

Flash MX und PHP

**Unser Online-Tipp
für noch mehr Wissen ...**



... aktuelles Fachwissen rund
um die Uhr – zum Probelesen,
Downloaden oder auch auf Papier.

www.InformIT.de

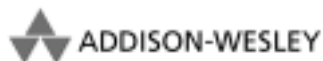
**Azad Adsay
Jürgen Lange**

Flash MX und PHP

**Flash-Anwendungen entwickeln
und dynamische Inhalte einbinden**

eBook

Die nicht autorisierte Weitergabe dieses eBooks
an Dritte ist eine Verletzung des Urheberrechts!



An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei der Deutschen Bibliothek erhältlich

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Buch erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen oder sollten als solche betrachtet werden.

Umwelthinweis:

Dieses Produkt wurde auf chlorfrei gebleichtem Papier gedruckt. Die Einschumpffolie – zum Schutz vor Verschmutzung – ist aus umweltverträglichem und recyclingfähigem PE-Material.

10 9 8 7 6 5 4 3 2 1

05 04 03 02

ISBN 3-8273-1989-7

© 2002 by Addison-Wesley Verlag,
ein Imprint der Pearson Education Deutschland GmbH
Martin-Kollar-Straße 10–12, D-81829 München/Germany
Alle Rechte vorbehalten

Einbandgestaltung:	Marco Lindenbeck, Parsdorf b. München
Lektorat:	Klaus Hofmann, khofmann@pearson.de
Korrektorat:	Petra Kienle, Fürstfeldbruck/Astrid Schürmann, Düsseldorf
Herstellung:	Anna Plenk, aplenk@pearson.de
CD Mastering:	Gregor Kopietz, gkopietz@pearson.de
Satz:	mediaService – Siegen (www.media-service.tv)
Druck und Verarbeitung:	Kösel, Kempten (www.KoeselBuch.de)

Printed in Germany

Danksagung

von Jürgen

Ich möchte neben meinen Eltern, die mir sehr früh den Umgang mit Computern und dem Internet ermöglicht haben, vor allem einer Person danken, die mir stetig Kraft gab und mich auch in den tiefsten Motivationslöchern immer wieder unterstützte: Danke Rike! Danke für alles, was du für mich getan hast! Des Weiteren möchte ich unserem Lektor Klaus Hofmann danken, der uns bei allen Fragen immer kompetent zur Seite stand. Abschließend danke ich auch den Interviewpartnern für ihr Entgegenkommen.

von Azad

Es gibt eine Reihe von Menschen, denen ich danken möchte: meinen Eltern, meinem Bruder, meinen Freunden, Jürgen, meinem Co-Autor und vielen weiteren Menschen. Ich werde darauf verzichten, alle namentlich zu nennen, nur so viel an alle Menschen aus meinem direkten Umfeld: Ich danke euch, für euer Vertrauen, eure Hilfe und euer Verständnis. Ihr habt alle einen großen Teil zu dem Buch beigetragen. Danke – ihr seid die Besten!

Inhaltsverzeichnis

E Einleitung

Über die Autoren	...11
Zielgruppe des Buches	...12
Danksagung	...12
von Jürgen	...12
von Azad	...12
Support und Website zum Buch	...13

Teil I: Einführung

1 Kurze Einführung in Flash MX

Die Vorzüge von Flash	...18
Die Flash-Werkzeuge	...20
Der normale Mauszeiger	...20
Das Linienwerkzeug	...21
Das Lassowerkzeug	...21
Das Textwerkzeug	...22
Das Ellipsen- und Rechteckwerkzeug	...24
Pinsel und Stiftwerkzeug	...24
Das Skalierwerkzeug	...24
Füllungstransformation	...24

Tintenfasswerkzeug und Farbeimerwerkzeug	...24
Die Pipette	...24
Der Radiergummi	...24
Bilder in Flash importieren	...25
Bilder dynamisch in Flash laden	...25
Flash-Filme in Flash laden	...25
Textdateien dynamisch in Flash laden	...26
Arbeiten mit Komponenten	...26
Das Prinzip der Komponenten	...27
Die Schaltflächen	...28
Combobox und Listbox	...29
Checkbox	...30
Radiobox	...31
Scrollbar einsetzen	...33
Komponenten anpassen	...34
Videos einbinden	...35
Videos importieren	...37
Video kontrollieren	...38
Qualität	...39
Videos auch für das Flash 5 Plug-In?	...42

2 PHP-Einführung

Was ist PHP?	...44
Installation mit Hilfe von PHPTriad	...44
Was ist PHPTriad?	...44
Die Installation	...45
Konfiguration von PHP und MySQL	...46
PHP in diesem Buch	...50

Guter Code	...51
Kommentare	...51
Einrückungen	...51
Eindeutige Variablennamen	...52
Quintessenz	...53
PHP und HTML	...53
Allgemeine Syntax	...53
Einbettung in HTML	...54
Variablen	...56
Arrays	...58
Indizierte Arrays	...58
Assoziative Arrays	...59
Mehrdimensionale Arrays	...60
Arrays leeren	...61
Konstanten	...62
Operatoren	...62
Zuweisungsoperatoren	...62
Arithmetische Operatoren	...63
String-Operatoren	...64
Logische Operatoren/ boolesche Werte	...64
Vergleichsoperatoren	...65
Fallunterscheidungen	...66
If-Else	...66
Switch	...68
Schleifen	...70
For-Schleifen	...70
While-Schleifen	...71
Do-While-Schleifen	...72
Funktionen	...73
Eigene Funktionen schreiben	...73
Parameter	...75
PHP-Funktionen	...77
Datumsprobleme	...78

3	PHP und externe Datenquellen	...81
	MySQL und SQL	...82
	SQL – Structured Query Language	...82
	MySQL und PHP	...85
	Das Dateisystem	...100
	Dateien anlegen und löschen	...100
	Dateien öffnen und schließen	...102
	Dateien lesen und schreiben	...103
	Übersicht der Dateifunktionen	...106
	Dateimanipulation und Verzeichnisse	...107
	PHP und FTP	...109

4	Kommunikation zwischen Flash und PHP	...111
	Textdateien einladen	...112
	Daten aus PHP-Dokumenten einladen	...112
	Sonderzeichen kodieren	...112
	Überprüfung des Ladevorgangs	...114

5	Das Cache-Verhalten steuern	...115
	Leeren des Cache	...116
	Den Cache austricksen	...117
	Den Cache mit Flash austricksen	...117
	Den Cache in PHP austricksen	...118

Teil II:

Anwendungen

...121

1 Eine Site komplett in Flash aufbauen

...123

- Das Intro ...124
- Hauptseite und Navigation ...135
- Text- und Bildelemente ein- und ausblenden, bewegen und skalieren ...143
 - Die Begrüßung ...143
 - Spot: Über mich ...144
 - Spot: Lebenslauf ...152
 - Spot: Portfolio ...158
 - Spot: Kontakt ...159

2 ECards versenden (Flash und PHP)

...163

- Der Flash-Film ...164
 - Das Storyboard ...165
 - Die Handlung ...166
 - Prozent-Preloader ...167
 - Die grafische Gestaltung der ECard ...169
- Die PHP- und Datenbankprogrammierung ...188
 - Aufbau der Datenbank ...188
 - Eingabe der Daten ...190
 - Die Vorschaumöglichkeit ...193
 - Überprüfung der Daten ...198
 - Eintragung in die Datenbank und E-Mail-Generierung ...200
 - Einfacher Passwortschutz ...202
 - Anzeige der persönlichen ECard ...203

3 Flash-Spiel mit Highscore (Flash und PHP)

...209

- Das Spiel programmieren ...210
- Der Highscore auf Datenbankbasis ...214
- Highscore-Ausgabe ...216

4 Kleines News-System (Flash und PHP)

...219

- Die Datenbank-Tabelle ...220
- Daten auslesen ...221
- Daten eintragen ...222
- Komplexes System mit erweiterter Tabelle ...224
 - Komplexes System: Daten auslesen ...224
 - Komplexes System: Daten eintragen ...226

5 Newsletter (PHP)

...229

- Mailadressen eintragen ...230
- Newsletter senden ...232
- Datenbankanbindung ...234

6 Feedback-Protokoll (Flash und PHP)

	...235
Tabelle anlegen	...236
PHP-Programmierung	...236
PHP-Dateien	...237
open.inc.php	...237
admin.php	...237
check_id.php	...238
enter.php	...239
read_db.php	...239
index.php	...241
Flash	...242
Szene 1: Preloading	...242
Szene 2: Intro	...243
Daten auslesen	...244
Einträge schreiben	...246
Administriermodus	...248

7 Texte ausleuchten (Flash und PHP)

...251

8 Bilder dynamisch mit PHP in Flash laden

...253

9 Ausblick Dreamweaver MX

	...261
Software-Review	...262
Workshop: Dreamweaver richtig konfigurieren	...264
Standard ade	...265
Site-Konfiguration	...266

Die CD zum Buch

...268

Anhang

...269

Statistiken	...269
Weblinks	...274
Website zum Buch	...274
Unsere Seiten	...274
High-End-Seiten	...274
Ressourcen	...274

Index

...276

Einleitung



Über die Autoren

Jürgen Lange, Jahrgang 1984, bereitet sich momentan auf sein Abitur vor, welches er im Juni 2003 ablegen wird. Beruflich beschäftigt er sich fast ausschließlich mit der Erstellung und Organisation von Internetprojekten. Im Dezember 2000 gründete er zusammen mit seinem Vater unter dem Namen .netzproduktion.com seine eigene Internetagentur (<http://www.netzproduktion.com>), die sich heute hauptsächlich mit der Programmierung von WebSoftware (Intranets, Onlineverwaltungssoftware, Webshops usw.) und der Erstellung datenbankgestützter Websites für Kunden in ganz Deutschland beschäftigt.

Jürgen entdeckte das Internet erst 1998 für sich. Über die macro-media.general.germany bekam er Kontakt zu Gleichgesinnten und entdeckte die Programmierung mit PHP, der er bis heute treu geblieben ist.

Er ist Initiator des Buchportals „BookWorker – der Onlineberater für IT-Bücher“ (<http://www.bookworker.de>), auf welchem er regelmäßig Bücher renommierter Verlage bespricht. Seine private Website findet man unter <http://www.smokey.de>, wo auch einige seiner Arbeiten zu sehen sind.

Sein Ziel ist es, die Agentur in absehbarer Zeit zu vergrößern und zu einer renommierten Adresse für neue, kreative und funktionale Ideen zu machen.

Azad Adsay ist seit seinem vierzehnten Lebensjahr im Internet aktiv. Zu seinen Kunden zählt er unter anderem den Axel-Springer-Verlag und WEA Records, für die er an Flash- und PHP-Lösungen arbeitete. Außerdem kooperiert er mit verschiedenen europäischen Agenturen, darunter die Pepperezak Multimedia GmbH, mit der er Projekte für Benson & Hedges und den Diogenes Verlag realisierte.

Azad ist 1984 geboren und halb kurdischer Abstammung. Er lebt in Hamburg und besucht ein Wirtschaftsgymnasium, an dem er 2004 sein Abitur machen wird.

Azad Adsay ist im Internet erreichbar unter <http://www.azadadsay.com>.

Zielgruppe des Buches

Dieses Buch soll und kann keine komplette Einführung in Flash sein. Die Autoren gehen eingangs zwar auf die Neuerungen und Highlights von Flash MX, setzen aber im Weiteren voraus, dass Sie bereits mit Flash gearbeitet haben.

Dieses Buch richtet sich in erster Linie an Leser, die mit Flash bereits Erfahrungen gemacht haben, ihre Kenntnisse vertiefen wollen, vor allem aber an der Einbindung von dynamischen Inhalten mit PHP interessiert sind.

Insofern erhalten Sie zunächst eine ausführliche Einführung in PHP und die Einbindung externer Datenquellen. Anschließend wird das Zusammenspiel von Flash und PHP genauer beleuchtet und in einer Reihe von Projektbeispielen veranschaulicht.

Support und Website zum Buch

Sollten Sie mit einem Skript oder einem Projektbeispiel aus dem Buch Probleme haben, können Sie sich gerne an die Autoren wenden. Die Website zum Buch finden Sie unter der URL <http://buch.bookworker.de>. Dort werden die aktuellsten Skripte zum Download sowie auch ein Diskussionsforum angeboten. Für Fragen zum Buch oder Projektanfragen wenden Sie sich bitte an buch@bookworker.de.

**1****Kurze Einführung in Flash MX ...17**

Die Vorzüge von Flash	...18
Die Flash-Werkzeuge	...20
Bilder in Flash importieren	...25
Flash-Filme in Flash laden	...25
Textdateien dynamisch in Flash laden	...26
Arbeiten mit Komponenten	...26
Videos einbinden	...35

**2****PHP-Einführung ...43**

Was ist PHP?	...44
Installation mit Hilfe von PHPTriad	...44
PHP in diesem Buch	...50
Guter Code	...51
PHP und HTML	...53
Variablen	...56
Arrays	...58
Konstanten	...62
Operatoren	...62
Fallunterscheidungen	...66
Schleifen	...70
Funktionen	...73
Datumsprobleme	...78

**3****PHP und externe Datenquellen ...81**

MySQL und SQL	...82
SQL – Structured Query Language	...82
Das Dateisystem	...100
PHP und FTP	...109

**4****Kommunikation zwischen Flash und PHP ...111**

Textdateien einladen	...112
Daten aus PHP-Dokumenten einladen	...112
Sonderzeichen kodieren	...112
Überprüfung des Ladevorgangs	...114

**5****Das Cache-Verhalten steuern ...115**

Leeren des Cache	...116
Den Cache austricksen	...117



Teil I: Einführung

Kurze Einführung in Flash MX

Azad Adsay

„Ändere die Welt; sie braucht es“.

Bertolt Brecht



■ ■ ■ *Abbildung 1.1: Das Logo von Flash MX*

Flash MX ist das neueste Produkt der Firma Macromedia. Es handelt sich hierbei um ein Programm, das die Vision eines Internets der vollkommenen Multimedialität darstellt. Videos, Animationen, Inhalts-Präsentationen, Sound-Wiedergabe, alles wird von Flash geboten.

Für das Buch setzen wir voraus, dass der Leser einigermaßen mit Flash umgehen kann. Zur Sicherheit folgen hier jedoch noch einmal ein paar Worte über Flash und den Umgang damit.



1.1 Die Vorzüge von Flash

Flash bietet viel mehr, als man glauben mag. ActionScript, die Flash-interne Programmiersprache, ist seit Flash Version 5 äußerst mächtig geworden. Online-Spiele und datenbankgestützte Webseiten sind nur ein Teil des durch Flash möglichen Angebots. Flash hat einen wesentlichen Vorteil, der jedoch gleichzeitig auch den größten Nachteil darstellt: das Flash Plug-In. Es sorgt dafür, dass den Benutzern immer dasselbe angezeigt wird. Egal, welchen Browser und welches Betriebssystem sie benutzen (im Idealfall zumindest). Dieser Umstand kann die Entwicklung sehr vereinfachen und penible Ausarbeitungen oder Testreihen, ob alle Systeme eine Webseite korrekt darstellen, überflüssig machen.

Doch dieser Vorzug bringt, wie gesagt, auch einen großen Nachteil mit sich. Was ist, wenn ein Benutzer kein Flash Plug-In hat? Er kann es sich zwar kostenlos bei Macromedia downloaden, aber gerade unerfahrene Internetnutzer oder Surfer mit langsamen Verbindungen werden den Download scheuen. Und dieser Aspekt lässt Kunden häufig von Flash absehen, wobei in Europa zum Beispiel rund 95% der Nutzer das Flash 4 Plug-In und rund 88% das Flash 5 Plug-In haben (mit dem Flash 5 Plug-In hat man automatisch auch das Flash 4, Flash 3 usw. Plug-In). Diese Zahlen stammen aus dem Dezember 2001. In Asien, Kanada, Amerika und Europa verfügen im Schnitt 87% aller Internetnutzer über das Flash 5 Plug-In, zum Teil auch, ohne es zu wissen, da das Plug-In standardmäßig mit dem Internet Explorer und dem Netscape Communicator installiert wird.

