechnung ist, wird als Name `computed#N` gebildet, wobei N eine fortlaufende Nummer ist.
- *column_source*. Name der Tabelle, die als Datenquelle diente.
- *max_length*. Maximale Länge (Breite) einer Spalte.
- *numeric*. Diese Eigenschaft ist 1, wenn es sich um einen numerischen Datentyp handelt.

Siehe auch `sybase_field_seek`.

sybase_fetch_object

Die Funktion `sybase_fetch_object` liest einen Datensatz in ein Objekt ein.

sybase_fetch_object

```
int sybase_fetch_object(int result);
```



Die Funktion arbeitet wie `sybase_fetch_array`. Die Feldnamen sind die Namen der Eigenschaften. Der Zugriff über Indizes ist nicht möglich. Es gibt keine signifikanten Geschwindigkeitsunterschiede zu `sybase_fetch_array` und `sybase_fetch_row`.

Siehe auch `sybase_fetch_array` und `sybase_fetch_row`.

sybase_fetch_row

sybase_fetch_row `sybase_fetch_row` holt einen Datensatz in ein Array.

◆ `array sybase_fetch_row(int result);`

Die Funktion verwendet nur ein indiziertes Array. Wenn keine Daten zu holen sind, wird `FALSE` zurückgegeben. Die Funktion setzt außerdem den internen Datensatzzeiger weiter.

Siehe auch `sybase_fetch_array`, `sybase_fetch_object`, `sybase_data_seek`, `sybase_fetch_lengths` und `sybase_result`.

sybase_field_seek

sybase_field_seek `sybase_field_seek` setzt den Datensatzzeiger auf eine bestimmte Position.

◆ `int sybase_field_seek(int result,
int field_offset);`

Siehe auch `sybase_fetch_field`.

sybase_free_result

sybase_free_result `sybase_free_result` gibt den Speicher frei, der durch die Ergebnisliste belegt wird.

◆ `int sybase_free_result(int result);`

Am Ende eines Skripts wird dieser Speicher automatisch geräumt. Der Aufruf macht nur Sinn, wenn es Speicherprobleme innerhalb eines Skripts gibt.

sybase_num_fields

sybase_num_fields `sybase_num_fields` gibt die Anzahl der Felder in einer Ergebnisliste zurück.

◆ `int sybase_num_fields(int result);`

Siehe auch: `sybase_db_query`, `sybase_query`, `sybase_fetch_field`, `sybase_num_rows`.

sybase_num_rows

sybase_num_rows gibt die Anzahl der Datensätze in der ausgewählten Ergebnisliste zurück.

sybase_num_rows

```
int sybase_num_rows(string result);
```

Siehe auch: sybase_db_query, sybase_query, sybase_fetch_field, sybase_num_fields.

sybase_pconnect

Die Funktion sybase_pconnect öffnet eine persistente (ständige) Verbindung.

sybase_pconnect

```
int sybase_pconnect(string servername,  
                   string username,  
                   string password);
```

Im Fehlerfall wird FALSE zurückgegeben, sonst das Verbindungshandle.

sybase_pconnect arbeitet ähnlich wie sybase_connect.

Die Verbindung wird nicht geschlossen, wenn das Skript endet. Wenn in einem anderen Skript eine Verbindung mit denselben Daten angefordert wird, versucht die Funktion zuerst eine persistente Verbindung wiederzubeleben, bevor eine neue geöffnet wird. Verbindungen mit dieser Funktion können nicht mit sybase_close geschlossen werden.

sybase_query

sybase_query sendet eine SQL-Anweisung an den Sybase-Datenbank-server.

sybase_query

```
int sybase_query(string query, int link_identifier);
```

Der Parameter *link_identifier* ist optional, ohne Angabe wird die aktuelle oder zuletzt geöffnete Verbindung genutzt.

Siehe auch: sybase_db_query, sybase_select_db, sybase_connect.

sybase_result

sybase_result holt die Ergebnisse und gibt ein Handle zurück.

sybase_result

```
int sybase_result(int result, int i, mixed field);
```

Der Parameter *field* kann der Feldindex, der Feldname oder die komplette Bezeichnung *tabelle.name* sein. Wenn in einer Abfrage ein Alias genutzt wird, muss hier der Alias auch verwendet werden.

Es ist schneller, den gesamten Datensatz zu lesen, wenn mehr als ein Feld benötigt wird. Außerdem ist die Verwendung eines numerischen Indizes schneller als die eines Feldnamens. Schneller sind generell die Funktionen `sybase_fetch_row`, `sybase_fetch_array` und `sybase_fetch_object`.

sybase_select_db

sybase_select_db `sybase_select_db` wählt eine Datenbank aus.

◆ `int sybase_select_db(string database_name, int link_identifizier);`

Im Erfolgsfall wird `TRUE` zurückgegeben. Wird der Parameter *link_identifizier* nicht angegeben, wird die zuletzt geöffnete Verbindung genutzt. Die Verwendung entspricht dem SQL-Befehl `USE`.

Siehe auch: `sybase_connect`, `sybase_pconnect` und `sybase_query`.

A.9 Systemnahe Funktionen

Funktionen zur Ausgabesteuerung

Die Funktionen zur Ausgabesteuerung kontrollieren den Ausgabepuffer. PHP schreibt dann Daten zuerst in den Puffer und sendet sie von dort an den Webserver. Diese Funktionen sind neu in PHP4.



flush

Die Funktion leert den Ausgabepuffer und sendet den Inhalt an den Webserver.

flush

```
void flush(void)
```



Beachten Sie, dass der gewünschte Effekt möglicherweise nicht sichtbar wird, wenn Sie HTML-Tabellen ausgeben. Der Browser stellt eine Tabelle erst dar, wenn sie vollständig ist.



ob_end_clean

Diese Funktion löscht den Pufferinhalt und schaltet die Pufferung anschließend aus.

ob_end_clean

```
void ob_end_clean(void)
```



ob_end_flush

Diese Funktion sendet den Pufferinhalt und schaltet die Pufferung anschließend aus.

ob_end_flush

```
void ob_end_flush(void)
```



ob_get_contents

Die Funktion liest den gesamten Inhalt des Puffers und gibt ihn als Zeichenkette zurück.

ob_get_contents

```
string ob_get_contents(void)
```



Sie können diese Funktion beispielsweise dazu einsetzen, bereits erzeugte Daten nachträglich zu manipulieren. Dazu wird der Puffer ausgelesen und gelöscht, der Inhalt verändert, mit echo wieder ausgegeben und dann gesendet.

ob_implicit_flush

ob_implicit_flush Die Funktion schaltet das implizite Leeren des Puffers ein. Nach jeder vollständigen Ausgabe, beispielsweise mit `echo` oder `print`, wird der Puffer geleert und der Inhalt gesendet.

◆ `void ob_implicit_flush([int flag])`

Nach dieser Funktion sind weitere `flush`-Aufrufe wirkungslos.

Siehe auch: `flush`, `ob_end_flush`

ob_start

ob_start Die Funktion startet die Ausgabepufferung. Wird diese nicht gestartet, werden alle erzeugten Daten sofort zum Webserver gesendet.

◆ `void ob_start(void)`

Verbindungsfunktionen**connection_aborted**

connection_aborted `connection_aborted` gibt `TRUE` zurück, wenn der Client nicht mehr verbunden ist.

◆ `int connection_aborted(void);`

connection_status

connection_status Die Funktion `connection_status` gibt einen Statusbericht in Form eines Bitfelds zurück.

◆ `int connection_status(void);`

connection_timeout

connection_timeout `connection_timeout` gibt `TRUE` zurück, wenn das Skript wegen Zeitüberschreitung endet.

◆ `int connection_timeout(void);`

Der Standardwert für die Zeitüberschreitung ist 30 Sekunden.

ignore_user_abort

ignore_user_abort `ignore_user_abort` legt fest, ob ein Skript bei Zeitüberschreitung beendet werden soll.

```
int ignore_user_abort(int [setting]);
```



Datenverarbeitungsfunktionen

pack

pack packt Daten in eine binäre Form.

pack

```
string pack(string format, mixed [args]...);
```



Die verschiedenen Daten werden in einer Zeichenkette kodiert. Die Funktion entspricht der Variante in Perl. Das Format besteht aus Formatcodes und Wiederholungsargumenten. Das Wiederholungsargument kann eine Ganzzahl oder ein * sein. Für a, A, h und H bestimmt dieser Wert, wie viele gleichartige Zeichen folgen. bei @ wird die absolute Position bezeichnet, ab der das nächste Zeichen steht.