Flash ermöglicht jedoch noch weit mehr, wie folgendes Szenario am Beispiel von Flash MX sehr schön zeigt: Ein Autohersteller, beispielsweise Mercedes, beschließt den Relaunch seiner Website. Neben Informationen zum Unternehmen selbst soll es Online-Spiele mit Gewinnen und viele Bilder und Details zu den eigenen Produkten geben. Zudem soll die Seite voll dynamisch sein. Mit HTML ist das realisierbar, aber bei weitem nicht so schön wie mit Flash. In Flash kann der Designer ein vollständiges Interface entsprechend seiner Vorstellungen entwickeln, auch in Kombination mit Software wie Adobe Photoshop oder Macromedia Fireworks für Grafikbearbeitungen. In dieses Interface lassen sich nun dynamisch Bilder laden, welche der entsprechende Sachbearbeiter von Mercedes selbst auf den Server laden und mit einem PHP-Webinterface freischalten kann. Das neue Flash MX Feature ermöglicht im Anschluss das Laden des Bildes. Aber auch ganze Videos können jetzt komfortabel eingebunden werden. Angenommen, Sie möchten wissen, wie der Mercedes SLK aussieht. Statt vieler lebloser Bilder sehen Sie, wie der Wagen auf Sie zukommt und knapp an Ihnen vorbeifährt. Sie können den Wagen aus unterschiedlichen Perspektiven betrachten usw.

Das Ganze lässt sich natürlich auch für das Fernsehen nutzen. Sie sehen die besten Szenen aus dem Länderspiel Deutschland gegen Israel

mit Kommentaren und Einblendungen, z.B. bei einem Foul – vieles wird möglich. Hinzu kommt die kleinere Dateigröße im Vergleich zu beispielsweise MPG-Dateien.

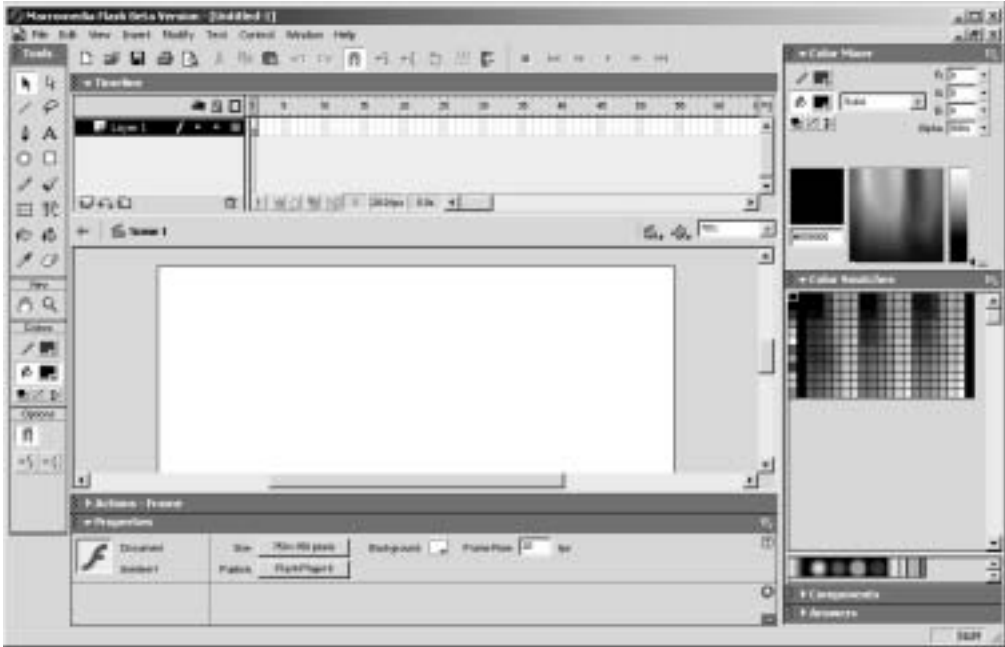
Im Mittelpunkt der Online-Spiele, ebenfalls in Flash entwickelt, stehen in unserem Beispiel neue Modelle von Mercedes. Die Informationen werden dem Kunden so auf spielerische Art und Weise übermittelt. Wer dann noch Ausdauer in diesem Spiel zeigt und besonders erfolgreich ist, der soll auch belohnt werden. Er darf seinen erzielten Score in einer Highscore-Liste eintragen und derjenige mit den meisten Punkten gewinnt. Die Entwicklung einer Highscore-Liste mit Flash, PHP und einer MySQL-Datenbank ist Thema eines späteren Kapitels.

Nun lässt sich dies jedoch fast alles auch mit HTML entwickeln. Flash vermittelt aber einen individuelleren Eindruck und durch die Komprimierungsmöglichkeiten von Bilddateien in Flash sowie die Möglichkeit, verschiedene Ebenen darzustellen, kann man eine neue, interessante Website entwickeln, die jeder Surfer gerne in Erinnerung behält.

Flash MX bietet eine Reihe von Neuerungen, die die Dynamik von Flash noch weiter fördern. Dazu gehört das dynamische Einladen von Bildern und Sound-Dateien oder auch die Möglichkeit, Videos in Flash zu importieren.

Entwicklern bietet Flash MX zusätzlich eine Steigerung des Komforts. Der Fensterwahn von Flash 5 und die Unzugänglichkeit von Flash 4 wurden behoben, so dass es jetzt ein richtiger Genuss geworden ist, mit Flash zu arbeiten.

Ein weiteres interessantes, jedoch nicht wirklich bewegendes Feature sind die Flash-UI-Komponenten. Das sind normale Checkboxes, Listen, Radiobuttons und vordefinierte Scroller, die für Entwickler jedoch nur eine Steigerung des Komforts bedeuten.



■ **Abbildung 1.2:** Das Flash MX Interface

1.2 Die Flash-Werkzeuge

In diesem Kapitel werden wir auf das Interface von Flash MX eingehen, auch im Vergleich zu Flash 5 und Flash 4. Grundlegende Begriffe werden hier erklärt; sie stellen eine Voraussetzung für das gemeinsame Arbeiten mit Flash dar.


Beginnen wir mit den Werkzeugen, die im linken Fenster von Flash angeordnet sind (Standardansicht).

Der normale Mauszeiger



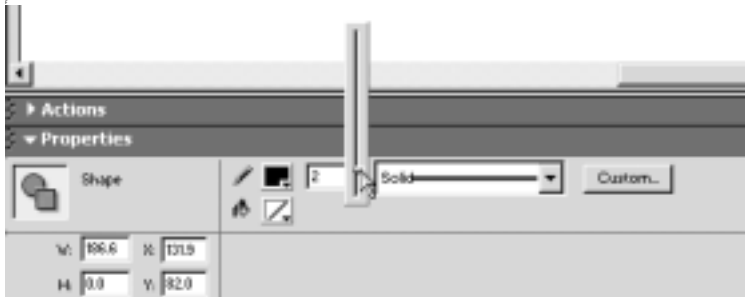
Der normale Mauszeiger ist der ganz einfache Standardcursor in Flash. Mit ihm lassen sich Linien verbiegen, markieren usw.

Das Linienwerkzeug

Mit diesem Werkzeug lassen sich Linien zeichnen. Wenn Sie beim Zeichnen die Taste  gedrückt halten, entstehen immer gerade Linien.



Eine Linie kann auch noch modifiziert werden, zum Beispiel in ihrer Breite:



■ ■ ■ *Abbildung 1.3: Modifizierung einer Linie*

Das Lassowerkzeug

Dieses Tool eignet sich sehr gut zum Markieren. Wenn Sie aus einem Vektorbild etwas ausschneiden wollen, zum Beispiel einen in Flash gezeichneten Kreis, können Sie dieses Werkzeug benutzen. Mit dem Tool kann der Bereich, der markiert werden soll, frei gezeichnet werden.



Das Lassowerkzeug ist das erste Werkzeug mit zusätzlichen Funktionen, das wir besprechen. Für uns ist vor allem der Polygon-Modus interessant (der Mauszeiger in der Abbildung zeigt auf ihn). Hiermit markieren Sie etwas durch Linien. Das heißt, Sie „stecken“ ein Gebiet ab, das markiert werden soll, um anschließend durch einen Doppelklick den gesamten Inhalt dieses Gebiets zu markieren.

Dieses Tool ist meines Erachtens deshalb so interessant, weil man mit dem normalen Lassowerkzeug häufig zuviel oder nicht geradlinig markiert.



■ ■ ■ *Abbildung 1.4:
Weitere Eigenschaften
des Lasso-Werkzeugs*

Das Textwerkzeug

A Das Textwerkzeug gehört zu den wichtigsten Werkzeugen in Flash. Mit diesem Tool zeichnet man, wie der Name schon vermuten lässt, Textfelder. Diese Textfelder lassen sich in drei Kategorien unterteilen.



Abbildung 1.5: Eigenschaften des Textfeldes

Sie sehen, die Eigenschaften der Textfelder sind recht vielfältig.

Um die einzelnen grundlegenden Funktionen zu erklären, sind in der nachfolgenden Abbildung die für uns interessanten Bereiche markiert und nummeriert.



Abbildung 1.6: Die einzelnen Eigenschaften des Textfeldes

► Die verschiedenen Arten von Textfeldern

Es gibt drei Arten von Textfeldern in Flash:

Static Text: Static Text, zu Deutsch statischer Text, ist die einfachste Form von Textfeldern. Sie kann vom Besucher der Seite nicht markiert oder geändert werden.

Dynamic Text: Die dynamischen Textfelder hingegen bieten schon etwas mehr. Sie können Variablen ausgeben, welche seit Flash 5 mit HTML formatiert werden können. Wenn Sie zum Beispiel in Flash 4 „<u>Hallo Welt!</u>“ schreiben, kommt auch genau das wieder raus. Seit Flash 5 wird das „Hallo Welt!“ unterstrichen angezeigt.

Input Text: Input-Textfelder dienen der Eingabe von Daten, wie beispielsweise in einem Kontaktformular. Input-Textfelder bekommen eine Variable zugewiesen.

► Schriftart

Hier können Sie die Schrift für das Textfeld auswählen. Sie können auch einzelne Wörter in einem Textfeld unterschiedlich formatieren.

► Schriftgröße

Hier können Sie eine beliebige Schriftgröße eintragen.

► Farbe

Die Farbe zu bestimmen ist Aufgabe dieser Funktion.

► Fett/Kursiv

Hier können Sie Wörter fett (B – Bold) oder kursiv (I – Italic) erscheinen lassen.

► Buchstabenabstände

Hier stellen Sie die Abstände der einzelnen Buchstaben ein. Die Skala reicht von - 60 (die Buchstaben liegen übereinander) bis + 60 (ein Buchstabe pro Zeile).

► Höhenstufen

Hier können Sie einstellen, ob Buchstaben oder Zahlen normal, superScript oder subScript dargestellt werden. Bei superScript sind sie hochgestellt, zum Beispiel $z\Delta$. Bei subScript sind die Buchstaben tiefgestellt.

► Ausrichtung

Hier können Sie ganz normal wie in Word die Ausrichtung des Textes festlegen.

► Link

Diese Funktion ermöglicht die Verknüpfung zu einer Webseite.

► Target

Target wird wie in HTML eingesetzt. Es bestimmt, in welchem Fenster der Link geöffnet werden soll (Beispiel: `_blank` öffnet ein neues Fenster).

Das Ellipsen- und Rechteckwerkzeug



Aufgrund der Ähnlichkeit dieser Werkzeuge werden beide zusammen erklärt. Hier zeichnen Sie simple Vektorobjekte, mit dem Ellipsenwerkzeug Kreise, mit dem Rechteckwerkzeug Vierecke.

Pinsel und Stiftwerkzeug



Auch diese Werkzeuge sind sich ziemlich ähnlich, weshalb eine Zusammenfassung nahe liegt. Mit beiden Werkzeugen ist es möglich, frei zu zeichnen. Der Stift lässt aber nur das Zeichnen mit einer Breite zu (eine nachträgliche Bearbeitung ist jedoch möglich). Für den Pinsel kann man gleich eine Breite und eine Form wählen.

Das Skalierwerkzeug



Mit diesem Werkzeug lassen sich Objekte skalieren und drehen.

Füllungstransformation



Dieses Werkzeug lässt die Manipulierung von Verläufen zu.

Tintenfasswerkzeug und Farbeimerwerkzeug



Das Tintenfasswerkzeug umrandet Vektoren, während das Farbeimerwerkzeug sie ausfüllt.

Die Pipette



Dieses Werkzeug ermöglicht es, eine Farbe herauszufinden. Klicken Sie zum Beispiel auf ein Bitmap, wird die Farbe unter dem Mauszeiger als neue aktive Farbe gewählt. Zeichnen Sie nun einen neuen Vektor, z.B.: ein Viereck, wird die Farbe verwendet, die Sie mit der Pipette ermittelt haben.

Der Radiergummi



Der Radiergummi ermöglicht das Wegradieren einzelner Vektorteile.

1.3 Bilder in Flash importieren

Das Importieren von Bilddateien in Flash ist sehr einfach. Über DATEI > IMPORTIEREN (oder auch `Strg` + `R`) lassen sich u.a. Bilder einfügen. Diese können Sie dann ganz normal in Flash einsetzen. Es empfiehlt sich, nicht allzu große Dateien zu verwenden, da die Ladezeit bei Flash-Filmen zu den häufigsten Gründen gehört, dass Besucher vorzeitig den Seitenbesuch abbrechen.

Um die beste Qualität zu erzielen, empfiehlt sich die Verwendung von BMP-Dateien.

Bilder dynamisch in Flash laden

Mit Flash 5 ging dies nur mit umständlichen PHP-Skripten. In Flash MX reicht eine Zeile. Das dynamische Einladen von Bilddateien ist in Flash MX zum Kinderspiel geworden.

```
loadMovieNum("logo.jpg", 0);
```

Mit dieser ActionScript-Zeile wird die Datei LOGO.JPG in Flash geladen. Folgendes Szenario ist vorstellbar: Der Online-Shop Müller setzt für die Webseite Flash ein und für die Aktualisierung ein PHP-Webinterface. Alle Produkte sind mit Bild und Informationen in einer Datenbank gespeichert. Aus dieser Datenbank holt PHP nun die Informationen und den Namen des Produkts, zum Beispiel LOGO.JPG. Die Daten werden an Flash weitergegeben, das Bild und die Informationen werden angezeigt, der Prozess ist beendet.

Mit Flash 5 hätte man ein Bild nehmen und mit Hilfe von PHP in eine SWF-Datei konvertieren können, um es anschließend in Flash zu laden. Jenes PHP-Skript ist komplex und der Arbeitsprozess für den Server ist umfangreicher, als wenn wir es mit Flash MX laden würden.

1.4 Flash-Filme in Flash laden

Flash bietet Ihnen die Möglichkeit, andere Flash-Filme in einen Hauptfilm zu laden. Dadurch lassen sich Ladezeiten deutlich verkürzen, indem der Besucher nur das lädt, was er sehen will – oder ein Film wird abgespielt, während ein anderer geladen wird.

Dies gestaltet sich auch sehr einfach. Ein Film wird mit dem folgenden Befehl geladen:

```
LoadMovie ("dateiname.swf", 0);
```

Der Film landet in der Hauptebene, dem Level o. Das lässt sich aber noch etwas variieren. Zum Beispiel können Sie einen Film in eine Filmsequenz laden. So bestimmen Sie beispielsweise eine genaue Position wie folgt:

```
LoadMovie ("dateiname.swf", "ziel");
```

1.5 Textdateien dynamisch in Flash laden

Das Laden von Textdateien ist in Flash äußerst einfach gestaltet. Schreiben Sie in eine Textdatei den Namen einer Variablen mit einem dazugehörigen Wert, z.B.:

```
text=Azad
```

In Flash laden Sie die Variable mit:

```
LoadVariablesNum("dateiname",0);
```

Damit ist es schon getan.

1.6 Arbeiten mit Komponenten

Flash MX bietet eine weitere Erleichterung für Entwickler: Die so genannten Flash MX-Komponenten sind eine Weiterentwicklung der SmartClips aus Flash 5. Diese Komponenten sind nichts anderes als von Macromedia entwickelte Filmsequenzen mit vordefinierten Parametern. Es gibt verschiedene Arten von Komponenten, beispielsweise Checkboxes, Comboboxen, Radiobuttons und Schaltflächen.

Neben der einfachen Integration in Flash, die mittels Drag&Drop sehr komfortabel geschieht, und der eingesparten Entwicklungszeit ist es ein weiterer Vorteil, dass die Komponenten im Nachhinein sehr leicht und bequem modifiziert werden können, und das so, dass die neuen Einstellungen für alle Komponenten gelten.

Die Eigenschaften und den Umgang mit einigen dieser Komponenten werde ich hier darlegen.