Derzeit sind folgende Codes implementiert:

- a Mit NULL abgeschlossene Zeichenkette
- A Mit SPACE abgeschlossene Zeichenkette
- h Zeichenkette aus Hex-Zahlen, unteres Nibble zuerst
- H Zeichenkette aus Hex-Zahlen, oberes Nibble zuerst
- c Zeichen mit Vorzeichen
- C Zeichen ohne Vorzeichen
- s Short, mit Vorzeichen (immer 16 bit, Maschinen-Reihenfolge)
- S Short, ohne Vorzeichen (16 bit, Maschinen-Reihenfolge)
- n Short, ohne Vorzeichen (16 bit, Großes Byte zuerst)
- v Short, ohne Vorzeichen (16 bit, Kleines Byte zuerst)
- i Integer, Vorzeichen, alle Parameter maschinenabhängig
- I Integer, ohne Vorzeichen, alle Parameter maschinenabhängig
- l Long, mit Vorzeichen (32 bit, Großes Byte zuerst)
- L Long, ohne Vorzeichen (32 bit, Großes Byte zuerst)
- N Long, ohne Vorzeichen (32 bit, Großes Byte zuerst)
- V Long, ohne Vorzeichen (16 bit, Kleines Byte zuerst)

- f Float, Vorzeichen, alle Parameter maschinenabhängig
- d Double, Vorzeichen, alle Parameter maschinenabhängig
- x NULL-Byte
- X 1 Byte
- @ NULL an der absoluten Position aufgefüllt



Beispiel

```
$binarydata = pack("nvc*", 0x1234, 0x5678, 65, 66);
```

Das Ergebnis ist eine 6 Byte große Sequence:

```
0x12, 0x34, 0x78, 0x56, 0x41, 0x42.
```

serialize

serialize

serialize erzeugt ein speicherbares Format von Daten.



```
string serialize(mixed value);
```

Die Funktion verpackt Variablen so, dass die Struktur und andere interne Informationen erhalten bleiben. Die Funktion erhält die Datentypen integer, double, string, array (mehrdimensional) und object (Eigenschaften werden serialisiert, Methoden gehen aber verloren).



Beispiel

```
$conn = odbc_connect("krause", "php", "cust_table");
$stmt = odbc_prepare($conn,
    "UPDATE sessions SET data = ? WHERE id = ?");
$sqldata = array(serialize($session_data), $PHP_AUTH_USER);
if (!odbc_execute($stmt, &$sqldata)) {
    $stmt = odbc_prepare($conn,
        "INSERT INTO sessions (id, data) VALUES(?, ?)");
    if (!odbc_execute($stmt, &$sqldata)) {
        echo "Ein Fehler trat auf.<p>";
    }
}
```

unpack

unpack

unpack entpackt mit pack verpackte Daten in ein Array.



```
array unpack(string format, string data);
```



Beispiel

```
$array = unpack("c2chars/nint", $binarydata);
```

Das resultierende Array enthält die Elemente "chars1", "chars2" und "int".

Siehe auch pack.

unserialize

unserialize erzeugt aus einer serialisierten Datensammlung Variablen.

unserialize

```
mixed unserialize(string str);

$conn = odbc_connect("mydbase", "php", "krause");
$stmt = odbc_prepare($conn,
    "SELECT * FROM session WHERE sid = ?");
$sqldata = array($PHP_AUTH_USER);
if (!odbc_execute($stmt, &$sqldata)
    || !odbc_fetch_into($stmt, &$tmp)) {
    $session_data = array();
} else {
    $session_data = unserialize($tmp[0]);

    if (!is_array($session_data)) {
        $session_data = array();
    }
}
```



Beispiel

Apache-Funktionen

Zum Verständnis der Funktionen konsultieren Sie bitte auch das Handbuch zum Apache-Webserver.

apache_lookup_uri

apache_lookup_uri führt eine Teilabfrage einer spezifischen URI aus und gibt alle Informationen darüber zurück.

apache_lookup_uri

```
class apache_lookup_uri(string filename);
```



Die untersuchte URI wird als Klasse mit folgenden Eigenschaften zurückgegeben:

- status
- the_request
- status_line
- method
- content_type
- handler
- uri
- filename
- path_info
- args
- boundary
- no_cache
- no_local_copy
- allowed

- send_bodyct
- bytes_sent
- byterange
- clength
- unparsed_uri
- mtime
- request_time

apache_note

apache_note

apache_note setzt oder liest Apache-Nachrichten.



```
string apache_note(string note_name,
                    string [note_value]);
```

Dies ist eine Apache-spezifische Funktion, die Werte der request note table setzt oder liest.

getallheaders

getallheaders

getallheaders holt alle HTTP-Header.



```
array getallheaders(void);
```

Die Funktion gibt ein assoziatives Array mit allen HTTP-Headern zurück, die der Apache zur Verfügung stellt. Beachten Sie, dass diese Funktion nur mit dem Apache funktioniert, wogegen die CGI-Umgebungsvariablen immer zur Verfügung stehen. Die Funktion phpinfo zeigt, welche CGI-Variablen Ihr Webserver bereitstellt.



Beispiel

```
$headers = getallheaders();
foreach($headers as $header => $value) {
    echo "$header: $value<br>";
}
```



Hinweis

Die Funktion kann nur eingesetzt werden, wenn Apache als Servermodul arbeitet.

virtual

virtual

virtual erstellt eine Unterabfrage und schließt eine Datei ein.



```
int virtual(string filename);
```

Die Funktion entspricht dem Einfügen einer Unterabfrage mit

```
<!--#include virtual...-->
```

Damit werden dem Apache-Server Daten zur Verarbeitung außerhalb der CGI-Umgebung übergeben. Trotzdem muss die eingeschlossene Datei CGI-Header erzeugen, um korrekt verarbeitet

werden zu können. In PHP sollten Sie zum Einschließen von Skripten die Funktionen `include` oder `require` verwenden.

Systemfunktionen

dl

`dl` lädt eine PHP-Bibliothek zur Laufzeit (Windows und Unix).

```
int dl(string library);
```

library ist der Pfad zur Bibliothek.

dl



leak

`leak` verschwendet Speicher.

```
void leak(int bytes);
```

Die Funktion wird zu Debugging-Zwecken eingesetzt.

leak



uniqid

`uniqid` erzeugt eine einmalige 32 Byte lange ID auf Basis des Präfix *prefix*.

```
int uniqid(string prefix);
```

Der Präfix kann bis zu 114 Zeichen lang sein.

```
$md5 = md5(uniqid(""));  
$unique_md5 = md5(uniqid(rand()));
```

uniqid



Programmausführung

escapeshellcmd

`escapeshellcmd` markiert Metazeichen der Shell.

```
string escapeshellcmd(string command);
```

Die Funktion durchsucht eine Zeichenkette und markiert (escaped) jedes Zeichen, das auf der Shell eine Bedeutung haben könnte. Dies ist sinnvoll, wenn Nutzer Kommandos eingeben und diese sollen auf der Shell mit den Funktionen `exec` oder `system` ausgeführt werden.

```
system(EscapeShellCmd($cmd))
```

escapeshellcmd



exec**exec**

exec führt ein externes Programm aus.

◆ `string exec(string command, string [array], int [return_var]);`

Die Funktion führt das Kommando *command*, zeigt aber keine Ausgaben an. Es gibt nur die letzte Zeile der Ausgaben des Kommandos aus. Wenn Sie alle Daten benötigen, müssen Sie die Funktion `passthru` verwenden.

Optional kann der Parameter *array* genutzt werden. In dieser Zeichenkette werden alle Ausgaben abgelegt. Wenn die Variable bereits existiert, werden die Ausgaben angehängt. Wenn Sie sicherstellen müssen, dass der Inhalt neu erstellt werden soll, müssen Sie vorher die Funktion `unset` aufrufen.

Wenn der optionale Parameter *return_var* angegeben wird, gibt die Funktion den Status des Kommandos zurück. Wenn der Nutzer von der Shellebene aus Daten eingeben kann und diese mit `exec` weiterverarbeitet werden, sollten Sie sicherstellen, dass Shellbefehle markiert werden, damit Nutzer keine böswilligen Kommandos zur Ausführung bringen können. Dies erfolgt mit `escapeshellcmd`.

Siehe auch unter `system`, `passthru`, `popen` und `escapesellcmd`.

system**system**

system startet ein externes Kommando und zeigt die Ausgaben an.