■■■ **Abbildung 1.7:**
Die Flash MX-Komponenten

Das Prinzip der Komponenten

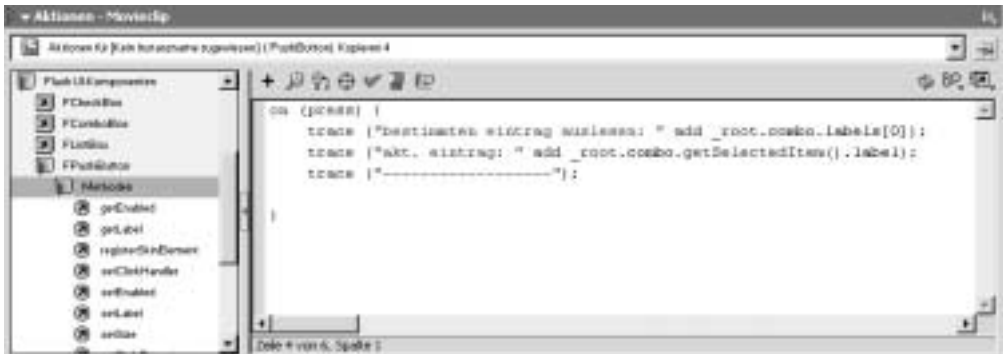
Bevor wir jedoch zu den Komponenten kommen, ein paar Informationen zum grundlegenden Umgang mit Komponenten.

Alle Komponenten sind Filmsequenzen. Wie Sie im nächsten Abschnitt erfahren werden, verhalten sich die Komponenten jedoch nicht wie gewöhnliche Filmsequenzen. Ein weiterer Unterschied ist, dass sie jetzt „Parameter“ hinzubekommen haben, die bei den normalen Objekteigenschaften stehen.



■ ■ ■ *Abbildung 1.8: Parameter einer Komponente*

Im Weiteren gibt es eine Reihe neuer Befehle für die Komponenten.



■ ■ ■ *Abbildung 1.9: Links sehen Sie die neuen Befehle, rechts sind die Befehle bereits eingesetzt.*

Neu ist zum Beispiel der Befehl `getLabel()`, welcher den Titel eines Elements anzeigt (sofern es nur einen gibt).

Weiterhin ist interessant, dass alle Objekte „deaktiviert“ werden können. Sie sind von da an für den Besucher nicht mehr nutzbar. Dieser Parameter einer jeden Komponente nennt sich „Enable“. Ändert man den Enable-Zustand eines Objekts auf „False“, ist es deaktiviert. Ändert man danach den Zustand wieder auf „True“, ist die Komponente erneut voll funktionsfähig.

```
on (press) {
    _root.list.setEnabled(false);
}
```

■ ■ ■ *Listing 1.1: Die Komponente „list“ wird deaktiviert.*

Listing 1.2: ■■■ `on (press) {`
Reaktivierung der `_root.list.setEnabled(true);`
Komponente „list“ `}`

Die Schaltflächen

Wir beginnen mit der Komponente der Schaltfläche. Es ist zwar noch nie ein Kunststück gewesen, mit Flash MX eine Schaltfläche zu entwickeln, doch die Möglichkeit, das Aussehen im Nachhinein global zu bearbeiten, ist ein Vorteil, den wir nutzen wollen.

Beginnen Sie, indem Sie eine Schaltfläche aus dem Komponentenfenster auf die Hauptbühne Ihres Films ziehen.

Der Knopf an sich ist eine simple Filmsequenz, die jedoch eine Besonderheit hat: Parameter. Jede Komponente ist um diese Eigenschaft erweitert.



■ ■ ■ **Abbildung 1.10:** Die Parameter der Schaltfläche

Im Gegensatz zu Filmsequenzen verhält sich diese Komponente aber genau wie eine Schaltfläche. Auch die Aktionen dieser Komponente verhalten sich so. Während Sie bei einer Filmsequenz den nachfolgenden Code hätten eingeben können, ohne einen Fehler zu provozieren, geht das jetzt nicht mehr.

Listing 1.3: ■■■ `onClipEvent (load) {`
Eine Aktion bei einer `trace ("Flash MX");`
Filmsequenz `}`

Ein knopftypisches ActionScript sieht hingegen so aus:

Listing 1.4: ■■■ `on (press) {`
Eine Aktion bei `trace ("Flash MX");`
einer Schaltfläche bzw. `}`
einer Schaltflächen-
Komponente

Combobox und Listbox



■ ■ ■ *Abbildung 1.11: Eine Combobox*

Eine so genannte Combobox ist eine kleine Box mit einer vordefinierten Auswahl. Es kann nur eine Option ausgewählt werden.



■ ■ ■ *Abbildung 1.12: Eine Listbox*

Eine Listbox zeigt wie eine Combobox vordefinierte Auswahlmöglichkeiten an. Im Gegensatz zu einer Combobox sind aber gleich mehrere Optionen einsehbar. Es können auch mehrere Einträge aktiviert werden.



■ ■ ■ *Abbildung 1.13: Eine Listbox mit zwei markierten Einträgen*

Der Nachteil einer Listbox ist, dass sie mehr Platz beansprucht. Jedoch ist sie für eine Mehrfachselektion die einzige Alternative.

Ich möchte diese beiden Komponenten hier besprechen, da sie sich in ihrer Entwicklung stark ähneln. Neue Einträge lassen sich mit `addItem()`; erzeugen, z.B.:

```
_root.combo.addItem("http://www.azadadsay.com");
```

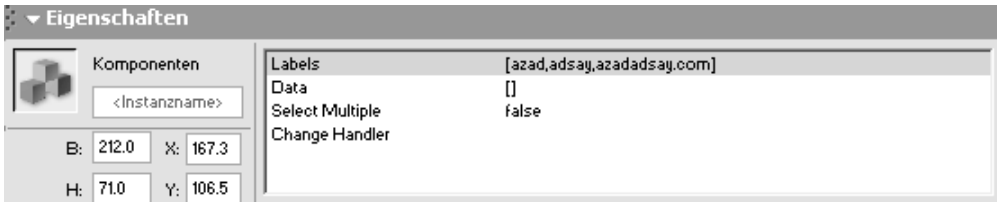
■ ■ ■ *Listing 1.5:
Neuen Eintrag erstellen*

Die Auslesemöglichkeiten ähneln sich ebenso. So können Sie mit dem Befehl `getSelectedItem()`; bei beiden Komponenten herausfinden, welcher Eintrag angezeigt wird.

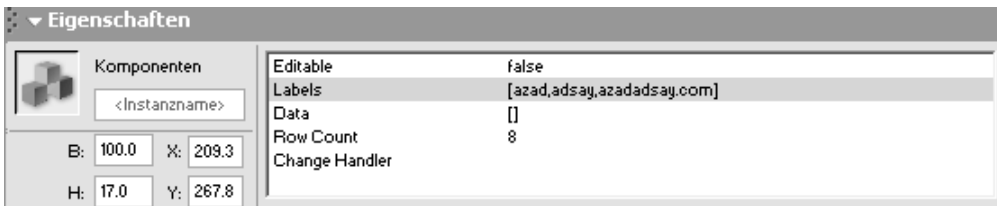
Listing 1.6: `trace ("akt. Eintrag: " + _root.list.getSelectedItemAt());`
Eintrag auslesen

Dieser Befehl ist so aber noch nicht vollständig. Es muss noch bestimmt werden, was von diesem Eintrag ausgegeben werden soll. Da es der Name, d.h. das „Label“, sein soll, muss der Befehl um `.label` erweitert werden:

Listing 1.7: `trace ("akt. Eintrag: " + _root.list.getSelectedItemAt().label);`
Bezeichnung eines vorgegebenen Eintrags ermitteln



■ **Abbildung 1.14:** Die Parameter einer Listbox



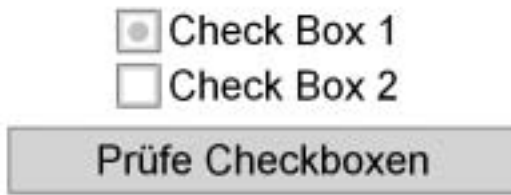
■ **Abbildung 1.15:** Die Parameter der Combobox

Auch die Parameter der beiden Komponenten ähneln sich stark. Bei beiden können Sie über den Parameter „Labels“ die Einträge etwas umständlicher, aber ohne Programmierung einfügen.

Checkbox

Wie die anderen Flash-MX-Komponenten ist auch die Checkbox aufgebaut wie ein Windows-Standard-Objekt. Checkboxes, Comboboxen etc. sind alle Bestandteile von Windows.

Checkboxes können verwendet werden, um zum Beispiel abzufra-gen, ob ein Besucher eine Bestimmung gelesen hat. Wenn dies der Fall ist und er die Bestimmung akzeptiert, muss er zuerst eine Checkbox an-wählen, um weiterzukommen.



■ ■ ■ **Abbildung 1.16:**
Zwei Checkboxen mit
einer Schaltfläche,
die obere Checkbox
ist angewählt

Die Checkboxen in Flash funktionieren ähnlich wie in HTML oder in Visual Basic. Mit dem Befehl `_root.check1.checked` lässt sich der Status einer Checkbox, hier die mit der Bezeichnung `check1`, auslesen. Ist die Checkbox markiert, wird `true` ausgegeben, andernfalls `false`.

```
on (press) {
    trace ("checkbox1 status: " + _root.check1.checked);
    trace ("checkbox2 status: " + _root.check2.checked);
}
```

■ ■ ■ **Listing 1.8:**
Die beiden Checkboxen
werden ausgelesen.

Dieses Skript löst folgende Ausgabe aus:

```
checkbox1 status: true
checkbox2 status: false
```

■ ■ ■ **Listing 1.9:**
Ausgabe in Flash

Radiobox



■ ■ ■ **Abbildung 1.17:**
Zwei Radioboxen mit einer
Schaltfläche zur Überprüfung

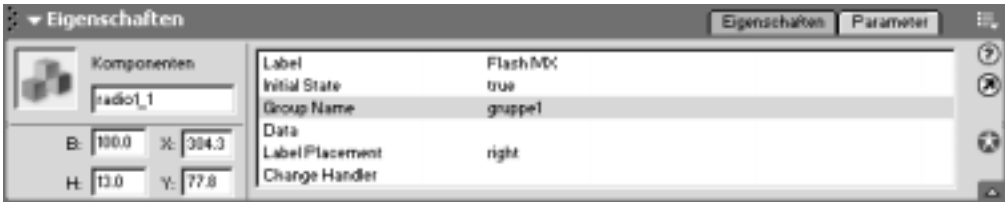
Radioboxen ähneln Checkboxen, weisen jedoch einen Unterschied auf. Wird eine Radiobox einer Gruppe aktiviert, werden alle anderen der Gruppe deaktiviert. Auch diese Komponente dürfte Ihnen aus Windows bekannt sein.

Ob eine Radiobox aktiviert ist oder nicht, erfahren Sie mit der Funktion `getState()`. Ist eine Radiobox aktiviert, wird `true` ausgegeben, andernfalls `false`, genau wie bei den Checkboxen.

```
on (press) {
    trace (_root.radio1_1.getState());
}
```

■ ■ ■ **Listing 1.10:**
Auslesen des Status
einer Radiobox

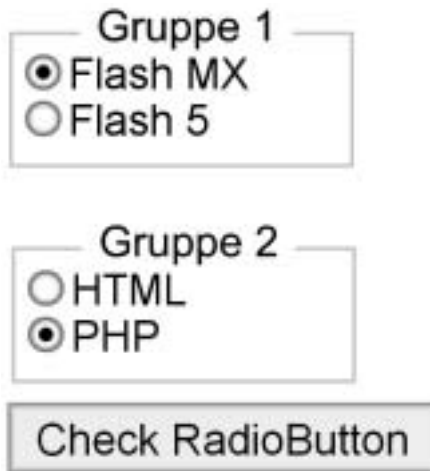
Um jedoch vier unterschiedliche Radioboxen nebeneinander laufen zu lassen, können Radioboxen gruppiert werden. Wird zum Beispiel eine Radiobox der Gruppe 1 ausgewählt, bleiben die Radioboxen der Gruppe 2 vollkommen unberührt.



■ **Abbildung 1.18:** Die Parameter einer Radiobox

Wie Sie in Abbildung 1.18 sehen können, ist diese Radiobox einer Gruppe zugeordnet. Diese Gruppe heißt „gruppe1“. Sie umfasst zwei Radioboxen, genau wie die Gruppe 2. Beide funktionieren vollkommen unabhängig voneinander.

Abbildung 1.19: ■■ Die zwei Gruppen sind hier eindeutig gekennzeichnet, beide funktionieren eigenständig.



Abschließend möchte ich noch kurz aufzeigen, wie man solch eine Gruppe u.a. auch weiterführend nutzen kann. Beispielsweise kann man eine Gruppe sperren, ähnlich wie bei einer einzelnen Radiobox:

Listing 1.11: ■■ Radioboxen sperren

```

on (press) {
    _root.gruppe2.setEnabled(false);
}
    
```

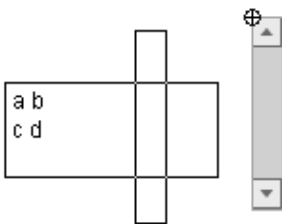
Gruppe 2 ist der Name der zweiten Gruppe. Mit diesem Skript würden beide Radioboxen gesperrt. Mit dem nachfolgenden Skript werden die Radioboxen wieder aktiviert:

```
on (press) {
    _root.gruppe2.setEnabled(true);
}
```

■■■ *Listing 1.12:*
Radioboxen wieder freigeben

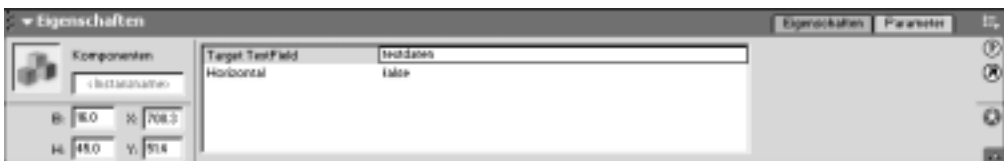
Scrollbalken einsetzen

Eine weitere Komponente sind vordefinierte Scrollbalken. Diese Scrollbalken können ohne viel Aufwand eingebaut werden und ermöglichen es, einen Text horizontal oder vertikal zu scrollen.



■■■ *Abbildung 1.20:*
Scrollbalken auf das Textfeld ziehen

Ziehen Sie einen Scrollbalken der Komponenten auf das Textfeld, danach positionieren Sie ihn wie den bereits vorhandenen Scrollbalken rechts neben das Textfeld. Durch diese kleine Methode wird der Scrollbalken direkt für dieses Textfeld eingestellt.



■■■ *Abbildung 1.21: Eigenschaften des Scrollbalkens*

Alternativ kann das Textfeld, für das der Scrollbalken verantwortlich sein soll, auch über die Eigenschaften, unter TARGET TEXTFIELD, modifiziert werden.

In die Schlüsselbildaktionen der Hauptebene muss nun noch eine Funktion eingebaut werden, die den Text des Textfeldes definiert:

```
der_text = "Hier ist Platz für einen laaangen Text, so lang, dass  
er auch ein bisschen gescrollt werden kann...";  
onData=function(){  
    textdaten.text=der_text;  
}
```

■■■ *Listing 1.13:*
Text definieren

Komponenten anpassen

Für die Komponenten können grundlegende Einstellungen vorgenommen werden, die für das ganze Projekt übernommen werden. Ein Beispiel wäre die Farbe der Markierungen für List- oder Comboboxen.

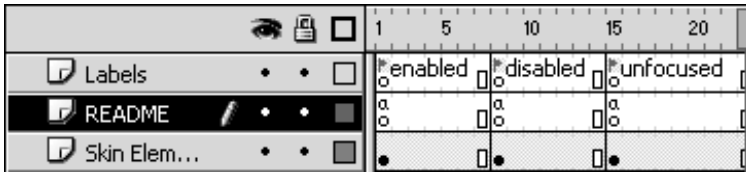
Diese Filmsequenz können Sie über die Bibliothek erreichen. Ihre Bezeichnung ist „FHighlight“, Sie finden die Sequenz unter den „Global Skins“, wie in der nachfolgenden Abbildung zu sehen:



■ ■ ■ **Abbildung 1.22: Die Flash-MX-Bibliothek**

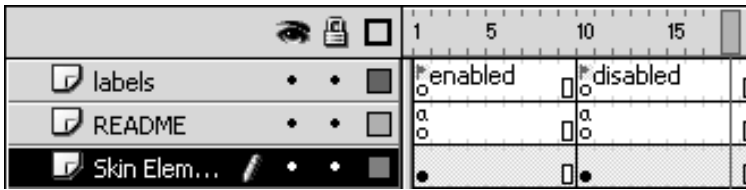
Dieser Quader kann eine beliebige Farbe oder ein beliebiges Muster bekommen.