◆ `string system(string command, int [return_var]);`

Die Funktion führt das Kommando *command* aus und gibt die Ausgaben zurück. Wenn das Kommando einen Status zurückgibt, kann dieser mit dem optionalen Parameter *return_var* ermittelt werden; die Variable ist als Referenz zu übergeben.

Wenn der Nutzer von der Shellebene aus Daten eingeben kann und diese mit `system` weiterverarbeitet werden, sollten Sie sicherstellen, dass Shellbefehle markiert werden, damit Nutzer keine böswilligen Kommandos zur Ausführung bringen können. Dies erfolgt mit `escapeshellcmd`.

Die Funktion gibt die Daten direkt zum Browser aus. Wenn PHP als Servermodul läuft und die Pufferung der Seite aktiviert ist, versucht `system` alle Daten des Puffers sofort an den Browser zu senden (siehe auch `flush`).

Siehe auch unter `exec`, `passthru`, `popen` und `escapesellcmd`.

passthru

passthru führt ein externes Programm aus und zeigt die Ausgaben an.

```
string passthru(string command, int [return_var]);
```

Die Funktion entspricht weitestgehend *exec*, die Ausgaben des aufgerufenen Kommandos werden aber ohne weitere Verarbeitung direkt an den Browser ausgegeben. Das ist sinnvoll, wenn Programme Binärdaten enthalten.

Siehe auch unter *exec* und *passthru*.

eval

eval wertet eine Zeichenkette als PHP-Code und führt den Code aus.

```
void eval(string code_str);
```

Denken Sie daran, dass der Code vollständig und einwandfrei sein muss. Beispielsweise darf das schließende Semikolon nicht fehlen.

```
<?php
$name = 'Tasse';
$drink = 'Kaffee';
$str = 'Das ist eine $name mit $drink drin.<br>';
echo $str;
eval( "\$str = \"\$str\"; " );
echo $str;
?>
```

Dieses Beispiel gibt aus:

```
Das ist eine $name mit $drink drin.
Das ist eine Tasse mit Kaffee drin.
```

die

die gibt eine Meldung aus und beendet das Skript.

```
void die(string message);
```

Dieses Sprachkonstrukt gibt nichts zurück. Nach dem Aufruf ist das Skript beendet.

```
<?php
$filename = '/path/to/data-file';
$file = fopen($filename, 'r')
    or die("unable to open file ($filename)");
?>
```

exit

exit exit beendet das aktuelle Skript.

◆ `void exit(void);`

function_exists

function_exists `function_exists` gibt TRUE zurück, wenn die benannte Funktion existiert.

◆ `int function_exists(string function_name);`

openlog

openlog `openlog` öffnet eine Verbindung zum Systemlog.

◆ `int openlog(string ident, int option, int facility);`

syslog

syslog `syslog` erzeugt eine Meldung für das Systemprotokoll.

◆ `int syslog(int priority, string message);`

Die Nachricht *message* wird an das Systemprotokoll gesendet. Wenn in der Zeichenkette die Zeichenkombination `%m` steht, wird an dieser Stelle der letzte Systemfehler eingetragen. Unter Windows NT wird das Ereignisprotokoll genutzt, unter Windows 9x hat die Funktion keine Bedeutung.

closelog

closelog `closelog` schließt die Verbindung zum Log.

◆ `int closelog(void);`

debugger_on

debugger_on `debugger_on` startet den Debugger.

◆ `int debugger_on(string address);`

Der Debugger sendet alle Nachrichten an einen bestimmten TCP-Port. Die Funktion ist in der Version 3.0.12 nicht fertig implementiert.

debugger_off

debugger_off `debugger_off` stoppt den Debugger.

```
int debugger_off(void);
```



register_shutdown_function

register_shutdown_function registriert eine Funktion, die am Ende des Skripts automatisch aufgerufen wird.

register_shutdown_function

```
int register_shutdown_function(string func);
```



Diese Funktion darf keine weiteren Ausgaben enthalten. Sie dürfen also die Funktionen echo oder print nicht einsetzen.

sleep

sleep verzögert den Skriptablauf um eine Anzahl Sekunden.

sleep

```
void sleep(int seconds);
```



Siehe auch usleep.

usleep

usleep verzögert den Ablauf um eine Anzahl Mikrosekunden.

usleep

```
void usleep(int micro_seconds);
```



Siehe auch sleep.

PHP-Informationen

error_log

error_log gibt eine Fehlermeldung aus.

error_log

```
int error_log(string message, int message_type,  
             string [destination], string [extra_headers]);
```



Die Fehlermeldung wird an einen TCP-Port oder eine Datei gesendet. Der Parameter *message_type* kann entsprechend folgender Liste festgelegt werden.

- 0. Nachricht wird an das Systemprotokoll gesendet (betriebs-systemabhängig).
- 1. Nachricht wird als E-Mail versendet. In diesem Fall muss der optionale Parameter *extra_header* angegeben werden.
- 2. Nachricht wird über den PHP-Debugging-Mechanismus gesendet. Dazu muss das Debugging eingeschaltet sein. Der Debugger nutzt ebenfalls einen TCP-Port zur Kommunikation.



Beispiel

- 3. Die Nachricht wird an eine Datei angehängt.

Der folgende Code schreibt eine Fehlermeldung in das Protokoll, wenn ein Zugriff auf die Datenbank nicht möglich ist:

```
if (!Ora_Logon($username, $password)) {
    error_log("Oracle ist nicht verfügbar!", 0);
}
```

Information an den Administrator per E-Mail:

```
if (!$foo = allocate_new_foo) {
    error_log("Fehler bei der Datenbank $dbase!", 1,
        "admin@oracle-db.de");
}
```

Andere Varianten für Fehlermeldungen an das Protokoll:

```
error_log("Fehlerinfo: ", 2, "127.0.0.1:7000");
error_log("Fehlerinfo: ", 2, "loghost");
error_log("Fehlerinfo: ", 3, "/var/tmp/my-errors.log");
```

error_reporting

error_reporting

error_reporting legt fest, welche Fehler angezeigt werden.



```
int error_reporting(int [level]);
```

Der Parameter *level* bestimmt in Form einer Bitmaske, welche Fehler zu einer Ausgabe von Fehlerinformationen an den Browser führen. Folgende Zahlen oder Konstanten können dazu kombiniert werden:

- 1. E_ERROR
- 2. E_WARNING
- 4. E_PARSE
- 8. E_NOTICE
- 16. E_CORE_ERROR
- 32. E_CORE_WARNING

getenv

getenv

getenv ermittelt den Wert einer Umgebungsvariablen.



```
string getenv(string varname);
```

Gibt den Inhalt einer Umgebungsvariablen zurück, oder FALSE im Fehlerfall.

```
$ip = getenv("REMOTE_ADDR");  
echo "Ihre IP-Nummer lautet: $ip";
```



Beispiel

get_cfg_var

get_cfg_var holt den Wert einer PHP-Konfigurationsoption.

```
string get_cfg_var(string varname);
```

Gibt den Wert der Konfigurationsvariablen *varname* zurück. Im Fehlerfall wird FALSE zurückgegeben. Die Konfiguration erfolgt in der Datei PHP.INI.

get_cfg_var



get_current_user

get_current_user gibt den Namen des aktuellen Nutzers aus.

```
string get_current_user(void);
```

Siehe auch getmyuid, getmypid, getmyinode und getlastmod.

get_current_user



get_magic_quotes_gpc

get_magic_quotes_gpc gibt den Status der magic_quotes_gpc-Variablen zurück.

```
long get_magic_quotes_gpc(void);
```

Die Konfigurationsvariable magic_quotes_gpc wird abgefragt, wobei 0 für aus und 1 für ein steht.

Siehe auch get_magic_quotes_runtime, set_magic_quotes_runtime.

get_magic_quotes_gpc



get_magic_quotes_runtime

get_magic_quotes_runtime ermittelt den Status für magic_quotes zur Laufzeit.

```
long get_magic_quotes_runtime(void);
```

Die Konfigurationsvariable magic_quotes_runtime wird abgefragt, wobei 0 für aus und 1 für ein steht.

Siehe auch get_magic_quotes_gpc, set_magic_quotes_runtime.

get_magic_quotes_runtime



getlastmod

getlastmod zeigt den Zeitpunkt der letzten Änderung an.

```
int getlastmod(void);
```

getlastmod



Die Angabe erfolgt als Unix-Zeitstempel und kann dadurch leicht weiterverarbeitet werden. Im Fehlerfall wird FALSE zurückgegeben.



```
echo "Letzte Änderung am ".date( "F d Y H:i:s.", getlastmod());
```

Siehe auch `date`, `getmyuid`, `get_current_user`, `getmyinode` und `getmypid`.

getmyinode

getmyinode

`getmyinode` gibt den Inode des Skripts zurück.



```
int getmyinode(void);
```

Gibt den Inode zurück, oder FALSE im Fehlerfall.