Diese Filmsequenz hat sozusagen drei verschiedene Stadien, wobei es sich lediglich um drei verschiedene Schlüsselbilder mit verschiedenen Bezeichnungen handelt: den Status „enabled“, welcher bei ganz normalen Markierungen genutzt wird, den Status „disabled“, wenn ein markiertes Element einer Komponente aktiviert und die entsprechende Komponente im Nachhinein deaktiviert wurde. Der dritte Status, „unfocused“, wird eingesetzt, wenn ein Objekt markiert wurde und der Besucher danach zu einer anderen Komponente gewechselt hat.



■ ■ ■ **Abbildung 1.23:**
Die unterschiedlichen Stadien

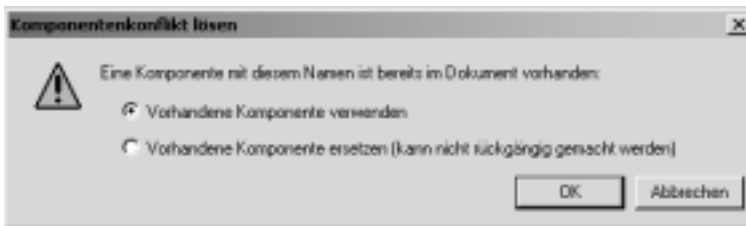
Im selben Ordner befindet sich die Filmsequenz „FBoundingBox“, die alle Hintergrundfarben für Komponenten wie die Combobox und die Listbox bestimmt. Auch diese Filmsequenz ist in zwei Stadien unterteilt.



■ ■ ■ **Abbildung 1.24:**
Die Stadien der FBoundingBox

Unter „enabled“ wird der normale Status gespeichert. Das ist die Standard-Hintergrundfarbe für die Komponenten. In „disabled“ wird die Gestaltung für gesperrte Komponenten gespeichert.

Alle Elemente können wie die beiden bereits angesprochenen Elemente modifiziert und für eigene Zwecke genutzt werden. Jedoch gibt es noch einen Punkt, den es zu beachten gilt:



■ ■ ■ **Abbildung 1.25:**
Komponentenkonflikt lösen

Wenn Sie eine Komponente hinzufügen, erscheint immer diese Abfrage. Sie müssen die erste Option dieser Abfrage „Vorhandene Komponente verwenden“ anwählen, andernfalls werden alle Komponenten zu Standardkomponenten zurückgesetzt.

1.7 Videos einbinden

Dieser Abschnitt hat zugegebenermaßen relativ wenig mit der Kommunikation von Flash und PHP zu tun. Jedoch gehört die Möglichkeit, Videos in Flash zu importieren, zu einer der interessantesten Neuerungen, für die Flash MX steht.

Für dieses Kapitel verwenden wir ein Video mit einer kleinen 3D-Sequenz. Dieses Video wurde freundlicherweise von meinem Freund Michael Lee mit Maya 3D entwickelt. Sie finden das Video auch auf der CD zum Buch.

Beachten Sie, dass Videos nur in Flash MX importiert werden können, folglich können die Videos auch nur mit dem Flash6-Plug-In angezeigt werden.



■ ■ ■ Abbildung 1.26: Das Video im Windows Media Player

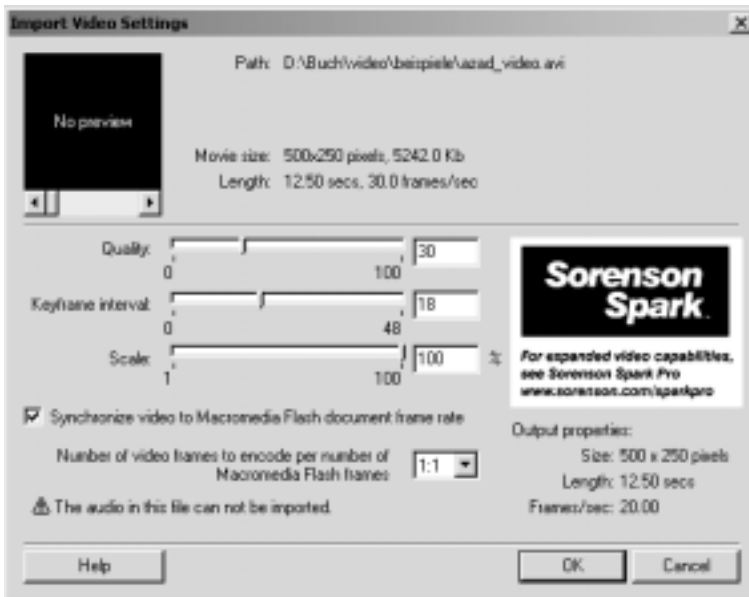
Sie benötigen für die Darstellung des Videos das Codec Plug-In namens DivX. Sie können es auf www.divx.com herunterladen, falls es nicht schon auf Ihrem System vorhanden ist.



Noch befindet sich das Video im Windows Media Player, doch das wird nicht so bleiben.

Videos importieren

Videos können Sie, wie die meisten anderen Dateien auch, ganz einfach über IMPORTIEREN im Menü DATEI in Flash importieren. Folgendes Fenster öffnet sich nun:



■ ■ ■ *Abbildung 1.27: Film in Flash importieren*

Zwei Optionen sind für uns interessant: QUALITY und SCALE. Bei QUALITY kann man die Qualität des Videos einstellen, bei SCALE lässt sich ein Anfangs-Skalierwert angeben. Wenn Sie also hier 80% wählen, wird das Video entsprechend verkleinert. Beide Optionen beeinflussen die Dateigröße und können im Nachhinein nur mit einem erneuten Import geändert werden.



■ ■ ■ *Abbildung 1.28: In Flash importiert*

Das importierte Video lässt sich in Flash selbst natürlich noch skalieren, büßt dann aber natürlich an Qualität ein.

Video kontrollieren

Das Video soll nun auch kontrolliert werden können. Das ist einfacher, als man glaubt. Es soll möglich sein, den Film zu stoppen, wieder abzuspielen, einzelne Bilder vor- und zurückzugehen, und natürlich soll man zum Anfang und zum Ende springen können.

*Abbildung 1.29: ■ ■ ■
Das Video in unserem
Player*



Legen Sie nun Symbole wie in dieser Abbildung an. Die Funktionsreihenfolge von links nach rechts: „Stopp, Abspielen, Zurück zum Beginn, Ein Bild zurück, Ein Bild vor, Zum letzten Bild“. Formatieren Sie alle Symbole in Schaltflächen, die nachfolgenden ActionScripts haben die folgenden Funktionen:

- Stoppen

```
on(press){
    stop();
}
```

- Abspielen

```
on(press) {
    play();
}
```

- Zum Anfang springen

```
on(press){
    gotoAndStop(1);
}
```

- Ein Bild zurückgehen

```
on (press) {
    prevFrame();
}
```

- Ein Bild vor gehen

```
on (press) {
    nextFrame();
}
```

- Zum letzten Bild springen

```
on (press){
    gotoAndStop(_totalframes);
}
```

Qualität

Die Qualität eines Videos ist sehr wichtig. Besuche ich eine Webseite, auf der mir ein Produkt vorgestellt wird, will ich nicht lange warten. Aber wenn das Video kaum erkennbar ist und mich an einen Film aus dem Jahr 1970 erinnert, verlasse ich auch sofort die Seite. Wie finden wir also nun den richtigen Weg, den Besucher nicht zu lange warten zu lassen, ihm aber dennoch ein qualitativ hochwertiges Video zu präsentieren?

Das Thema Qualitätsmanagement mit Videos, auch in Kombination mit Flash, könnte sicher ein eigenes Buch beanspruchen. Es reicht von der Art, wie eine 3D-Sequenz aus einem 3D-Programm exportiert wird, bis zur Darstellung, wie in Flash importiert wird. Wir werden hier aber nur eine Seite ansprechen: die Art des Imports.



■■■ *Abbildung 1.30: Die Qualität ist hier auf 30 eingestellt.*



■■■ *Abbildung 1.31: Die Qualität dieses Videos ist 100.*

Sie sehen hier die deutlichen Unterschiede der Videoqualitäten. Kann man einem Besucher die Qualität des oberen Videos zumuten? Die Größe des Videos beträgt 221 KB. Das entspricht bei ISDN einer Wartezeit von etwa 32 Sekunden. Das Video in Topqualität hat aber eine Größe von 6,46 MB erreicht und ist somit für ISDN oder gar Modemsurfer unzumutbar.

Nun muss der Film aber nicht voll skaliert sein. Wenn wir ihn verkleinern, macht sich die schlechte Qualität weniger bemerkbar.



■ ■ ■ *Abbildung 1.32: Video bei 100% x 100% -Skalierung*



■ ■ ■ *Abbildung 1.33: Video bei 80% x 80% -Skalierung*



■ ■ ■ *Abbildung 1.34: Video bei 60% x 60% -Skalierung*

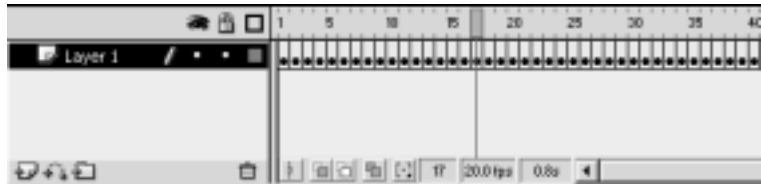
Bei allen drei Videos beträgt die Dateigröße konstant 244 KB. Die Qualität des letzten Videos ist aber am angenehmsten für den Besucher.

Videos auch für das Flash 5 Plug-In?

Mit einem kleinen Trick ist es möglich, auch Flash-5-Benutzer in den Genuss eines Videos kommen zu lassen. Dazu verlassen wir aber die Vorstellung des Videos an sich.

Importieren Sie ganz normal das Video in Flash MX. Nun exportieren Sie den Film als JPG-Sequenz. Wenn Sie in den Ordner schauen, sehen Sie eine Reihe von durchnummerierten JPG-Bildern.

Öffnen Sie in Flash MX ein neues Dokument und importieren Sie das erste Bild dieser Sequenz. Flash fragt Sie, ob es sich bei den Bildern um eine Sequenz handelt und ob alle weiteren Bilder mit eingebunden werden sollen. Bejahen Sie dies und warten Sie die Bearbeitungszeit ab. (Dieser Vorgang kann einige Zeit in Anspruch nehmen – Benutzer mit schlecht ausgerüsteten Computern sollten hier mit einer längeren Wartezeit oder gar einem Systemabsturz rechnen.)



■■■ *Abbildung 1.35: Die Zeitleiste von Flash*

Jedes Einzelbild wird in Flash MX importiert. Diese Methode hat letztendlich jedoch auch einen Nachteil: In der Regel dürfte die Dateigröße des Videos steigen.

Dennoch ist es so auch möglich geworden, Videos komfortabel für das Flash 5 Plug-In zu importieren und den Besuchern zu präsentieren. Den größten Vorteil dieser Methode stellt aber die Abwärtskompatibilität dar. Das Flash 4 Plug-In haben rund 95% aller Websurfer, es ist also eine hohe Abdeckungsquote gegeben. Und die Verbreitung des Flash MX Plug-In hat gerade erst begonnen.

PHP-Einführung

Jürgen Lange

„Die Technik, welche weder gut noch böse ist, ist ohne Bezug zur Moral. Die Moral steckt nicht in dem Hammer, sondern in dem Menschen, der ihn führt. Die Technik bedarf einer moralischen Instanz, welche eine Kontrolle über ihre Anwendung zum Nutzen des Menschen ausübt.“

Peter Bamm

In diesem Kapitel erwerben Sie grundsätzliches PHP-Wissen. Sie werden in Vorgehensweisen von der Installation bis zum Schreiben eigener Funktionen eingeführt und erhalten anhand von Beispielen einen kurzen Einblick in die praktische PHP-Programmierung.



2.1 Was ist PHP?

PHP ist im Grunde genommen nichts weiter als eine Skriptsprache zur Erstellung von dynamischen Websites. Es wurde 1994 von Rasmus Lerdorf in den ersten Versionen entwickelt und hat sich seitdem zu einer der sich am meisten und schnellsten verbreitenden Programmiersprachen im Web gemauert.

Die Funktionsweise von PHP unterscheidet sich im groben Vergleich zu HTML nur darin, dass PHP nicht wie eine HTML-Seite einfach vom Server heruntergeladen wird, sondern vor dem Download an den auf dem Server installierten PHP-Parser gesendet wird. Der Parser verarbeitet den PHP-Code dann entsprechend und gibt die gewünschten Werte zurück.

2.2 Installation mit Hilfe von PHPTriad

Die Installation von PHP stellt für viele Einsteiger eine große Hürde dar. Es gibt bereits viele Tutorials zur Installation und Konfiguration des Apache-Webservers inklusive PHP und MySQL.

PHP arbeitet natürlich nicht ausschließlich mit dem Apache-Webserver, wird aber am häufigsten in dieser Kombination genutzt (WAMP – Windows, Apache, MySQL, PHP). Auch aus diesem Grund haben wir uns entschlossen, auf diese Installation noch einmal genauer einzugehen. Über andere Installations-Tutorials informiert Sie die Info-Box.



Weitere Installationsmöglichkeiten finden Sie gesammelt unter <http://www.dynamic-webpages.de/php/installation.php>!

Einige Beispiele:

WAMP: <http://www.dynamic-webpages.de/php/install-windows95-nt.php>
– Installation auf Win95/Win98/NT

UNIX: <http://www.dynamic-webpages.de/php/install-unix.php> – Installation auf UNIX-Systemen

Was ist PHPTriad?

PHPTriad ist eine ausführbare Installationsroutine, die sowohl Apache, PHP, MySQL (einschließlich des Verwaltungstools PHPmyAdmin) und sogar Perl einfach und funktionsfähig installiert. Beachten Sie, dass PHP nicht im CGI-Modus installiert wird, was man aber zur Not noch ändern könnte. Mehr dazu erfahren Sie unter den oben angegebenen Ressourcen.

Die Installation

Zunächst laden Sie bitte die aktuellste Version von PHPTriad unter <http://sourceforge.net/projects/phptriad/> herunter. (Dort finden Sie die aktuellste Version, auch wenn sich die Projektseite an sich unter <http://www.phpgeek.com> befindet.)

Der Nachteil solcher Installationspakete ist, dass zumeist nicht die aktuellsten Versionen der einzelnen Distributionen darin enthalten sind. Demzufolge sehen Sie sich die einzelnen Versionen der enthaltenen Module genau an und updaten Sie gegebenenfalls einzelne Teile. Dazu folgende Hinweise:

PHPmyAdmin

Aktuellste Version unter

<http://phpmyadmin.sourceforge.net/>

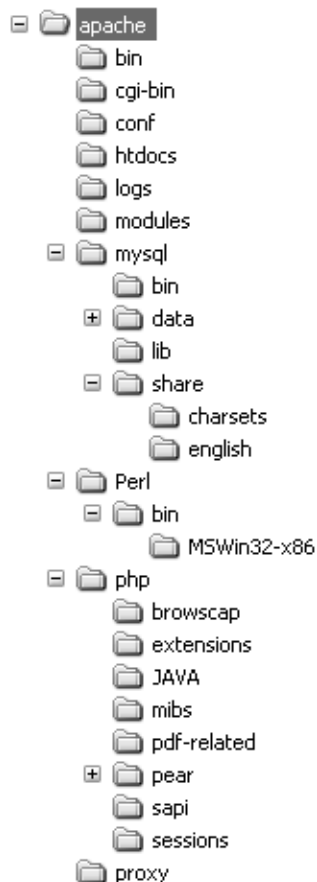
Nach Installation von PHPTriad: Einfach den Ordner entpacken und im Document-Root (siehe ► `HTTPD.CONF`) überschreiben bzw. in das Document-Root hinein kopieren.

PHP-Update

Aktuellste Version unter

http://sourceforge.net/project/showfiles.php?group_id=9325&release_id=69192

Nach Installation von PHPTriad: Downloaden und `.exe`-Datei ausführen! Damit sollten Sie vorerst auskommen. Führen Sie nun die PHPTriad.exe-Datei aus und folgen Sie den Anweisungen am Bildschirm. Die Installationsroutine erstellt einen Ordner APACHE mit folgender Struktur (siehe Abbildung 2.1).



■ ■ ■ **Abbildung 2.1:**
Ordnerstruktur nach
der Installation von
PHPTriad (Version 2.1.1)

Konfiguration von PHP und MySQL

Die für uns wichtigen Ordner sind /CONF, /HTDOCS, /PHP und /MYSQL.