Siehe auch `getmyuid`, `get_current_user`, `getmypid` und `getlastmod`.

getmypid

getmypid

`getmypid` gibt die Prozess-ID zurück.



```
int getmypid(void);
```

Gibt die Prozess-ID zurück, oder FALSE im Fehlerfall.

Siehe auch `getmyuid`, `get_current_user`, `getmyinode` und `getlastmod`.

getmyuid

getmyuid

`getmyuid` gibt die User-ID des Skripteigentümers zurück.



```
int getmyuid(void);
```

Gibt die User-ID zurück, oder FALSE im Fehlerfall.

Siehe auch `getmypid`, `get_current_user`, `getmyinode` und `getlastmod`.

getrusage

getrusage

`getrusage` ermittelt die Nutzung der aktuellen Ressourcen.



```
array getrusage(int [who]);
```

Die Funktion gibt ein assoziatives Array mit den Daten des Systemaufrufes `getrusage` zurück.



```
$dat = getrusage;
echo $dat["ru_nswap"];           // Anzahl der Swaps
echo $dat["ru_majflt"];         // Anzahl der Seitenfehler
echo $dat["ru_utime.tv_sec"];   // Nutzerzeit used(seconds)
echo $dat["ru_utime.tv_usec"]; // Nutzerzeit (microseconds)
```

highlight_file

Mit dieser Funktion wird der Inhalt einer Datei auf PHP- und HTML-Tags hin durchsucht und diese Information farblich abgesetzt als HTML-Code ausgegeben. Die Funktion sendet den Inhalt direkt an den Webserver.

```
void highlight_file(string filename)
```

Als Parameter wird ein Dateiname erwartet, kein Dateihandle. Für diese Funktion gibt es einen Alias `show_source`.

highlight_string

Mit dieser Funktion wird der Inhalt einer Zeichenkette auf PHP- und HTML-Tags hin durchsucht und diese Information farblich abgesetzt als HTML-Code ausgegeben. Die Funktion sendet den Inhalt direkt an den Webserver.

```
void highlight_string(string codestring)
```

Siehe auch `highlight_file`.

ini_get

`ini_get` holt den Wert einer Konfigurationsoption aus der Datei `PHP.INI`.

```
string ini_get(string varname)
```

Im Fehlerfall (Option nicht vorhanden) wird `FALSE` zurückgegeben.

ini_restore

`ini_restore` macht die letzte Änderung rückgängig und stellt den Originalwert wieder her.

```
string ini_restore(string varname)
```

ini_set

`ini_set` setzt den Wert einer Konfigurationsoption. Der alte Wert wird zurückgegeben oder `FALSE`, wenn die Option in `PHP.INI` nicht definiert war.

```
string ini_set(string varname, string newvalue)
```

phpinfo

phpinfo phpinfo gibt umfangreiche Informationen über die PHP-Installation in Form von Tabellen direkt zum Browser aus. In Kapitel 1 finden Sie eine Darstellung.



```
int phpinfo(void);
```

Siehe auch `phpversion`.

phpversion

phpversion `phpversion` zeigt die aktuelle Version an.



```
string phpversion(void);
```

Die Funktion gibt eine Zeichenkette mit der vollständigen Versionsinformation zurück.



Beispiel

Das Beispiel gibt etwa folgenden Text aus: »Die aktuelle PHP-Version ist: 4.0.2«:

```
echo "Die aktuelle PHP-Version ist: ".phpversion();
```

Siehe auch `phpinfo`.

extension_loaded

extension_loaded `extension_loaded` ermittelt, ob eine Bibliothek oder Erweiterung verfügbar ist.



```
bool extension_loaded(string name);
```

Die Funktion gibt TRUE zurück, wenn die Erweiterung oder Bibliothek verfügbar ist. Die Funktion `phpinfo` zeigt die Namen der Erweiterungen an.

Siehe auch `phpinfo`.

putenv

putenv `putenv` setzt den Wert einer Umgebungsvariablen.



```
void putenv(string setting);
```



Beispiel

```
putenv("UNIQID=$uniqid");
```

set_magic_quotes_runtime

set_magic_quotes_runtime `set_magic_quotes_runtime` setzt den aktuellen Wert der Variablen `magic_quotes_runtime`.

```
long set_magic_quotes_runtime(int new_setting);
```



Schaltet magic_quotes_runtime mit 1 ein oder mit 0 aus.

Siehe auch get_magic_quotes_gpc, get_magic_quotes_runtime.

set_time_limit

set_time_limit begrenzt die Laufzeit eines Skripts.

set_time_limit

```
void set_time_limit(int seconds);
```



Der Standardwert beträgt 30 Sekunden. Wenn das Skript nicht innerhalb dieser Zeit beendet wird, bricht der PHP-Interpreter die Verarbeitung mit einem Fehler ab. Dieser Wert kann mit der Funktion set_time_limit geändert werden. Die Angabe erfolgt in Sekunden. Die Angabe entspricht der Bearbeitung von max_execution_time in PHP.INI. Mit dem Wert 0 kann die Laufzeit auf unendlich gesetzt werden.

Wenn die Funktion aufgerufen wird, startet der Zeitzähler erneut. Wenn also nach 20 Sekunden Laufzeit die Funktion mit dem Wert 40 gerufen wird, beträgt die maximale Gesamtlaufzeit 60 Sekunden.

Im Safe-Mode hat die Funktion keine Auswirkungen. Damit soll verhindert werden, dass Webspacer-Kunden auf größeren Servern zuviel Leistung für sich in Anspruch nehmen.

var_dump

Die Funktion gibt eine Variable aus, einschließlich einer Darstellung des Datentyps und des Inhalts.

var_dump

```
void var_dump(mixed expression)
```



Das folgende Beispiel zeigt die Anwendung:

```
<pre>
<?php
    $a = array (1, 2, array ("a", "b", "c"));
    var_dump ($a);
?>
</pre>
```



Beispiel

print_r

Die Funktion gibt ähnlich wie var_dump Informationen über eine Variable aus. Die Darstellung ist umfangreicher, aber auch unübersichtlicher.

print_r

◆ void **print_r**(mixed expression)

Die Anwendung entspricht der von `var_dump`.

Beachten Sie, dass die Funktion Rekursionen auflöst. Wenn jedoch ein Array einen Verweis auf sich selbst enthält, läuft die Funktion endlos. Dies ist beispielsweise der Fall, wenn `print_r($GLOBALS)` geschrieben wird, da das Array `$GLOBALS` einen Verweis auf sich selbst enthält.

B Cross-Referenz

Die Cross-Referenz stellt Befehle verschiedener Skriptsprachen gegenüber. Ein Wertung erfolgt nicht. Der direkte Vergleich ist oft nicht gegeben. Umsteiger finden aber auf diese Weise ziemlich schnell den passenden Namen und können sich in der Referenz oder im Index besser orientieren.



B.1 Cross-Referenz VBScript

VBScript-Funktionen

VBScript	PHP	Bemerkung, siehe...
<i>Array- und Zeichenkettenfunktionen</i>		
array	array	
filter	-	
instr	strpos	
instrrev	strrpos	
join	implode, join	
lbound	-	
ubound	count	
lcase	strtolower	
ucase	strtoupper	
left	-	substr
mid	substr	
right	-	substr
len	strlen	
ltrim	ltrim	
rtrim	chop, rtrim	
trim	trim	
replace	str_replace	
space	-	str_repeat (PHP4)
split	explode	
strcomp	strcmp	

VBScript	PHP	Bemerkung, siehe...
string	str_repeat	PHP4
strreverse	strrev	
<i>Testfunktionen</i>		
isarray	is_array	
isdate	checkdate	
isempty	isset	
isnull	empty	
isnumeric	is_float, is_int	
isobject	is_object	
<i>Datums- und Zeitfunktionen</i>		
date	date("",getdate)	
time	date("",getdate)	
now	getdate	
dateadd	-	mktime
datediff	-	mktime
datepart	-	mktime
dateserial	-	
datevalue	-	checkdate
day	date("D",getdate())	
hour	date("H",getdate())	
minute	date("i",getdate())	
month	date("m",getdate())	
second	date("s",getdate())	
weekday	date("w",getdate())	
year	date("Y",getdate())	
monthname	date("X",getdate())	
weekdayname	date("x",getdate())	
timeserial	-	
timevalue	-	checkdate
<i>Formatierungsfunktionen</i>		

VBScript	PHP	Bemerkung, siehe...
FormatCurrency	printf	number_format
FormatDateTime	date	
FormatNumber	number_format	
FormatPercent	printf	
<i>Funktionen zur Typumwandlung</i>		
asc	ord	
ascb	-	ord
ascw	-	ord
CBool	-	
CByte	-	
CCur	-	
CDate	-	
Cdbl	(double)	
CSng	(double)	
Chr	chr	
CInt	(int)	
CLng	-	
CStr	(string)	
Hex	dechex	
Oct	decoct	
RGB	-	
<i>Systemfunktionen</i>		
CreateObject	new	
GetObject	-	
ScriptEngine	phpversion	
TypeName	gettype	
Vartype	gettype	

Operatoren

VBScript	PHP	Bemerkung
<i>Mathematische Operatoren</i>		

VBScript	PHP	Bemerkung
^	^	
-	-	Negation
*	*	
/	/	
\	\	
Mod	%	Rest der Division
+	+	Addition
-	-	Subtraktion
&	.	Zeichenkettenaddition
<i>Vergleichsoperatoren</i>		
=	=	
<>	!=	
<	<	
>	>	
<=	<=	
>=	>=	
is		Äquivalentes Objekt
<i>Logische Operatoren</i>		
not	not, !	Nicht
and	and, &&	Und
or	or,	Oder
xor	xor	Exklusives Oder
eqv		Äquivalent
imp		Implikation

Methoden der VBScript-Objekte

Die folgende Tabelle listet alle Methoden in alphabetischer Reihenfolge auf. Dabei wird keine Rücksicht darauf genommen, von welchen Objekten die Methoden abstammen. Das Objekt ist aber jeweils mit aufgeführt. Bei der Notation wird natürlich der Objektname durch eine Instanz (Objektvariable) des Objekts ersetzt. Insofern mag sich die hier gezeigte Notation von der in VBScript üblichen unterscheiden. Die Vorgehensweise erlaubt aber eine vergleichsweise kompakte Darstellung.

VBScript	Bemerkung
Methodennamen, alphabetisch sortiert (2. Wert)	
dictionary.add	-
folder.addfolder	mkdir
filesystemobject.buildpath	-
err.clear	-
textstream.close	fclose
file.copy	copy
folder.copy	copy
filesystemobject.CopyFile	copy
filesystemobject.CopyFolder	copy
filesystemobject.CreateFolder	mkdir
textstream.CreateTextFile	fopen
file.delete	unlink
folder.delete	rmdir
filesystemobject.DeleteFile	unlink
filesystemobject.DeleteFolder	rmdir
filesystemobject.DriveExists	-
dictionary.Exists	-
filesystemobject.FileExists	file_exists, is_file
filesystemobject.FolderExists	is_folder
filesystemobject.GetAbsolutePathName	dirname
filesystemobject.GetBaseName	basename
drive.GetDrive	-
filesystemobject.GetDriveName	-
filesystemobject.GetExtensionName	-
filesystemobject.GetFile	-
filesystemobject.GetFileName	-
filesystemobject.GetFolder	-
filesystemobject.GetFolderName	-
filesystemobject.GetSpecialFolderName	-

VBScript	Bemerkung
filesystemobject.GetTempName	-
dictionary.Items	-
dictionary.Keys	-
file.Move	rename (mit Pfadangaben)
folder.Move	rename (mit Pfadangaben)
filesystemobject.MoveFile	rename
filesystemobject.MoveFolder	rename
textstream.OpenAsTextStream	fopen
textstream.OpenTextFile	fopen
err.Raise	-
textstream.Read	fgetc
textstream.ReadAll	implode(file())
textstream.ReadLine	fgets
dictionary.Remove	-
dictionary.RemoveAll	-
textstream.Skip	fseek
textstream.SkipLine	-
textstream.Write	fputs, fwrite
textstream.WriteBlankLines	fputs("\n")
textstream.WriteLine	fputs, fwrite

Eigenschaften der Objekte

Auch die Eigenschaften der Objekte lassen sich mit PHP vielfältig abbilden.

VBScript	PHP	Siehe auch...
<i>Eigenschaften, alphabetisch nach den Namen sortiert (2. Wert)</i>		
textstream.AtEndOfLine	-	fgets
textstream.AtEndOfStream	feof	
file.Attributes	fileperms	
folder.Attributes	-	
drive.AvailableSpace	diskfree	

VBScript	PHP	Siehe auch...
textstream.Column	ftell	
dictionary.CompareMode	-	
dictionary.Count	-	
file.DateCreated	fileatime	
folder.DateCreated	fileatime	
file.DateLastAccessed	filectime	
folder.DateLastAccessed	filectime	
file.DateLastModified	filemtime	
folder.DateLastModified	filemtime	
err.Description	\$php_errormsg	
file.Drive	<p>Unter Windows können Laufwerksbuchstaben im Pfad benutzt werden. Sie können die Laufwerksangaben beispielsweise mit</p> <pre>\$drivearray = explode(":", \$path); \$driveletter = \$drivearray[0];</pre> ermittelt werden.	
folder.Drive		
file.DriveLetter		
folder.DriveLetter		
filesystemobject.Drives		
drive.DriveType		
folder.Files	-	
drive.FileSystem	-	
drive.FreeSpace	diskfree	
err.HelpContext	-	
err.HelpFile	-	
drive.IsReady	is_readable	
folder.IsRootFolder	-	
dictionary.Item	-	
dictionary.key	-	
textstream.line	-	fseek
file.name	-	
folder.name	-	
err.number	-	
file.ParentFolder	-	
folder.ParentFolder	-	

VBScript	PHP	Siehe auch...
file.Path	-	
folder.Path	-	
drive.Path	-	
drive.RootFolder	-	
drive.SerialNumber	-	
drive.ShareName	-	
file.ShortName	-	
folder.ShortName	-	
file.Size	filesize	
folder.Size	filesize	
err.source		error_reporting
folder.SubFolder	-	
drive.TotalSize	-	filesize
file.Type	lstat	
folder.Type	lstat	
drive.VolumeName	-	

Befehle

VBScript	PHP	Bemerkung
<i>Befehle, alphabetisch sortiert</i>		
CALL	-	
CONST	define	
DIM	-	
DO ... LOOP	while () { ... }	
EXIT DO	exit	
ERASE	-	
EXIT	exit	
FOR EACH ... NEXT	foreach	ab PHP 4
FUNCTION	function () { ... return xy; }	

VBScript	PHP	Bemerkung
IF ... THEN ... ELSE	if () { ... } else { }	
ON ERROR RESUME NEXT	@	@function
OPTION EXPLICIT	-	
PRIVATE	-	Standard
PUBLIC	-	
RANDOMIZE	randomize	
REDIM	-	
REM	// oder /* */	
SELECT CASE	select () :{ }	
SET	new	
SUB	function () { ... }	
WHILE WEND	while () { ... }	

Objekte für den Web-Server

ASP verfügt über eine Reihe von Objekten, die spezielle Funktionen des Webservers steuern. In PHP gehört die Steuerung des Webservers zu den Standardfunktionen. Entsprechend finden Sie eine Gegenüberstellung von ASP-Methoden und -Eigenschaften mit entsprechenden PHP-Funktionen.

VBScript	PHP	Bemerkung
<i>Application</i>		
Contents	-	
StaticObject	-	
OnEnd	-	
OnStart	-	
Lock	-	
UnLock	-	

<i>ObjectContext</i>		
OnTransactionAbort	-	

VBScript	PHP	Bemerkung
OnTransactionCommit	-	
SetAbort	-	
SetComplete	-	
<i>Request</i>		
ClientCertifikate	-	
Cookie	\$HTTP_COOKIE_VARS[]	und direct
Form	\$HTTP_FORM_VARS[]	und direct
QueryString	\$HTTP_GET_VARS[]	und direct
ServerVariables	Stehen direkt zur Verfügung	
BinaryRead	-	
TotalByte	-	
<i>Response (Methoden)</i>		
Cookies	setcookie	
AddHeader	header	
AppendToLog	error_log	
BinaryWrite	fwrite	
Clear	-	
End	-	
Flush	flush	
Redirect	header	
Write	echo	
<i>Response (Eigenschaften)</i>		
Buffer	-	
CacheControl	-	
CharSet	-	
ContentType	header	
Expires	-	
ExpiresAbsolute	-	
IsClientConnected	!connection_aborted	
PICS	-	

VBScript	PHP	Bemerkung
Status	connection_status	
<i>Server (Methoden)</i>		
CreateObject	new	
HTMLEncode	htmlentities	
MapPath	-	
URLEncode	rawurlencode	
<i>Server (Eigenschaften)</i>		
ScriptTimeOut	set_time_limit	
Session (Kollektionen)	(siehe Session-Funktionen)	PHPLIB, PHP4
Contents	-	
StaticObject	-	
Session (Methode)	(siehe Session-Funktionen)	PHPLIB, PHP4
Abandon	-	
Session (Eigenschaften)	(siehe Session-Funktionen)	PHPLIB, PHP4
CodePage	-	
LCID	-	setlocale
SessionID	(siehe Session-Funktionen)	
TimeOut	set_time_limit	

Datenbankobjekt (ADO)

Eine Entsprechung für ADO gibt es in PHP nicht. Es stehen jedoch viele Funktionen in Abhängigkeit von der eingesetzten Datenbank zur Verfügung. Alternativ können die ODBC-Funktionen genutzt werden. In jedem Fall müssen Sie jedoch zu den entsprechenden ADO-Funktionen zur Steuerung der Datenbank (wie `RS.Add` usw.) den entsprechenden SQL-Befehl ermitteln. Ohne SQL-Kenntnisse können Sie in PHP keine Datenbankapplikationen schreiben.