► HTTPD.CONF

Beginnen wir mit dem Ordner /CONF. Darin befindet sich eine Datei namens HTTPD.CONF, die Sie nun bitte öffnen. Sie sehen sehr viel Text, der hauptsächlich Erläuterungen beinhaltet. Achten Sie nunmehr nur auf die Zeilen ohne #-Zeichen davor.



```

httpd.conf - Editor
Datei Bearbeiten Format Ansicht ?
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
ServerAdmin lange@netzproduktion.com

#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work.  The name you
# define here must be a valid DNS name for your host.  If you don't understand
# this, ask your network administrator.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# 127.0.0.1 is the TCP/IP local loop-back address, often named localhost.  Your
# machine always knows itself by this address.  If you use Apache strictly for
# local testing and development, you may use 127.0.0.1 as the server name.
ServerName localhost

#
# DocumentRoot: The directory out of which you will serve your
# documents.  By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
DocumentRoot "D:/_Projekte/"

#
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# permissions.
<Directory />
    options FollowSymLinks
    AllowOverride None
</Directory>

```

■ Abbildung 2.2: Die Apache-Konfigurationsdatei httpd.conf

Tragen Sie bei SERVER-ADMIN Ihre E-Mail-Adresse ein. Sie wird beispielsweise später beim Aufrufen einer fehlerhaften Seite bzw. einer nicht vorhandenen Seite angezeigt.

ServerAdmin lange@netzproduktion.com

Unter DOCUMENT ROOT tragen Sie den Ordner ein, welchen Sie als Projekt-Ordner für PHP-Dateien verwenden. Dies kann zum Beispiel

```
DocumentRoot "D:/_Projekte/"
```

oder auch

```
DocumentRoot "C:/"
```

sein. Der Nachteil der zweiten Variante ist, dass Sie später beim Aufruf des LOCALHOST immer den Pfad zu Ihrer entsprechenden PHP-Datei angeben müssen. Dies ist zeitaufwändiger, als wenn Sie den tiefstmöglichen Ordner (eben den Projekt-Ordner) auswählen würden. Außerdem ist es nicht sinnvoll, als Document-Root den Festplatten-Root anzugeben, da dadurch die Sicherheit bei gleichzeitiger Online-Aktivität leidet!

Der nächste Schritt besteht darin, auch die <DIRECTORY>, etwa drei Absätze weiter unten, auf das Document-Root zu aktualisieren. Tragen Sie dort also dasselbe ein wie unter DOCUMENT ROOT.

```
<Directory "D:/_Projekte/">
```

Für die Standardkonfiguration sollten diese drei Änderungen vorerst ausreichen. Speichern und schließen Sie also die Datei und öffnen Sie nun im Ordner /PHP die Datei PHP.INI.

► PHP.ini

Dies ist die Konfigurationsdatei für den PHP-Parser. Scrollen Sie langsam bis zum Absatz [SESSION].



■■■ *Abbildung 2.3: php.ini – Konfigurationsdatei für den PHP-Parser*

Dort verändern Sie den `SESSION.COOKIE_PATH` auf `C:/APACHE/PHP/SESSIONS`.

Als Nächstes legen Sie im Ordner `/PHP` eben diesen Ordner `/SESSIONS` an. Hierin werden die PHP-Session-Dateien abgelegt. Dazu aber später mehr.

Haben Sie das soweit erledigt, speichern und schließen Sie wiederum die Datei `PHP.INI`.

► **MySQL**

Abschließend konfigurieren Sie nun das MySQL-Datenbankmodul.

Sie finden im Ordner `/MYSQL/BIN` eine Datei mit dem Namen `WINMYSQLADMIN.EXE`, die Sie bitte öffnen.



■ ■ ■ *Abbildung 2.4: Die Datei winMySQLadmin.exe*

Sie werden nun nach einem Benutzernamen und einem Passwort gefragt, die Sie ab sofort als MySQL-Zugangskombination für Ihre Datenbankskripte nutzen werden.



■ ■ ■ *Abbildung 2.5: Der WinMySQLAdmin mit zugehörigem Passwort und Benutzernamen*

Als Nächstes sollten Sie, sofern noch nicht aktiv, den Server-Standalone und den MySQL-Service starten (siehe Abbildung). Klicken Sie dazu mit der rechten Maustaste und wählen Sie, wie in der Abbildung gezeigt, die Optionen START THE SERVER-STANDALONE und START THE SERVICE. Dies brauchen Sie nur einmal zu tun, da sich ab sofort MySQL schon beim Start Ihres Betriebssystems aktiviert.



■ ■ ■ **Abbildung 2.6: MySQL-Service und Standalone starten**

Abschließend verstecken Sie den WinMySQLAdmin, indem Sie die Option HIDE ME mit Hilfe der rechten Maustaste auswählen. Sie sehen nun in der Taskleiste rechts unten eine Ampel, welche grün leuchten sollte. Sollte dies nicht der Fall sein, klicken Sie im Startmenü unter PROGRAMS auf PHPTRIAD > MYSQL > MYSQL-D.

Kopieren Sie den Ordner /PHPMYADMIN aus /APACHE/HTDOCS in Ihr DOCUMENT ROOT.



■ ■ ■ **Abbildung 2.7: Ordnerstruktur im Startmenü (WinXp – PHPTriad 2.2.1) mit verschobenen „PHPmyAdmin“ und „Launch Site“**

► Der Test

Die Installationsroutine hat in Ihrem Startmenü einen Eintrag namens PROGRAMS > PHPTRIAD erstellt. Des Weiteren gibt es die Option START APACHE im Unterordner APACHE CONSOLE. Klicken Sie diese bitte an.

Es sollte sich ein neues DOS-artiges Fenster öffnen, mit dem folgendem Inhalt: „APACHE/1.3.23 <WIN23> RUNNING...“ (Windows-Systeme).

Sollte das geschehen sein, öffnen Sie als Nächstes Ihren Standard-Browser (das ist in den meisten Fällen der Internet Explorer). Geben Sie folgenden Link ein: <http://localhost/PHPmyAdmin/>. Es sollte sich Ihnen nun ein Bild ähnlich wie in der Abbildung bieten.



— Abbildung 2.8: PHPmyAdmin auf localhost

Dies ist das Zeichen, dass sowohl der Apache als auch MySQL richtig konfiguriert sind.

Öffnen Sie Ihre Skripte von nun an immer nach folgendem Schema (Apache ist gestartet!):

Im Browser öffnen: `http://localhost/+ Pfad zum Skript.`

Beispiel: `http://localhost/Ordner/Unterordner/index.php`

2.3 PHP in diesem Buch

Wir nutzen momentan PHPTriad mit PHP 4.1.1. Darauf bauen wir unsere Skripte im Buch auf. Der größte Teil funktioniert aber auch mit PHP-Releases unter PHP 4.0.

Es empfiehlt sich natürlich immer, die aktuellste PHP-Version zu nutzen. Die momentan aktuellste PHP-Version ist 4.1.2. Hierbei handelt es sich um einen Bugfix-Release, in dem unter anderem eine Sicherheitslücke in Multipart/Form-Data POST Requests geschlossen wurde.

2.4 Guter Code

Bevor wir mit dem Programmieren erster Skripten beginnen, möchten wir auf die allgemein anerkannten Konventionen für guten Code eingehen. Denn sollte es einmal vorkommen, dass Sie im Team arbeiten oder andere Leute Ihren Code lesen müssen, so erleichtert guter Code die Arbeit enorm. Kommentare, Einrückungen und eindeutige Variablenamen sind nur einige Schlagwörter in diesem Zusammenhang.

Kommentare

Kommentare sind die wichtigsten Anhaltspunkte für andere Programmierer, um Ihren Code zu verstehen. Geizen Sie nie mit Kommentaren.

Kommentare setzen Sie innerhalb der PHP-Skripte, möglichst immer mit einer Zeile Abstand, auf verschiedene Weisen. PHP kennt drei Varianten:

```
<?php
// Dies ist ein einzeiliger, am häufigsten verwendeter Kommentar

/* Dies ist ein mehrzeiliger Kommentar, welcher erst wieder durch
das entsprechende Zeichen am Ende des Kommentars beendet wird */

# Dies ist wiederum ein einzeiliger Kommentar

?>
```

■ ■ Listing 2.1:
*Kommentare für guten
Code verwenden*

Wenn Sie Kommentare setzen wollen, sollten Sie sich immer fragen:

- Was tue ich gerade?
- Wo tue ich es?
- Warum auf diese Weise?
- Warum an dieser Stelle?
- Wie wirkt sich das auf andere Programmteile aus?

Einrückungen

Einrückungen sind im Prinzip nichts anderes als die Verschachtelung zusammengehöriger Skriptmodule, die das Lesen des Codes aber wesentlich erleichtern. Das folgende Listing müssen Sie noch nicht verstehen, es soll lediglich die Verschachtelung demonstrieren.

Listing 2.2: ■■■
Demonstration des Einrückungseffekts. Innerhalb der geschweiften Klammer wird alles nach rechts versetzt eingerückt.

```
<?php
$tag = date("w");
switch ($tag) {
    case 0: echo "Sonntag"; break;
    case 1: echo "Montag"; break;
    case 7: echo "Sonntag"; break;
    case 6: echo "Samstag" ; break;
    case 5: echo "Freitag"; break;
    case 4: echo "Donnerstag" ; break;
    case 3: echo "Mittwoch"; break;
    case 2: echo "Dienstag"; break;
    default: echo "komischer Tag";
}

echo date(", d.m.Y ");

?>
```

Wie Sie sehen, ist der Effekt deutlich zu erkennen. Obwohl Sie den Code vielleicht noch nicht verstehen, wissen Sie trotzdem, dass alles innerhalb der geschweiften Klammern zur Funktion SWITCH() gehört.

Eindeutige Variablennamen

Wenn die Skripte länger werden, wächst auch die Anzahl der Variablen. In PHP müssen Sie Ihre Variablen nicht vorher deklarieren, wie es zum Beispiel in Objekt-Pascal (Delphi) nötig ist. Sie brauchen auch keine Datentypen für Ihre Variablen zu vergeben. PHP sucht sich automatisch den am besten geeigneten Datentyp aus und kann diesen auch innerhalb eines Dokuments beliebig oft ändern.

Eine Konvention bezüglich Variablen sollten Sie allerdings immer einhalten. Die Vergabe von Variablennamen gehört zu den häufigsten Aufgaben eines Programmierers, wird aber am wenigsten reflektiert.

Sie sollten Ihre Variablen wie ein Telefonbuch führen. Sie sollten immer wissen, wofür die Variable zuständig ist, sobald Sie sie nur einmal, selbst ohne zugehörigen Code, lesen. Es haben sich natürlich mehrere Schemata für die Vergabe von Variablennamen etabliert. Man kann grob unterscheiden in kurze und sprechende Variablennamen. Beispiele dafür sind:

Listing 2.3: ■■■
Verschiedene Arten von Variablennamen

```
<?php
$name_angestellter_firma_netzproduktion = "Jürgen Lange";
$name_modull = "Jürgen Lange";

?>
```

Welches System Sie letztendlich nutzen, kann Ihnen niemand vorschreiben, Sie müssen Ihren eigenen Programmierstil entwickeln und mit ihm zufrieden sein. Es gibt schließlich auch Menschen, die ein komplettes Programm in eine Zeile schreiben und damit besser klarkommen, als wenn sie nach den strengsten Konventionen programmiert hätten.

Quintessenz

Das Wichtigste, wie auch immer Sie programmieren, ist Kontinuität. Kontinuität ist die oberste Direktive, denn auch der undurchsichtigste Code wird irgendwie durchschaubar, sofern er konstant dem gleichen Stil folgt.

Des Weiteren könnten folgende Leitsätze der Programmierung Ihnen den Einstieg erleichtern:

- Machen Sie Ihren Code leicht lesbar.
- Fügen Sie, wo möglich, Kommentare hinzu.
- Wählen Sie sprechende Variablennamen.
- Unterteilen Sie Ihren Code in logische Funktionsgruppen.
- Trennen Sie separate Teile des Codes ab.
- Erstellen Sie eine Dokumentation.

2.5 PHP und HTML

Allgemeine Syntax

PHP-Skripte werden, wie Sie sicherlich schon bemerkt haben, mit bestimmten Zeichenfolgen eingeleitet und beendet. Auch hier gibt es wieder verschiedene Möglichkeiten:

► Variante 1 – XML-Stil (Extensible Markup Language)

```
<?php
// Hier kommt PHP-Code
?>
```

Diese Variante ist die am häufigsten verwendete Groß- und Kleinschreibung hinter dem ? wird nicht beachtet, sprich <?php funktioniert genauso gut wie <?PHP.

► **Variante 2 – SGML-Stil (Standard Generalized Markup Language)**

```
<?
// Hier kommt PHP-Code
?>
```

Hier gilt dasselbe wie für Variante 1, nur ohne PHP-Spezifizierung.

► **Variante 3 – ASP-Stil (Active Server Pages)**

```
<%
// Hier kommt PHP-Code
%>
```

Diese Variante setzt den Wert *ON* bei der Option `ASP_TAGS` in der PHP-Konfigurationsdatei `PHP.INI` – im Absatz *Language Options* – voraus.

► **Variante 4 – JavaScript-Stil**

```
<Skript language="PHP">
//Hier kommt PHP-Code
</Skript>
```

Die meiner Meinung nach umständlichste und deswegen auch am seltensten benutzte Variante.

Einbettung in HTML

Der große Vorteil von PHP ist, dass es sich in jeglichen HTML-Code integrieren lässt. Sie können mittels PHP HTML-Tabellen, Formulare oder Menüs erzeugen. Ein PHP-Skript lässt sich dabei ganz einfach in den HTML-Code einbetten.

Beispiel:

Listing 2.4: ■■■
*PHP-Integration in
HTML*

```
<html>
<head>
<title>Flash und PHP - Integration von PHP in HTML</title>
</head>
<body>
<?php
// Wenn eine gezogene Nummer 0 ist, hat der Kandidat eine Niete
// Wenn die gezogene Nummer 1 oder größer ist, hat der Kandidat
// einen Gewinn
?>
<?php if ($nummer == 0) { ?>
<table width=460 border=0 cellspacing=0 cellpadding=0
align=center>
<tr>
<td>
```

```

        <div align=center> Niete !</div>
    </td>
</tr>
</table>
<?php } ?>
<?php if ($nummer >= 1) { ?>
<table width=460 border=0 cellspacing=0 cellpadding=0
    align=center>
    <tr>
        <td>
            <div align=center> Gewinn !</div>
        </td>
    </tr>
</table>
<?php } ?>
</body>
</html>

```

Auch hier gibt es wieder zwei verschiedene Möglichkeiten, die jeder Programmierer für sich entdecken sollte. Man kann zum einen die PHP-Skripte direkt in den HTML-Code integrieren und vor jeder Ausgabe das Skript wieder beenden, wie im Beispiel oben, oder man erzeugt selbst die Ausgabe per PHP.

Dazu ebenfalls ein Beispiel:

```

<?php

echo"<html><head><title>Flash und PHP - Integration von PHP in
HTML</title></head>";

echo"<body>";

// Wenn eine gezogene Nummer 0 ist, hat der Kandidat eine Niete
// Wenn die gezogene Nummer 1 oder größer ist, hat der Kandidat
// einen Gewinn
if ($nummer == 0) {
echo"<table width=\"460\" border=\"0\" cellspacing=\"0\"
    cellpadding=\"0\" align=\"center\">
    <tr>
        <td>
            <div align=center> Niete !</div>
        </td>
    </tr>
</table>";
}
if ($nummer >= 1) {
echo"
<table width=460 border=0 cellspacing=0 cellpadding=0
    align=center>
    <tr>

```

■ ■ ■ *Listing 2.5:*
Gleiches Ergebnis
durch HTML-Erzeugung per PHP

```

        <td>
            <div align=center> Gewinn !</div>
        </td>
    </tr>
</table>";
}
echo"</body></html>";
?>

```

Dieses Skript liefert die gleiche Ausgabe wie das erste, sofern die Variable \$nummer identisch ist, es ergibt sich also von der Wirkungsweise her kein Unterschied.

Einige Programmierer arbeiten zur Zeitersparnis bei der Erstellung von HTML mit WYSIWYG-Editoren („What you see is what you get“), wie z.B. Dreamweaver. Dieser unterstützt jedoch nur die erste Variante.

Das Arbeiten mit beispielsweise Dreamweaver hat aber den Vorteil, dass man schnell bestimmte Tabelleneigenschaften ändern kann, was bei Variante 2 ziemlich mühselig werden kann. Wer HTML allerdings sowieso von Hand programmiert, dem wird das egal sein.