B.2 Cross-Referenz Perl

Auch zu Perl kann PHP eine ernsthafte Alternative darstellen. Einige Befehle lassen sich direkt übertragen, für andere gibt es überschaubare Umsetzungen.

Funktions- und Befehlsübersicht

Die Übersicht listet nur die wichtigsten Perl-Befehle auf und nicht die vielen Bibliotheken und Funktionserweiterungen, die sich im Lauf der Zeit gebildet haben. Die Übersetzung von Perl nach PHP wird deshalb nur in sehr seltenen Fällen direkt möglich sein. Die rechte Spalte zeigt Befehle an, die teilweise der gesuchten Funktion entsprechen oder als Ausgangspunkt für die Suche dienen können.

Perl	PHP	Bemerkung
<i>Arraybefehle</i>		
pop	array_pop	ab PHP4, current
push	array_push	ab PHP4
shift	array_shift	ab PHP4, next
splice	array_splice	ab PHP4
unshift	array_unshift	ab PHP4
<i>Ein- und Ausgabe</i>		
binmode	fopen	Parameter!
close	fclose	
closedir	closedir	
die	die	
eof	feof	
fileno	-	
flock	flock	
format	-	
getc	fgetc	
print	echo	
printf	printf	
read	fread	
readdir	readdir	
rewinddir	rewinddir	
seek	fseek	
sprintf	sprintf	
syscall	system, exec	

Perl	PHP	Bemerkung
sysread	-	
syswrite	-	
truncate	ftell	
warn	-	die
write	fputs, fwrite	
<i>Bibliotheksmodul (Auswahl)</i>		
do { ... }	while { ... }	
no	-	
require	require	
use	include	
<i>Hashbefehle</i>		
delete	-	
each	each	
exists	in_array (PHP4)	array_walk
keys	array_keys (PHP4)	key
values	array_values (PHP4)	
<i>Listenverarbeitung</i>		
grep	array_walk	
join	implode, join	
map	-	
reverse	strrev	
sort	sort	
<i>Skalarbehandlung</i>		
chomp	-	
chop	chop	
chr	chr	
hex	HexDec	
index	strpos	
lc	strtolower	
lcfirst	-	

Perl	PHP	Bemerkung
length	strlen	
oct	OctDec	
ord	ord	
rindex	strrpos	
substr	substr	
uc	strtoupper	
ucfirst	ucfirst	
<i>Nummerische Funktionen</i>		
abs	abs	
atan2	atan2	
cos	cos	
exp	exp	
hex	hex	
int	ceil	
log	log	
oct	oct	
rand	rand	
sin	sin	
sqrt	sqrt	
srand	srand	
<i>Reguläre Ausdrücke</i>		
pos	-	
quotemeta	quotemeta	
split	split	
study	-	
<i>Dateibehandlung</i>		
chdir	chdir	
chmod	chmod	
chown	chown	
chroot	-	

Perl	PHP	Bemerkung
glob	-	
mkdir	mkdir	
open	fopen	
opendir	opendir	
rename	rename	
rmdir	rmdir	

Programmsteuerung

Perl	PHP	Bemerkung
<i>Programmkontrolle</i>		
continue	continue	
die	die	
do	while	
eval	eval	
exit	exit	
goto	-	
last	-	break
next	-	continue
redo	-	
return	return	
<i>Kontrollbefehle</i>		
defined	defined	
local	per Default lokal	global
my	-	
scalar	-	
undef	unset	
wantarray	-	
<i>Prozessbefehle</i>		
alarm	-	
fork	-	
getppid	-	

Perl	PHP	Bemerkung
kill	-	
sleep	sleep, usleep	
system	system, exec	
wait	-	
waitpid	-	

Socketbefehle

Socketbefehle befinden sich bei Perl in der Bibliothek `socket`. Wegen der großen Bedeutung für die Webserverprogrammierung ist diese Bibliothek bei allen Distributionen dabei. PHP verfügt über Funktionen, die direkt die Socketprogrammierung unterstützen.

Perl	PHP	Bemerkung
<i>Socket-Zugriff</i>		
accept	-	
bind	-	
connect	fsocketopen	
getpeername	-	
getsocketname	-	
listen	fread	
recv	fgets	
shutdown	fclose	
socket	-	

C Server-Variablen und Statuscodes

C.1 Server-Variablen

Die folgende Übersicht zeigt alle Servervariablen auf einen Blick. Sie können in PHP direkt darauf zugreifen, da Servervariablen als globale Variablen zur Verfügung stehen, beispielsweise:

```
echo $REMOTE_ADDR;
```

Variablenname	Beschreibung	IIS	Apache
ALL_HTTP	Alle HTTP-Header, die vom Client zum Server gesendet wurden. Das Ergebnis sind Header, die mit HTTP_ beginnen.	•	•
ALL_RAW	Alle HTTP-Header, die vom Client zum Server gesendet wurden. Im Ergebnis werden Header gesendet, die kein Präfix haben.	•	•
APPL_MD_PATH	Gibt den Pfad zur Metabasis der Applikation an (nur IIS/Windows).	•	
APPL_PHYSICAL_PATH	Gibt den physischen Pfad zur Metabasis der Applikation an (nur IIS/Windows).	•	
AUTH_PASSWORD	Das Kennwort einer Autorisierung, wenn es im Kennwortfeld des Browsers eingegeben wurde (nur IIS/Windows).	•	
AUTH_TYPE	Art der Autorisierung, wenn Nutzer Zugriff auf ein geschütztes Dokument haben möchten (nur IIS/Windows).	•	
AUTH_NAME	Name des Nutzers bei Eingabe in das Kennwortfeld des Browsers.	•	
CERT_COOKIE	Eindeutige ID eines Clientzertifikats.	•	
CERT_FLAGS	Flag des Clientzertifikats, Bit 0 ist 1, wenn das Clientzertifikat vorhanden ist, Bit 1 ist 1, wenn das Clientzertifikat nicht überprüft wurde.	•	
CERT_ISSUER	Das Issuer (Herausgeber)-Feld des Clientzertifikats.	•	
CERT_KEYSIZE	Bitzahl bei einer SSL-Verbindung.	•	
CERT_SECRETKEYSIZE	Anzahl der Bits eines privaten Zertifikatschlüssels.	•	
CERT_SERIALNUMBER	Die Seriennummer des Zertifikats.	•	
CERT_SERVER_ISSUER	Das Issuer (Herausgeber)-Feld des Serverzertifikats (Issuer-Feld).	•	

Variablenname	Beschreibung	IIS	Apache
CERT_SERVER_SUBJECT	Beschreibung des Zertifikats (Server).	•	
CERT_SUBJECT	Beschreibung des Zertifikats (Client).	•	
CONTENT_LENGTH	Länge des zu sendenden Inhalts.	•	•
CONTENT_TYPE	Art des Inhalts (MIME-Type)	•	•
DATE_LOCAL	Datum/Uhrzeit des Servers		•
DATE_GMT	Datum/Uhrzeit des Servers in Zone GMT		•
DOCUMENT_NAME	Name des ausführenden Dokuments		•
DOCUMENT_ROOT	Pfad zum Dokument ohne Dateiname		•
GATEWAY_INTERFACE	Art des Interface, das der Server benutzt.	•	•
HTTPS	Ist ON, wenn der Server SSL benutzt.	•	•
HTTPS_KEYSIZE	Schlüssellänge der HTTPS-Verbindung (40bit, 128bit...)	•	•
HTTPS_SECRETKEYSIZE	Schlüssellänge bei privaten Zertifikaten	•	•
HTTPS_SERVER_ISSUER	Issuer-Feld des Serverzertifikats bei sicherer Übertragung	•	•
HTTPS_SERVER_SUBJECT	Beschreibung	•	•
INSTANCE_ID	ID-Nummer der Instanz (nur IIS)	•	
INSTANCE_META_PATH	Der Metabasispfad (nur IIS)	•	
LOCAL_ADDR	Die in der Anforderung benutzte Serveradresse	•	•
LOGON_USER	Ein Nutzerkonto	•	
PATH_INFO	Pfadinformation für den Client	•	•
PATH_TRANSLATED	Übertragung der Pfadinformation ins physische Format	•	•
QUERY_STRING	Inhalt des Querystrings (Parameter-URL)	•	•
REMOTE_ADDR	Die IP-Adresse des Nutzers	•	•
REMOTE_HOST	Name des Computers des Nutzers	•	•
REQUEST_METHOD	Die Methode der Datenübertragung eines Formulars. Kann GET, PUT oder POST sein.	•	•
REQUEST_URI	URI der Anforderung		•
SCRIPT_FILENAME	Name eines Skripts		•
SCRIPT_NAME	Name eines Skripts	•	
SERVER_NAME	Der Hostname des Servers, eine DNS- oder IP-Adresse	•	•

Variablenname	Beschreibung	IIS	Apache
SERVER_PORT	Port, der vom Server benutzt wird (normalerweise 80)	•	•
SERVER_PORT_SECURE	Port, der bei sicherer Übertragung benutzt wird (Standard: 443)	•	•
SERVER_PROTOCOL	Das verwendete Protokoll und die Version (z.B.: HTTP1.1)	•	•
SERVER_SIGNATURE	Signatur des Servers (einstellbare Info)		•
SERVER_SOFTWARE	Der Name und die Version der auf dem Server laufenden Software	•	•

Wichtige Anmerkung: Nicht alle Variablen stehen in allen Umgebungen zur Verfügung. Einige Variablen des Apache sind nur verfügbar, wenn der Server entsprechend eingerichtet wurde. Beispielsweise muss `SERVER_SIGNATURE` angegeben sein; dieser Wert ist nicht standardmäßig verfügbar. Testen Sie Skripte vor der Auslieferung auf mehreren Servern!

C.2 HTTP-Statuscodes

In der HTTP-Referenz finden Sie alle Codes der Version 1.1 in fünf Gruppen. In den jeweiligen Einleitungen wird auf Unterschiede zur Version 1.0 eingegangen. Die Liste basiert auf der Übersetzung des vom W3C am 15.5.1998 herausgegebenen Papiers.

Übersicht der Codes

Informationscodes

100 Continue

Informational 1xx

101 Switching Protocols

Erfolgsmeldungen

200 OK

Successful 2xx

201 Created

202 Accepted

203 Non-Authoritative Information

204 No Content

205 Reset Content

206 Partial Content

Umleitungsmeldungen

Redirection 3xx	300 Multiple Choices
	301 Moved Permanently
	302 Moved Temporarily
	303 See Other
	304 Not Modified
	305 Use Proxy

Client-Fehler

Client Error 4xx	400 Bad Request
	401 Unauthorized
	402 Payment Required
	403 Forbidden
	404 Not Found
	405 Method Not Allowed
	406 Not Acceptable
	407 Proxy Authentication Required
	408 Request Timeout
	409 Conflict
	410 Gone
	411 Length Required
	412 Precondition Failed
	413 Request Entity Too Large
	414 Request-URL Too Long
	415 Unsupported Media Type

Server-Fehler

Server Error 5xx	500 Internal Server Error
	501 Not Implemented
	502 Bad Gateway

503 Service Unavailable

504 Gateway Timeout

505 HTTP Version Not Supported

C.3 Die Codes im Detail

Informationscodes – Informational Codes 1xx

Diese Gruppe enthält einfache, informelle Antworten, die nur aus der Statuszeile und optional verfügbaren Headern bestehen. Der Code wird durch eine Leerzeile abgeschlossen. 1xx-Codes sind unter HTTP 1.0 nicht explizit definiert, so dass Server, die einen HTTP 1.0-Client bedienen, diese Codes nicht unbedingt ausgeben müssen.

100 Continue

Der Client kann die Anfragen fortsetzen. Damit wird der Client informiert, dass die bisherigen Anfragen empfangen wurden, die Antwort bearbeitet wird, möglicherweise aber noch zusätzliche Informationen nötig sind, um korrekt zu antworten. Der Client sollte unmittelbar danach den Rest der Anfrage senden oder diese Meldung ignorieren, wenn er bereits fertig ist. Der Server *muss* eine abschließende Meldung senden, wenn die Anfrage komplett ist.

100 Continue

101 Switching Protocols

Der Server hat eine Anfrage nach dem Wechsel des Applikationsprotokolls verstanden und ist mit dem Wechsel einverstanden. Der Server wechselt zu den Protokollen, die im Antwort-Header-Feld definiert wurden. Diese Information steht unmittelbar hinter der Leerzeile, die die Meldung 101 abschließt. Die Änderung des Protokolls sollte nur dann benutzt werden, wenn sich für die folgende Übertragung tatsächlich ein Vorteil ergibt.

101 Switching
Protocols

Erfolgsmeldungen – Successful 2xx

Diese Gruppe enthält Erfolgsmeldungen, die im Anschluss an eine erfolgreiche Transaktion gesendet werden. Erfolgreiche Transaktionen resultieren aus dem Verstehen, Akzeptieren oder Empfangen von Daten.

200 OK

200 OK Die Anfrage wurde erfolgreich bearbeitet. Die Information, die zusätzlich zur Anfrage gesendet wurde, hängt von der Art der Anfrage ab.

- Nach GET werden die Daten im Body mitgeliefert, die angefordert wurden.
- Nach HEAD werden die Headerdaten geliefert.
- Mit POST wird eine Beschreibung oder der Inhalt einer Aktion gesendet.
- Bei TRACE wird die Anfrage geliefert, die der Server verstanden hat.

201 Created

201 Created Die Anfrage wurde ausgeführt und führte zu einer neuen Ressource (Datenquelle), die erfolgreich erzeugt wurde. Die neu erzeugte Ressource kann durch verschiedene Informationen, die im Body der Meldung stehen, beschrieben werden. Antworten können eine oder mehrere URLs sein. Wenn der Server 201 sendet, ist die Ressource bereits erzeugt. Dauert die Aktion an, auch wenn sie Erfolg versprechend verläuft, sendet der Server die Meldung 202.

202 Accepted

202 Accepted Die Anfrage wurde verstanden und akzeptiert, aber noch nicht abschließend ausgeführt. Diese Meldung sollte nicht eingesetzt werden, um mehrstufige Transaktionen zu überwachen. Sie erlaubt es aber beispielsweise Batchprozessen, automatisch und kontinuierlich zu laufen, ohne dass der Server jede Aktion sofort ausführt. Im Body sollte eine Information stehen, die über den Verlauf der Aktion berichtet (erwartete Ausführungszeit, Status, Zwischenbericht).

203 Non-Authoritative Information

203 Non-Authoritative Information Der Header enthält zusätzliche (Meta-)Informationen, die nicht vom Webserver selbst erzeugt wurden, sondern von einer Software, die von einem weiteren Anbieter auf dem System eingesetzt wird (Third-Party-Software). Der Einsatz macht nur Sinn, wenn die Meldung 200 unzureichend ist.

204 No Content

Der Server hat die Anfrage verstanden, es gibt aber nichts zu senden. Damit wird der Client informiert, dass er seine Sicht der Inhalte (HTML-Seite, Formular) nicht ändern muss. Die Meldung wird mit der Leerzeile beendet.

204 No Content

205 Reset Content

Die Anfrage wurde erfolgreich ausgeführt, und der Client sollte nun das Dokument neu aufbauen. Bei Formularen sollte der Client die Felder löschen und die erneute Eingabe der Daten ermöglichen.

205 Reset Content

206 Partial Content

Der Server beantwortet die Anforderung eines Teils der Daten mit GET. Die Header müssen Informationen über die Größe und Art der Daten enthalten (Range- und Content-Range-Header).

206 Partial Content

Umleitungsmeldungen – Redirection 3xx

Diese Klasse beschreibt Statuscodes, die weitere Aktionen des Clients verlangen.