Wichtig ist noch zu beachten, dass der PHP-Interpreter nicht mit den Anführungsstrichen innerhalb von HTML zurecht kommt, d.h. Anführungsstriche wie z.B. bei

```

<table width="460" border="0" cellspacing="0" cellpadding="0"
align="center">

```

müssen dann mit einem „\“ versehen werden, damit der Parser Bescheid weiß, dass es sich um ein Sonderzeichen handelt.

```

<table width=\ "460\ " border=\ "0\ " cellspacing=\ "0\ "
cellpadding=\ "0\ " align=\ "center\ ">

```

PHP-Befehle enden immer mit einem Semikolon ;.

2.6 Variablen

Variablen verweisen auf einen bestimmten Speicherplatz im Arbeitsspeicher des Rechners, der mit beliebigem Inhalt gefüllt werden kann. In PHP werden Variablen auf ganz einfache Weise deklariert, indem man das \$-Zeichen vor eine Zeichenkette setzt. Beim Variablennamen wird nicht zwischen Groß- und Kleinschreibung unterschieden. Des Weiteren entfällt in PHP ebenfalls die explizite Typzuweisung, die beispielsweise bei Objekt-Pascal (Delphi) nötig ist. Variablen entstehen, sobald sie einem Wert zugewiesen werden. Dazu als Beispiel ein einfaches Skript:

```
<?php

$variable1 = 100;
$variable2 = 100.5;

// Addition von $variable1 und variable2
$ergebnis = $variable1 + $variable2;

// Ausgabe des Ergebnisses
echo "$ergebnis";

// Zeilenumbruch erzeugen
echo "<br>";

// Variablen mit Zeichenketten
$autor    = "Jürgen Lange<br>";
$buchtitel = "PHP und Flash";

// Verschmelzung von $autor und $buchtitel zu $ausgabe
$ausgabe = $autor.$buchtitel;

// Ausgabe der Variable $ausgabe
echo "$ausgabe";

?>
```

■■■ Listing 2.6:
Arbeiten mit Variablen

Ausgabe:

```
200.5
Jürgen Lange
PHP und Flash
```

Wie Sie sicherlich am Skript erkannt haben, werden Variablen und damit deren Inhalt nicht sofort ausgegeben. In PHP bedarf es dazu einer Funktion `echo()`, deren Anwendung im unteren Skript nochmals explizit gezeigt wird.

```
<?php

$variable = "Hallo Welt";
echo "$variable<br>";
echo("$variable");

?>
```

■■■ Listing 2.7:
Hallo Welt

Ausgabe:

```
Hallo Welt
Hallo Welt
```

Die Funktion `echo()` kann sowohl mit Klammern als auch mit einfachen Anführungsstrichen genutzt werden.

2.7 Arrays

Das Thema Arrays ist meiner Meinung nach immer ein kleiner Knackpunkt. Wer das System der Arrays verstanden hat, kann große Probleme damit lösen, wer nicht, der wird sich auf ewig um die Anwendung der Arrays drücken wollen. Aber: Ohne geht's nicht ☺!

Arrays sind Wertelisten, die verschiedene, zusammengehörige Werte enthalten. Ein gutes Beispiel für die Anwendung ist die Speicherung von Mitarbeiternamen einer Firma in einem Array.

Arrays sind gleichermaßen Variablensammlungen, in denen mehrere Variablen gleichen Typs zusammengefasst werden. Sie besitzen Dimensionen, wobei das einfachste Array nur eine Dimension hat, und können beliebig viele Werte aufnehmen.

Indizierte Arrays

Ein Beispiel für indizierte Arrays ist die vorhin angesprochene Speicherung der Mitarbeiternamen einer Firma.

Schauen Sie sich das Listing genau an, es wird Ihnen beim Verstehen des Array-Systems helfen!

Listing 2.8: ■■■
*Verschiedene Formen
des indizierten Arrays*

```
<?php

// Schreibweise der Variablenzuweisung in einem indizierten
Array
$mitarbeiter[] = "Jürgen Lange";
$mitarbeiter[] = "Martin Brochier";
$mitarbeiter[] = "Azad Adsay";
$mitarbeiter[] = "Ulrike Dittmann";

// Gleiches Ergebnis bei dieser Schreibweise
$mitarbeiter[0] = "Jürgen Lange";
$mitarbeiter[1] = "Martin Brochier";
$mitarbeiter[2] = "Azad Adsay";
$mitarbeiter[3] = "Ulrike Dittmann";

// Ausgabe für 1. Beispiel
echo"Ausgabe für <b>1. Beispiel</b><br><br>";

echo"Mitarbeiter 1: $mitarbeiter[0]<br>";
echo"Mitarbeiter 2: $mitarbeiter[1]<br>";
echo"Mitarbeiter 3: $mitarbeiter[2]<br>";
echo"Mitarbeiter 4: $mitarbeiter[3]<br><br>";

// Alternativ mit Schlüsselwort array();

$mitarbeiter = array("Jürgen Lange",
                    "Martin Brochier",
```

```

        "Azad Adsay",
        "Ulrike Dittmann");

echo"Ausgabe für <b>Schlüsselwort array()</b><br><br>";

echo"Mitarbeiter 1: $mitarbeiter[0]<br>";
echo"Mitarbeiter 2: $mitarbeiter[1]<br>";
echo"Mitarbeiter 3: $mitarbeiter[2]<br>";
echo"Mitarbeiter 4: $mitarbeiter[3]";

?>

```

Ausgabe:

```

Ausgabe für 1. Beispiel
Mitarbeiter 1: Jürgen Lange
Mitarbeiter 2: Martin Brochier
Mitarbeiter 3: Azad Adsay
Mitarbeiter 4: Ulrike Dittmann

```

Ausgabe für Schlüsselwort array():

```

Mitarbeiter 1: Jürgen Lange
Mitarbeiter 2: Martin Brochier
Mitarbeiter 3: Azad Adsay
Mitarbeiter 4: Ulrike Dittmann

```

Die Funktion `ARRAY()`; eröffnet allerdings noch weitere Möglichkeiten, beispielsweise die der Verschachtelung, mit der man Arrays von Arrays (mehrdimensionale Arrays) bilden kann usw., auf die ich etwas später noch eingehen werde.

Assoziative Arrays

Der Unterschied zum indizierten Array ist eigentlich nur der, dass Sie bei assoziativen Arrays mit Schlüsseln, also Wertepaaren (einige von Ihnen kennen das bestimmt noch aus der Schule), arbeiten.

Assoziative Arrays erlauben auch Zeichenketten als Index.

```

<?php

// erstes Beispiel
$mitarbeiter["JL"] = "Jürgen Lange";
$mitarbeiter["AA"] = "Azad Adsay";

// Ausgabe erstes Beispiel
echo "Mitarbeiter mit den Initialen JL ist:
    ".$mitarbeiter["JL"]. "<br>";
echo "Mitarbeiter mit den Initialen AA ist:
    ".$mitarbeiter["AA"];

```

Listing 2.9:
*Erzeugung und
Ausgabe von
assoziativen Arrays*

```

// zweites Beispiel
$schueler[Nummer1] = "Jürgen Lange";
$schueler[Nummer2] = "Azad Adsay";

// Umbruch erzeugen
echo "<br>";

// Ausgabe zweites Beispiel
echo "Schüler1 ist ".$schueler["Nummer1"]. "<br>";
echo "Schüler2 ist ".$schueler["Nummer2"];

?>

```

Ausgabe:

```

Mitarbeiter mit den Initialen JL ist: Jürgen Lange
Mitarbeiter mit den Initialen AA ist: Azad Adsay
Schüler1 ist Jürgen Lange
Schüler2 ist Azad Adsay

```

Hierbei ist zu beachten, dass bei der Ausgabe zwischen einer Zeichenkette (String) und dem Array selbst zu unterscheiden ist. Es muss also jedes Mal ein Anführungszeichen gesetzt werden, bevor ein assoziativer Array-Teil ausgegeben wird. Außerdem ist der Punkt (.) nötig, um die Ausgabe auszuführen.

Auch hierbei gibt es wieder die Möglichkeit, das Array mit der Funktion `ARRAY()`; zu erzeugen.

Listing 2.10: ■■■
*Assoziatives Array mit
array()*

```

<?php

$schueler = array("JL" => "Jürgen Lange",
                 "AA" => "Azad Adsay");

echo $schueler["JL"];
echo "<br>";
echo $schueler["AA"];

?>

```

Das Besondere hierbei ist der OPERATOR „=>“, der dem Index den Wert zuweist.

Mehrdimensionale Arrays

Wie einige Absätze vorher schon angesprochen, können Arrays mehrere Dimensionen haben, beispielsweise um die Namen der Schüler in Vor- und Zunamen zu unterteilen. In diesem Zusammenhang kommt die Funktion `ARRAY()`; zum Tragen.

```

<?php

$schueler = array("schueler1" =>array("Jürgen","Lange"),
                 "schueler2" =>array("Azad","Adsay"),
                 "schueler3" =>array("Martin","Brochier")
                );

// Ausgabe des Namens von Schüler 1
echo "Schüler 1: ", $schueler["schueler1"][0], " ";
echo $schueler["schueler1"][1], "<br>";

// Ausgabe des Namens von Schüler 2
echo "Schüler 2: ", $schueler["schueler2"][0], " ";
echo $schueler["schueler2"][1], "<br>";

// Ausgabe des Namens von Schüler 3
echo "Schüler 3: ", $schueler["schueler3"][0], " ";
echo $schueler["schueler3"][1], "<br>";

?>

```

■ ■ ■ *Listing 2.11:*
**Mehrdimensionales
Array**

Ausgabe:

```

Schüler 1: Jürgen Lange
Schüler 2: Azad Adsay
Schüler 3: Martin Brochier

```

Das Listing wirkt auf den ersten Blick sicherlich ein wenig umständlich und zu ausführlich, aber das Beispiel dient auch nur zur Veranschaulichung, da man normalerweise doch etwas komplexere mehrdimensionale Arrays verwendet.

Wichtig ist nur, dass diesmal statt des Punkts (.) ein Komma (,) gesetzt werden muss und dass die Ausgabe (Auswahl der Namen) über die Schreibweise

```
$array["index_assoziatives_array"]["index_indiziertes_array"]
```

erfolgt. Um beispielsweise den Vornamen des ersten Schülers und den Nachnamen des dritten Schülers zu bekommen, wäre folgende Codezeile nötig:

```

echo "Schüler 3: ", $schueler["schueler1"][0], " ";
echo $schueler["schueler3"][1], "<br>";

```

Arrays leeren

Falls Sie den Inhalt eines großen Arrays einmal während eines Prozesses nicht mehr benötigen sollten, empfehle ich Ihnen, ihn aus Performance-Gründen zu löschen.

Das Löschen eines Arrays ist über die gleiche Methode möglich, über die Sie ein neues Array erzeugen:

```
$leeres_array = array();
```

2.8 Konstanten

Konstanten sind, wie der Name schon sagt, Variablen, deren Inhalt einmalig definiert wird und dann über das Projekt nicht mehr geändert werden kann.

Konstanten werden mit Hilfe der Funktion `DEFINE()` deklariert.

*Listing 2.12: ■■■
Konstanten definieren
und ausgeben*

```
<?php
// Konstante definieren
define("Konstante", 1000);

// Konstante ausgeben
echo(Konstante);

?>
```

Ausgabe:

```
1000
```

Konstanten haben in PHP eine geringere Bedeutung, da sie sehr umständlich zu definieren sind, können aber das eine oder andere Mal ganz nützlich sein.

2.9 Operatoren

Wir haben in den vorherigen Abschnitten bereits unbewusst eine kleine Anzahl von Operatoren verwendet, beispielsweise um Variablen festzulegen oder mehrdimensionale Arrays mit Hilfe der `ARRAY()`-Funktion zu erzeugen.

Zuweisungsoperatoren

Die einfachsten Operatoren sind die Zuweisungsoperatoren. Man benötigt sie beispielsweise, um Variablen einen Wert zuzuweisen. Dazu wird das Gleichheits-Zeichen „`=`“ verwendet.

```

<?php

$neue_zahl = 0;
$zahl      = 1984;

// Zahl wird zu 0
$zahl      = $neue_zahl;

// Zahl, Zahl1, Zahl2, Zahl3 und Zahl 4 werden 1984
$zahl      = $zahl2 = $zahl3 = $zahl4 = 1984;

?>

```

■ ■ Listing 2.13:
Zuweisungsoperator
„=“

Arithmetische Operatoren

Früher oder später wird es natürlich notwendig, Zahlenwerte zu addieren, zu subtrahieren oder andere Rechenoperationen damit auszuführen. Hier kommen die arithmetischen Operatoren ins Spiel.

```

$zahl + $zahl2; // Addition

$zahl - $zahl2; // Subtraktion

$zahl / $zahl2; // Division

$zahl * $zahl2; // Multiplikation

$zahl % $zahl2; // Modulo (Rest der Ganzzahldivision)

```

Letzteres setzt voraus, dass beide Operanden ganzzahlig sind.

Des Weiteren haben Sie die Möglichkeit, Werte mit einem Befehl um eins zu erhöhen oder zu verringern. Dazu benutzen Sie die Inkrement- und Dekrementoperatoren.

```

$zahl++

$zahl--

```

Hierbei ist es entscheidend, wo die Operatoren angewendet werden, ob vor der Variablen oder danach. Ein Beispiel wird das verdeutlichen.

```

<?php

// Inkrementoperator
$x = $y++; // x wird y, danach wird y um 1 erhöht
$x = ++$y; /* y wird um 1 erhöht, danach wird x der Wert von y
zugewiesen */

// Dekrementoperator
$x = $y--; // x wird y, danach wird y um 1 verringert
$x = --$y; /* y wird um 1 verringert, danach wird x der Wert von
y zugewiesen */

?>

```

■ ■ Listing 2.14:
Wirkungsweise des
Inkrement- und
Dekrementoperators

String-Operatoren

Für Zeichenketten gibt es eine wichtige Operation: das Vereinen von Zeichenketten. Das wird, wie ebenfalls bereits angewendet, über den Punkt (.) erreicht.

Listing 2.15: ■■■
Kombination von Zeichenketten

```
<?php
$zeichenkette1 = "PHP und";
$zeichenkette2 = " Flash";

// Vereinigung der beiden Zeichenketten
echo ($zeichenkette1 . $zeichenkette2);

?>
```

Ausgabe:

PHP und Flash

Wichtig ist hierbei nur, dass nicht die Leerzeichen vor und hinter dem Punkt für den Abstand zwischen den Zeichenketten zuständig sind, sondern das Leerzeichen innerhalb der Zeichenketten!

Logische Operatoren/boolesche Werte

Die booleschen Werte („boolean“, benannt nach dem Mathematiker George Boole) unterscheiden nur zwei Zustände: Ist das Licht an oder ist das Licht aus bzw. allgemein gesagt ist etwas wahr oder falsch (TRUE oder FALSE). Solche Werte kann man auch verknüpfen, um beispielsweise abzufragen, ob ein Wert UND ein anderer Wert wahr sind. Ein zweites Beispiel wäre die Frage, ob der erste Wert NICHT wahr UND der zweite Wert WAHR ist. Für solche Abfragen verwendet man logische Operatoren (siehe Tabelle).

Operator	Bezeichnung	Beispiel	Ergebnis
&& and	Logisches UND	true && false true AND false	false
 or	Logisches ODER	false true false OR true	true
xor	Entweder oder	false xor true	true
!	Negation	!(false)	true

■■■ **Tabelle 2.1: Logische Operatoren/boolesche Werte**

Vergleichsoperatoren

Folgende Operatoren dürften jedem, der sich bereits ein wenig mit Programmierung beschäftigt hat oder sich noch dunkel an seine Schulzeit erinnert, bereits logisch erscheinen.

Vergleichsoperatoren werden am häufigsten für Fallunterscheidungen, wie z.B. eine IF-ELSE-Struktur, benötigt. Dabei werden zwei Werte miteinander verglichen. Und wie in der Grundschule gibt es hier ebenfalls Operatoren wie z.B. GRÖßER ALS, KLEINER ALS usw.

```
<?php
$gewicht = 100;

// Wenn das Gewicht größer 150 ist, dann hast du Übergewicht
if ($Gewicht >= '150') {
    echo "Du hast Übergewicht!";
}

// Alles andere ist im grünen Bereich
else {
    echo "Alles im grünen Bereich!";
}

?>
```

■ ■ ■ *Listing 2.16:*
Vergleichsoperatoren –
Beispiel

Operator	Bezeichnung	Beispiel	Ergebnis
==	gleich	10 == 10	true
===	identisch (gleicher Datentyp!)	10 === "10"	false
!=	ungleich	10 != 9	true
!==	nicht genau gleich	10 !== "10"	true
>	größer als	10 > 100	false
<	kleiner als	10 < 100	true
>=	größer gleich	10 >= 10	true
<=	kleiner gleich	10 <= 9	false

■ ■ ■ *Tabelle 2.2: Vergleichsoperatoren*

Ein Anwendungsbeispiel dazu:

```
<?php
if(10 !== "10") {
    echo"Richtig";
}
```

■ ■ ■ *Listing 2.17:*
10 ist nicht gleich "10"

```

else {
    echo "Falsch";
}

?>

```

2.10 Fallunterscheidungen

Auch Fallunterscheidungen sind bis jetzt schon einige durchgeführt worden. Beispielsweise ist das letzte Listing ebenfalls eine Fallunterscheidung, bei der überprüft wird, ob 10 nicht genau gleich „10“ ist (also inklusive Datentyp). Da dies zutrifft, wird nur der echo-Befehl „richtig“ angezeigt, nicht aber „falsch“.

Fallunterscheidungen sind essentiell für jede Programmiersprache, jeden Programmierer und jedes Programm. Kein anspruchsvolles Programm kommt ohne solche Fallunterscheidungen aus. Sie ermöglichen dynamisches Reagieren auf Benutzereinflüsse und auf sich ändernde Variablenwerte.

„WENN DIES – DANN TUE DAS – ANSONSTEN JENES“

If-Else

If-Else ist die am häufigsten verwendete Fallunterscheidung. Sie erlaubt es, mit Hilfe der gerade kennen gelernten Vergleichsoperatoren, Variablen zu vergleichen und dementsprechende Algorithmen auszuführen. Sie haben unbewusst schon sehr viele If-Else-Fallunterscheidungen in diesem Kapitel gesehen, weshalb ich nur noch kurz auf die einzelnen Strukturen eingehen möchte.

► IF

```

if (bedingung1) {
    anweisung1;
}

```

Wenn `bedingung1` zutrifft, wird die `anweisung1` ausgeführt und auch nur dann!

► ELSE

```

if (bedingung1) {
    anweisung;
}
else {
    anweisung2;
}

```

Wenn bedingung1 zutrifft, wird anweisung1 ausgeführt. Wenn bedingung1 nicht zutrifft, wird anweisung2 ausgeführt!

► ELSEIF

```
// Bedingung 1
if (bedingung1) {
    anweisung1;
}

// Bedingung 2
elseif (bedingung2) {
    anweisung2;
}

// Bedingung 3
elseif (bedingung3) {
    anweisung3;
}
// beliebig viele weitere elseif-Blöcke
...
// alles andere
else {
    anweisungN;
}
```

Wenn bedingung1 zutrifft, wird anweisung1 ausgeführt. Wenn bedingung2 zutrifft, wird anweisung2 ausgeführt. Wenn bedingung3 zutrifft, wird anweisung3 ausgeführt usw. Wenn keine der Bedingungen zutrifft, wird anweisungN ausgeführt!

Genauso wie mehrere Else-if-Blöcke nacheinander funktionieren, funktionieren auch beliebig viele If-Blöcke nacheinander. Des Weiteren lassen sich innerhalb eines solchen Blocks, mit Hilfe der logischen Operatoren, mehrere Bedingungen überprüfen. Ein Beispiel dazu:

```
<?php
$zahl1 = 100;
$zahl2 = 200;

if($zahl1 <= $zahl2 AND $zahl1 != $zahl2) {
    echo "Zahl1 ist kleiner gleich Zahl2 und doch ungleich Zahl2";
}

else {
    echo "Zahl1 ist gleich Zahl2";
}

?>
```

■ ■ Listing 2.18:
**Mehrere Bedingungen
in einem If-Block**

Ausgabe:

```
Zahl1 ist kleiner gleich Zahl2 und doch ungleich Zahl2
```

Switch

Es gibt einen Befehl, um das Unterscheiden der Fälle bei vielen möglichen Bedingungen ein wenig zu erleichtern. Die Funktion `switch()` ermöglicht es, einen Wert auf beliebig viele Optionen zu überprüfen, ohne ständig neue Blöcke schreiben zu müssen.

Listing 2.19: ■■■

*Ausgabe des
Wochentags mit Hilfe
der SWITCH()-Funktion*

```
<?php
/* $heute wird mit dem Namen des heutigen Tages auf Englisch
belegt */
$heute = date("l");

echo "$heute<br>";

// mit switch wird überprüft, welcher Tag das ist
// und dieser dann auf Deutsch ausgegeben

switch($heute) {
case "Monday" :
    echo "Montag";
    break;
case "Tuesday" :
    echo "Dienstag";
    break;
case "Wednesday" :
    echo "Mittwoch";
    break;
case "Thursday" :
    echo "Donnerstag";
    break;
case "Friday" :
    echo "Freitag";
    break;
case "Saturday" :
    echo "Samstag";
    break;
case "Sunday" :
    echo "Sonntag";
    break;
}

?>
```

Ausgabe (13.04.02):

```
Saturday
Samstag
```

Hierbei ist zu beachten, dass jede Option mit dem `BREAK;`-Befehl zu beenden ist.

Denselben Effekt hätte man natürlich auch mit If-Strukturen erreichen können. Nur der Anschaulichkeit halber dazu noch ein Beispiel:

```
<?php

// $heute wird mit dem Namen des heutigen Tages auf Englisch
belegt
$heute = date("l");

// Heutigen Tag in Englisch ausgeben
echo "$heute<br>";

// Welcher Tag ist heute?
if ($heute == "Monday") {
    echo "Montag";
}
if ($heute == "Tuesday") {
    echo "Dienstag";
}
if ($heute == "Wednesday") {
    echo "Mittwoch";
}
if ($heute == "Thursday") {
    echo "Donnerstag";
}
if ($heute == "Friday") {
    echo "Freitag";
}
if ($heute == "Saturday") {
    echo "Samstag";
}
if ($heute == "Sunday") {
    echo "Sonntag";
}

?>
```

■ ■ ■ *Listing 2.20:*
Gleicher Effekt mit
If-Strukturen

Ausgabe (13.04.02):

```
Saturday
Samstag
```

Dieses Beispiel ist weitaus umständlicher als bei Verwendung der SWITCH()-Funktion.

2.11 Schleifen

Schleifen sind ebenfalls eine wichtige Grundlage für professionelles Programmieren. Wie die Fallunterscheidungen haben auch die Schleifen wieder mit Bedingungen zu tun, nur dass sie nicht nur überprüfen, ob eine Bedingung erfüllt ist oder nicht, sondern auch dazu beitragen, dass eine Bedingung erfüllt bzw. nicht mehr erfüllt wird. Schleifen lassen bestimmte Anweisungen so oft „durchlaufen“, bis eine Bedingung erfüllt bzw. wiederum nicht mehr erfüllt ist.

For-Schleifen



■ ■ ■ *Abbildung 2.9: Ausgabe einer Zahlenfolge mit Hilfe einer For-Schleife*

Die For-Schleife ist eine der grundlegendsten Schleifen in PHP. Sie kann zum Beispiel dazu genutzt werden, eine Anweisung genau 100 Mal durchlaufen zu lassen. Die Syntax der For-Schleife sieht wie folgt aus:

```
For(Variable = Anfangswert, Bedingung, Durchlauf) {
    Anweisung;
}
```

ANFANGSWERT (INIT): die Anweisung, in der geklärt wird, von wo aus die Variable hochgezählt werden soll

BEDINGUNG: die Bedingung, die nach jedem Durchlauf auf Gültigkeit geprüft wird

DURCHLAUF: meist Inkrement- oder Dekrementoperator (siehe arithmetische Operatoren)

Deutlicher wird das Ganze anhand eines Praxisbeispiels. Bleiben wir doch gleich bei den schon angesprochenen 100 Durchläufen:

```
<?php
for($i = 0; $i <= 100; $i++) {
    echo "$i<br>";
}

?>
```

■ ■ Listing 2.21:
For-Schleife

Ausgabe:

```
1
2
3
...
```

While-Schleifen

While-Schleifen ähneln meiner Meinung nach sehr If-Else-Strukturen, da sie einen Block auch nur so lange ausführen, solange die Bedingung erfüllt ist. Aus der Syntax

```
While (Bedingung) {
    Anweisung;
}
```

lässt sich entnehmen, dass eine While-Schleife auch nicht unmittelbar (wie z.B. die For-Schleife) in die Bedingung eingreift, d.h. die Bedingungsvariable erhöht oder verringert. Das sollte immer innerhalb der Anweisung geschehen, wie Sie später in einem Beispiel sehen werden. Die Gefahr von so genannten Endlos-Schleifen ist hier sehr hoch.

Endlos-Schleifen führen zu enormer Prozessorbelastung und bringen meistens nichts als Ärger; versuchen Sie also, so etwas zu vermeiden.

Eine While-Schleife hat noch einen Vorteil. Sie muss nicht zwangsläufig einmal durchlaufen werden. Sollte die Bedingung nicht mehr erfüllt sein, wenn der PHP-Parser an die Stelle kommt, wo eine While-Schleife genau diese Bedingung verlangt, dann wird sie gnadenlos übergangen. Sie nimmt somit auch die Funktion einer Sicherungsschleife ein. Wenn Sie also ab einem bestimmten Punkt in Ihrem Programm eine Variable mit genau diesem Wert benötigen, kann eine While-Schleife Ihnen helfen. Die Anwendungsmöglichkeiten sind da wirklich komplex.

Aber nun zu einem Beispiel:

Listing 2.22: ■■■
While-Schleife

```
<?php
$i = 100;

while ($i >= 0) {

    // $I ausgeben, Zeilenumbruch
    echo "$i<br>";

    // $i verringern
    $i--;

}
?>
```

Ausgabe:

```
100
99
98
...
```

Do-While-Schleifen

Die Do-While-Schleife unterscheidet sich unter den vorliegenden Gesichtspunkten nur in einer Sache von der normalen While-Schleife: Ihre Bedingung wird erst am Ende des Blocks überprüft. Genauer gesagt heißt das, dass diese Schleife (wie z.B. die Repeat-Until-Schleife bei Objekt-Pascal – Delphi) mindestens einmal durchlaufen wird. Die Syntax lautet wie folgt:

Listing 2.23: ■■■
Do-While-Schleife

```
Do {
    Anweisung;
} while (Bedingung);
```

Dazu ebenfalls ein Beispiel:

```
<?php
// Zahl definieren
$zahl = 0;

// Schleifenbeginn
do {
    // Zahl um eins erhöhen und ausgeben (mit Umbruch)
    $zahl = $zahl + 10;
    echo"$zahl<br>";
```

```
// Schleifenende
}
//Bedingung
while ($zahl < 1000);

?>
```

2.12 Funktionen

Funktionen sind eines der interessantesten Themen von PHP. Sie sind zwar im Grunde genommen nichts anderes als eine Möglichkeit, um Ihren Code gut zu strukturieren, eröffnen aber auf der anderen Seite ungeahnte Möglichkeiten für Ihre PHP-Programmierung.

Wie schon gesagt, ist eine Funktion eine Art Code-Speicher, den Sie immer wieder in verschiedenen Teilen Ihres Programms verwenden können. Das erspart erstens das Debugging und zweitens sehr viel Schreibarbeit.

Die Syntax einer Funktion lautet wie folgt:

```
function Funktionsname([Parameter]) {
    [Kommandos]
    [return $Rückgabewert;]
}
```

Das heißt im Prinzip nichts anderes, als dass Sie die Möglichkeit haben, eine Standardroutine mit einem Parameter, also einem variablen Wert, zu füttern, um am Ende einen Rückgabewert, wiederum eine Variable, zu erhalten. In der Mathematik dürfte Ihnen jetzt der Ausdruck $f(x)$ einfallen. Und Sie haben Recht. Es ist nichts anderes.

Neben den unendlich vielen Standardfunktionen, die Ihnen PHP bereits „mitliefert“ (darunter u.a. Datenbankfunktionen, FTP-Funktionen, String- oder Array-Funktionen), können Sie natürlich auch unendlich viele eigene Funktionen schreiben. Das möchte ich Ihnen anhand eines Beispiels erläutern.

Eigene Funktionen schreiben

Für unser Beispiel denken wir uns den imaginären Problemfall, dass man eine Funktion benötigt, die eine beliebige Zahl quadriert. Das ist sicher kein schwerwiegendes Problem, aber es veranschaulicht sehr gut die Funktionsweise von Funktionen.

Listing 2.24: ■■■
**Eigene Funktionen
 schreiben**

```
<?php
// Funktion zum Quadrieren einer Zahl schreiben
// Funktionsname: QuadratErrechnen

function QuadratErrechnen($zahl) {
    $ergebnis = $zahl * $zahl;

    // Wert an Code zurückgeben
    // Achtung zurückgeben ist nicht gleich ausgeben!
    return $ergebnis;
}

// Ende Funktion QuadratErrechnen
$meinezahl = 10;

// $meinezahl an Funktion übergeben
$meinergebnis = QuadratErrechnen($meinezahl);

// Ergebnis ausgeben
echo "$meinergebnis";

?>
```

Ausgabe:

100

Beachten Sie bitte: Sobald Sie den RETURN-Befehl verwenden, wird die Funktion abgebrochen und der Wert an den Code übergeben. Folgendes Beispiel würde also nicht wie erhofft funktionieren:

Listing 2.25: ■■■
Fehlerhafte Funktion

```
<?php

function Fehlerhaft($zahl) {

    $ergebnis = $zahl * $zahl;

    return $ergebnis;

    $ergebnis = $ergebnis +10;

    return $ergebnis;
}

$zahl = 10;

$ausgabe = Fehlerhaft($zahl);

echo"$ausgabe";

?>
```

Ausgabe:

```
100
```

Hier wird das Ergebnis 100 ausgegeben, nicht das erwartete Ergebnis von 110.

Parameter

Sie können an Funktionen Parameter übergeben, die innerhalb des Funktionsblocks Verwendung finden. Schreiben Sie sie dazu einfach innerhalb der runden Klammern, wie in dem folgenden Beispiel. Mehrere Parameter werden per Komma getrennt.

```
<?php
function Addition($zahl1,$zahl2) {
    $ergebnis = $zahl1 + $zahl2;
    return $ergebnis;
}

$meinezahl1 = 2000;
$meinezahl2 = 5000;

$meinergebnis = Addition($meinezahl1,$meinezahl2);

echo "$meinergebnis";

?>
```

■■■ *Listing 2.26:*
*Mehrere Parameter in
einer Funktion*

Ausgabe:

```
7000
```

In die runden Klammern werden alle benötigten Parameter einer Funktion geschrieben. Es gibt allerdings auch optionale Parameter einer Funktion, die nicht unbedingt übergeben werden müssen. Dazu habe ich eine Funktion vorbereitet, die Ihren Text fett schreibt und eine Fehlermeldung ausgibt, wenn Sie keinen Text angegeben haben.

```
<?php
function Fett($text = "Kein Text angegeben !") {
    echo "<b>$text</b>";
}

```

■■■ *Listing 2.27:*
Funktion FETT()

```
$fettertext = Fett();  
echo "$fettertext";  
?>
```

Ausgabe:

Kein Text angegeben !

Wenn Sie jedoch einen Text angeben, wird dieser automatisch fett geschrieben:

Listing 2.28: ■■■
**Funktion FETT() zum
Zweiten**

```
<?php  
function Fett($text = "Kein Text angegeben !") {  
    echo "<b>$text</b>";  
}  
$text = "Das ist mein fett geschriebener Text!";  
$fettertext = Fett($text);  
echo "$fettertext";  
?>
```

Ausgabe:

Das ist mein fett geschriebener Text!

PHP-Funktionen

PHP beinhaltet von vornherein eine riesige Bibliothek mit Stammfunktionen für die unterschiedlichsten Zwecke. Um einen kurzen Überblick zu bekommen, wofür PHP bereits Funktionen bereithält, hier eine kurze Auflistung.