300 Multiple Choices

Die Ressource enthält Informationen über mehrere Datenquellen an bestimmten Orten. Der Client wird aufgefordert, eine entsprechende Wahl zu treffen. Dieser Standard spezifiziert keine Vorschriften, die die Art und Weise der Transaktion beschreiben.

300 Multiple Choices

301 Moved Permanently

Die angeforderte Ressource wurde permanent an einen anderen Ort verschoben. Der Body der Meldung enthält die neue URL. Wenn die Anfrage GET oder HEAD war, sollte der Client dem neuen Link automatisch folgen, ansonsten erst nach einer Bestätigung durch den Nutzer.

301 Moved Permanently

302 Moved Temporarily

Die angeforderte Ressource wurde nur zeitweilig verschoben. Die neue URL wird gesendet. Wenn die Anfrage GET oder HEAD war, sollte der Client dem neuen Link automatisch folgen, ansonsten erst nach einer Bestätigung durch den Nutzer.

302 Moved Temporarily

303 See Other

303 See Other Die Anfrage kann nur von einer anderen Stelle der Ressource angefordert werden. Der Client sollte dazu die GET-Methoden verwenden. Die Meldung bedeutet nicht, dass die Ressource verschoben wurde.

304 Not Modified

304 Not Modified Das angeforderte Dokument hat sich seit der letzten Anforderung nicht geändert. Die Meldung enthält keine Daten im Body, dafür die folgenden Header-Felder:

- Date (Datum),
- Tag und/oder Content-Location, wenn die Meldung 200 bereits erfolgte,
- Expires (Verfallsdatum), Cache-Control (Speicher) und/oder Vary (Varianz), wenn es die Feldwerte erfordern.

305 Use Proxy

305 Use Proxy Die Ressource, die angefordert wurde, sollte aus dem Cache des Proxys genommen werden, der im Location-Feld angegeben wurde. Der Client sollte die Anfrage erneut unter Nutzung des Proxy starten.

Client-Fehler – Client Error 4xx

Fehler auf der Client-Seite. Der Server vermutet, dass die Ursache des Fehlers auf der Seite des Clients zu suchen ist.

400 Bad Request

400 Bad Request Die Anfrage wurde nicht verstanden. Der Client sollte die Anfrage nicht wiederholen, ohne dass Änderungen vorgenommen wurden.

401 Unauthorized

401 Unauthorized Die Anfrage erforderte eine Authentifizierung des Nutzers. Die Antwort muss ein WWW-Authenticate-Header-Feld beinhalten.

402 Payment Required

402 Payment Required Bezahlung erforderlich. Dieser Code ist eine Meldung für die Zukunft.

403 Forbidden

Die Anfrage wurde verstanden, die Ausführung ist aber verboten. Es ist nicht das Problem einer mangelhaften Autorisierung, sondern die Anfrage ist grundsätzlich nicht erlaubt. Der Client sollte die Anfrage nie wiederholen.

403 Forbidden**404 Not Found**

Die angeforderte Ressource wurde nicht gefunden. Meist bei Schreibfehlern oder gelöschten Dateien. Wurde die Ressource ohne Weiterleitung verschoben, sollte die Meldung 410 benutzt werden.

404 Not Found**405 Method Not Allowed**

Die Methode, die zur Ausführung spezifiziert wurde, ist für diese Ressource nicht erlaubt.

405 Method Not Allowed**406 Not Acceptable**

Es wurden bestimmte Charakteristiken zur Selektierung der Ressource gesendet und keine dieser Angaben führte zu einer Antwort.

406 Not Acceptable**407 Proxy Authentication Required**

Dieser Code ähnelt der Meldung 401 (Unauthorized), zeigt aber an, dass sich der Client zuerst am Proxy autorisieren muss.

407 Proxy Authentication Required**408 Request Timeout**

Der Client hat innerhalb der erwarteten oder zulässigen Wartezeit nicht reagiert.

408 Request Timeout**409 Conflict**

Durch den aktuellen Status der Ressource konnte die Anfrage nicht ausgeführt werden. Dieser Code sollte nur gesendet werden, wenn der Client in der Lage ist, darauf zu reagieren, oder der Fehler beim Client lag (Ressource falsch). Solche Zustände können bei der Verwendung der PUT-Methoden mit Programmen fremder Hersteller auftreten.

409 Conflict**410 Gone**

Die Ressource ist nicht mehr verfügbar und ein neuer Platz ist nicht bekannt.

410 Gone

411 Length Required**411 Length Required**

Bei der Anforderung fehlte die Längenangabe (Bytezahl) der Ressource.

412 Precondition Failed**412 Precondition Failed**

Die in den Request Headern übergebene Precondition ist falsch oder wurde nicht verstanden.

413 Request Entity Too Large**413 Request Entity Too Large**

Der Server meint, dass die angeforderte Ressource zu groß ist, um übertragen zu werden.

414 Request-URL Too Long**414 Request-URL Too Long**

Die URL war so lang, dass der Server sie nicht mehr bearbeiten konnte. Kommt vor allem bei sehr langen GET- und POST-Methoden vor.

415 Unsupported Media Type**415 Unsupported Media Type**

Der Typ der Ressource und der in der Anfrage genannte Typ stimmen nicht überein. Der Typ wird nicht unterstützt.

Server-Fehler – Server Error 5xx

Diese Fehler werden angezeigt, wenn der Server selbst Probleme hat, eine Anfrage auszuführen. Die zusätzlich gelieferten Informationen im Body sollte der Client anzeigen.

500 Internal Server Error**500 Internal Server Error**

Interner Server-Fehler. Es kann sich um hardwarebedingte Fehlfunktionen handeln (Speichermangel, Plattenfehler usw.).

501 Not Implemented**501 Not Implemented**

Die Anfrage konnte nicht nur für die benannte Ressource nicht ausgeführt werden, sondern wird grundsätzlich nicht unterstützt.

502 Bad Gateway**502 Bad Gateway**

Wenn der Server als Proxy oder Gateway arbeitet, zeigt er damit an, dass der übergeordnete Server fehlerhafte Daten geliefert hat.

503 Service Unavailable**503 Service Unavailable**

Der Server ist zeitweilig überlastet und kann keine weiteren Anfragen bearbeiten. Die Existenz dieses Codes bedeutet nicht, dass

der Server bei jeder Überlastung wirklich noch auf neue Anfragen reagiert.

504 Gateway Timeout

Wenn der Server als Proxy oder Gateway arbeitet, zeigt er damit an, dass der übergeordnete Server keine Daten innerhalb der erwarteten Zeitspanne geliefert hat.

**504 Gateway
Timeout**

505 HTTP Version Not Supported

Das angeforderte Protokoll wird nicht unterstützt. Die unterstützten Protokolle werden im Body gesendet.

**505 HTTP Version
Not Supported**

D An den Autor

Für Fragen, Anregungen aber auch Kritik und Hinweise steht Ihnen der Autor gern zur Verfügung. Dieses Buch soll eine solide Basis für alle sein, die in der schnell wachsenden PHP-Gemeinde Fuß fassen wollen und ein solides, deutschsprachiges Arbeitsmittel benötigen. Insofern sind Verbesserungsvorschläge und konstruktive Anmerkungen gern gesehen und finden in künftigen Auflagen sicher ihre Entsprechung.

Aktuelle Informationen finden Sie im Internet

Alle Skripte, Bugfixes und Korrekturen finden Sie im Internet unter der folgenden Adresse:

- <http://www.php.comzept.de/php4>

Mehr Informationen rund um PHP sind auf der Partnersite zum Buch zu entdecken:

- <http://www.phpwelt.de>

Den Autor selbst können Sie auf seiner Homepage näher kennenlernen:

- <http://www.joerg.krause.net>

Informationen über professionelle Unterstützung

Hilfe finden Sie beim Autor, wenn es um eines der folgenden »Probleme« geht:

- Entwicklung professioneller Websites jeder Größenordnung
- Projektmanagement und Programmierung in PHP, ASP, JavaScript, HTML usw.
- Schulungen, Seminare, Programmierkurse, Workshops
- Fachliche Unterstützung für Start-Ups, Venture Capitalists und Old Economy ;-)

Anfragen senden Sie bitte direkt per E-Mail an:

- joerg@krause.net

Leider schaffe ich es nicht immer, jede E-Mail sofort zu beantworten. Sie können aber sicher sein, dass jede E-Mail gelesen wird. Insofern können Sie dieses Medium jederzeit nutzen, um irgendetwas los zu werden – was auch immer Sie gern mitteilen möchten.