Funktionsart	Beispielfunktionen (Auswahl)
Variablenfunktionen	define(), global(), gettype()
Array-Funktionen	array(), asort(), current(), next(), shuffle(), array_count()
Zeichenkettenfunktionen	addslashes(), explode(), md5(), soundex(), htmlspecialchars(), rawurlencode()
Rechtschreibfunktionen	pspell_new(), pspell_suggest()
Funktionen für reguläre Ausdrücke	ereg(), ereg_replace(), split()
Datums-, Kalender- und Zeitfunktionen	checkdate(), date(), localtime(), time(), getdate(), jdttojulian(), juliantojd()
Mathematische Funktionen	abs(), cos(), rand(), round(), sqrt(), tan(), sin()
Dateifunktionen	dir(), closedir(), chmod(), dirname(), fopen(), fclose(), diskfreespace(), file_exists(), is_dir(), popen()
Session-Funktionen	session_destroy(), session_id(), session_start(), session_register(), session_name(), session_encode()
Netzwerkfunktionen	fsockopen(), long2ip(), checkdnsrr(), gethostbyaddr()
FTP-Funktionen	ftp_connect(), ftp_login(), ftp_mkdir(), ftp_pwd(), ftp_rename(), ftp_delete(), ftp_quit()
Verschlüsselungsfunktionen	mcrypt_get_key_size()
XML-Funktionen	xml_parser_create(), xml_set_element_handler()
PDF-Funktionen	pdf_get_info(), pdf_open(), pdf_close(), pdf_begin_page(), pdf_circle(), pdf_rotate()
Shockwave-Flash-Funktionen	swf_openfile(), swf_getframe(), swf_placeobject(), swf_fontsize(), swf_rotate(), swf_definetext()
Mail-Funktionen	mail(), imap_append(), imap_createmailbox(), imap_header()
Bildfunktionen	getimagesize(), imagearc(), imagecreate(), imagefontheight(), imagecreatefromjpeg(), imagecreatefrompng(), imagecreatefromgif()
Datenbankfunktionen	dbase_create(), dbase_open(), MySQL_connect(), MySQL_fetch_array(), MySQL_query(), msql(), odbc_do(), odbc_connect(), ocilogon(), ora_error(), pg_connect()

■ ■ ■ **Tabelle 2.3: PHPs große Vielfalt an Stammfunktionen (Auswahl des Autors)**

Diese und viele weitere Funktionen, ihren Zweck und ihre Funktionsweise finden Sie im offiziellen PHP-Manual unter <http://www.php.net>.

2.13 Datumsprobleme

Um einmal die Anwendung dieser Funktionen in der Praxis zu zeigen, habe ich das Thema Daten und Zeiten gewählt, da Ihnen dieses Gebiet sehr häufig in der Programmierung begegnen wird.

Das aktuelle Datum in verschiedenen Schreibweisen bekommt man ganz einfach mit der Funktion `DATE()`; Die Funktion `DATE()`; benötigt wiederum einige Parameter, die festlegen, in welchem Format das Datum ausgegeben werden soll. Dazu folgende Übersicht:

Parameter	Erläuterung
a	für „am“ und „pm“
A	für „AM“ und „PM“
d	für Tag des Monats mit zwei Stellen und führender Null: „01“ bis „31“
D	für Tag der Woche mit Abkürzung mit drei Buchstaben: „Sun“ usw.
F	für Monat, ausgeschrieben: „June“
h	für Stunde im 12-Stunden-Format: „01“ bis „12“
H	für Stunde im 24-Stunden-Format: „00“ bis „23“
g	für Stunde im 12-Stunden-Format ohne führende Null: „1“ bis „12“
G	für Stunde im 24-Stunden-Format ohne führende Null: „0“ bis „23“
i	für Minuten: „00“ bis „59“
j	für Tag des Monats ohne führende Null: „1“ bis „31“
l (kleines „L“)	für Wochentag voll ausgeschrieben: „Sunday“
L	für boolescher Wert, ob Schaltjahr oder nicht: „0“ oder „1“
m	für Monat mit führender Null: „01“ bis „12“
n	für Monat ohne führende Null: „1“ bis „12“
M	für Monat als Abkürzung: „Jan“
s	für Sekunden mit führender Null: „00“ bis „59“
S	für das Suffix englischer Zahlen: „th“, „nd“ oder „st“ ...
t	für Anzahl der Tage in einem Monat: „28“ bis „31“
U	für Sekunden seit Beginn der Unix-Epoche am 1.1.1970
w	für numerische Darstellung des Wochentags: „0“ (Sonntag) bis „6“ (Samstag)
Y	für vierstelliges Jahr: „2002“
y	für zweistelliges Jahr: „02“
z	für Tag im Jahr: „0“ bis „356“
Z	für die Differenz zur Zeitzone in Sekunden: „-43200“ bis „43200“ (12 Stunden)

■ **Tabelle 2.4:** Parameter der Funktion `DATE()`

Quelle: <http://www.php.net/manual/de/function.date.php>

Diese Parameter werden dann in einem Skript in die `DATE()`-Funktion eingesetzt und liefern die entsprechenden Werte zurück. Dazu verwende ich noch einmal das erste Listing aus diesem Kapitel:

```
<?php

// Aktuellen Tag - numerisch - ermitteln
$tag = date("w");

// per Fallunterscheidung ins Deutsche übersetzen und ausgeben
switch ($tag) {

    case 7: echo "Sonntag";
           break;
    case 0: echo "Sonntag";
           break;
    case 1: echo "Montag";
           break;
    case 2: echo "Dienstag";
           break;
    case 3: echo "Mittwoch";
           break;
    case 4: echo "Donnerstag" ;
           break;
    case 5: echo "Freitag";
           break;
    case 6: echo "Samstag" ;
           break;

    default: echo "komischer Tag";
}

// Datum im Format TT.MM.YYYY nach Komma ausgeben
echo date(", d.m.Y ");

?>
```

■ ■ ■ Listing 2.29:
Die `DATE()`-Funktion in
Aktion

Ausgabe (am 14.04.2002):

```
Sonntag, 14.04.2002
```




PHP und externe Datenquellen

Jürgen Lange

„Stimmt es, dass die Arbeit am Computer dumm macht?“ – „Sie haben nichts zu befürchten. Computer können einen wahn-sinnig machen, aber nicht dumm.“

Egmont Kakarot-Handtke

Nachdem Sie einen ersten, kleinen Einblick in PHP bekommen haben, schauen wir uns nun die Programmierung mit PHP etwas genauer an. Der wichtigste Aspekt hierbei ist die Interaktion mit externen Datenquellen, wie z.B. Datenbanken (insbesondere MySQL), Dateien, FTP-Server, Flash usw.

Ein wichtiger Schwerpunkt dieses Kapitels liegt hierbei auf der Erläuterung der Datenbankprogrammierung mit MySQL, mit der ich auch gleich beginnen möchte.

3.1 MySQL und SQL

MySQL hat sich zur Standard-Datenbankapplikation in Verbindung mit PHP gemausert. Die Vorteile liegen klar auf der Hand: Ein großer Funktionsumfang ermöglicht schnellen Zugriff auf schnelle Datenbanken, verbunden mit einfacher Administration, großem Leistungsumfang und Stabilität.

3.2 SQL – Structured Query Language

SQL (Structured Query Language) ist eine Programmiersprache der 4. Generation und *die* Sprache zum Aufbau, zur Verwaltung und zur Abfrage von relationalen Datenbanken. Sie wurde von IBM im Rahmen eines Forschungsprojekts entwickelt und 1987 international standardisiert. (Fast) alle Datenbanksysteme benutzen und arbeiten mit dieser Sprache.

SQL ist die wichtigste Datenbankabfragesprache und Grundlage für viele Erweiterungen, wie z.B. mSQL, PostGreSQL, MySQL usw.

Trotz der vielen Erweiterungen gibt es einen einfach zu erlernenden, stabilen Sprachkern, auf den ich hier etwas genauer eingehen möchte.

► Abfragebefehl SELECT

Die SQL-Abfrage erfolgt mit dem Befehl SELECT unter Angabe von bis zu sechs Komponenten. Die allgemeine Syntax hat die Gestalt:

```
SELECT [ALL oder DISTINCT] {spalten oder *}
FROM tabelle [alias] [tabelle [alias]] ...
[WHERE {bedingung oder unterabfrage}]
[GROUP BY spalten [HAVING {bedingung oder unterabfrage}]]
[ORDER BY spalten [ASC oder DESC]...];
```

Darunter ist Folgendes zu verstehen:

SELECT [DISTINCT] – Wähle die Werte aus der/den Spalte(n) (doppelte Datensätze nur einmal). * steht für alle Spalten ...

FROM – aus der Tabelle bzw. den Tabellen ...

WHERE – wobei die Bedingung(en) erfüllt sein soll(en) ...

GROUP BY – und gruppier die Ausgabe von allen Zeilen mit gleichem Attributwert zu einer einzigen ...

HAVING – wobei darin folgende zusätzliche Bedingung(en) gelten muss/müssen...

ORDER BY [ASC/DESC] – und sortiere nach den Spalten (auf- bzw. absteigend).

Die SQL-Befehle sind maximal 256 Zeichen lang und müssen mit Semikolon abgeschlossen werden.

Beispiele für SQL-Befehle:

```
SELECT name,straße,ort FROM adressbuch WHERE name = "Jürgen Lange";
```

```
SELECT * FROM userverwaltung WHERE sicherheitslevel = "senior";
```

Aus der Tabelle SCHÜLER sollen alle Zeilen selektiert werden, in denen der Name MÜLLER steht.

Die Umsetzung in SQL lautet: `SELECT * FROM Schüler WHERE Name = 'Müller';`

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

S_{Name = Müller}(Schüler)

SNr	Vorname	Name
12	Paul	Müller
308	Olaf	Müller

■ ■ ■ *Abbildung 3.1: Die Tabelle Schüler*

Die WHERE-Klausel liefert also die Selektion. Um zu zeigen, dass alle Spalten in der Ausgabetabelle angezeigt werden sollen, wird ein Joker – das Stern-Symbol „*“ – verwendet.

Nun sollen aus der Tabelle SCHÜLER alle Zeilen selektiert werden, in denen der Name "Müller" steht und deren Vorname mit "O" beginnt.

Die Umsetzung in SQL lautet: `SELECT * FROM Schüler WHERE Name = 'Müller' AND Vorname LIKE 'O*';`

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

S_{Name = Müller}(Schüler)

SNr	Vorname	Name
308	Olaf	Müller

■ ■ ■ *Abbildung 3.2: Die Tabelle Schüler mit der zweiten Select-Abfrage*

► NOT – AND – BETWEEN – ORDER BY – LIMIT – OR

Dies alles sind logisch klingende (Teil-)Befehle in SQL. Der Programmierer selektiert die Daten am besten immer so, wie er es (auf Englisch) aussprechen würde:

ZEIGE name,ort,alter VON tabelle WO name = "Jürgen" ODER name = "Azad" UND alter = "18" UND id ZWISCHEN 1 und 200 SORTIERE NACH absteigend BEGRENZE von 0 an auf 2;

In SQL (grob) übersetzt würde das lauten:

```
SELECT name,ort,alter FROM tabelle WHERE name = "Jürgen" OR name = "Azad" AND alter = "18" AND id BETWEEN 1 AND 200 ORDER BY desc LIMIT 0,2;
```

(Anmerkung: Dieser String läuft nicht, er soll nur veranschaulichen, was alles möglich ist).

► DELETE

Um eine Tabelle zu leeren, wendet man den Befehl DELETE an. Beispiele:

```
DELETE * FROM tabelle WHERE id = 10
```

> Löscht die Zeile, wo die ID-Nummer gleich 10 ist.

```
DELETE * FROM tabelle
```

> Leert die Tabelle tabelle.

► UPDATE

Wie der Name schon sagt, kann man mit dem Befehl UPDATE einzelne Datenbankeinträge aktualisieren und ändern. Die Syntax hierbei ist etwas anders als die von SELECT und DELETE:

```
UPDATE tabelle SET spalte = <formel> WHERE id = 10
```

Dieser String ersetzt den Eintrag in der Spalte spalte in Zeile 10 durch die <formel>.

► DISTINCT

DISTINCT ermöglicht es, doppelte Einträge nicht zu berücksichtigen und entsprechende Inhalte nur einmal auszugeben. Das ist beispielsweise bei einer Suche nach einem Autor in einer Verlagsdatenbank sehr erfreulich, wenn man nur wissen möchte, ob dieser Autor für den Verlag geschrieben hat oder nicht. Eine weitere Anwendungsmöglichkeit ist das Adressbuch, bei dem mehrere Leute die gleiche Telefonnummer haben. DISTINCT erlaubt es ohne Probleme, die Telefonnummer einmalig auszugeben.

```
SELECT DISTINCT telefonnummer FROM adressbuch WHERE nummer = "01805-4345"
```

► LIKE

Um ähnlich klingende Einträge aus einer Datenbank zu ziehen, benutzt man in SQL den Befehl LIKE. Er ermöglicht durch einige optionale Parameter eine spaltenweise Suche nach bestimmten Wörtern oder Wortstämmen.

```
SELECT stadt FROM deutschland WHERE city LIKE "%Berlin%"
```

Diese Abfrage liefert Städtenamen aus der Tabelle Deutschland, die ähnlich wie Berlin klingen.

- % steht für beliebig viele Zeichen ohne irgendwelche Einschränkung:
 - A% – alle Worte, die mit A beginnen (Alabama, Andromeda)
 - %flug – alle Worte, die auf -flug enden (Rundflug, Hinflug)
- [A-E]% – steht für alle Worte, die mit den Buchstaben A bis E beginnen.
 - [A]n% – alle Worte, die mit An- anfangen
 - [A-EG]% – Worte, die mit A, B, C, D, E oder G anfangen
- [^U]% – steht für Worte, die nicht mit U beginnen.
- _ (Unterstrich) steht für genau ein beliebiges Zeichen.

MySQL und PHP

MySQL ist eine spezielle Weiterentwicklung von SQL, gleicht aber in den Grundfunktionen der Syntax von SQL. Das machen wir uns in diesem Abschnitt des Kapitels zunutze, um die ersten Datenbankverbindungen zu erstellen.

PHP hält für uns mehrere Datenbankfunktionen (siehe PHP-Einführung, Funktionen) bereit. Die wichtigsten sind:

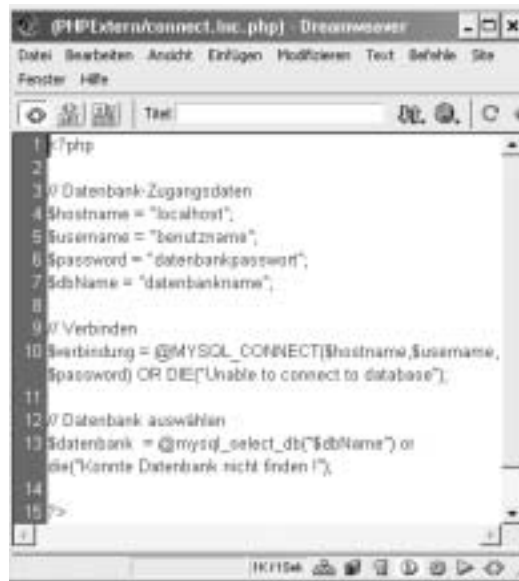
- MySQL_connect()
- MySQL_select_db()
- MySQL_query()
- MySQL_fetch_row()
- MySQL_fetch_array()
- MySQL_close()

Diese Auflistung entspricht auch ungefähr der Anwendungsreihenfolge der Funktionen bei einem Datenbank-Connect mit anschließendem Datentransfer.

► connect.inc.php

Zuallererst schreiben wir eine Datei, in der die Daten stehen, welche immer wieder zum Aufbau der Datenbankverbindung benötigt werden. Der Hintergrund ist einfach: Stellen Sie sich vor, Sie schreiben mehrere Skripte, die eine Verbindung zur selben Datenbank benötigen. Sie tragen überall Ihre Verbindungsdaten (beispielsweise das Passwort oder den Benutzernamen) einzeln in die Skripte ein. Sicherlich funktioniert das, aber nur bis zu dem Zeitpunkt, an dem sich Ihre Daten ändern, weil Sie beispielsweise zu einem anderen Provider wechseln. Dann müssten Sie alle Dateien einzeln

bearbeiten, um die Verbindung wieder herstellen zu können (mal abgesehen davon, dass öfter sogar einzelne Dateien vergessen werden)!



■ ■ ■ **Abbildung 3.3:** *connect.inc.php*

Wenn Sie aber ein einzelnes Verbindungsskript schreiben, das diese Daten enthält und dann dieses eine Skript in jedes andere einbinden (in PHP gibt es dafür zwei Funktionen – `require()`; und `include()`), haben Sie so die Möglichkeit, alle benötigten Dateien auf einmal zu manipulieren. Wir nennen diese Datei `connect.inc.php`. Erstellen Sie also eine neue Datei (beispielsweise mit dem Windows-Editor oder Vergleichbarem) und geben Sie folgenden Code ein.

Listing 3.1: ■ ■ ■
connect.inc.php

```

<?php

// Datenbank-Zugangsdaten
$hostname = "localhost";
$username = "benutzername";
$password = "datenbankpasswort";
$dbName = "datenbankname";

// Verbinden
$verbindung = @MYSQL_CONNECT($hostname,$username,$password) OR
DIE("Unable to connect to database");

// Datenbank auswählen
$datenbank = @MySQL_select_db("$dbName") or die("Konnte
Datenbank nicht finden!");

?>
  
```

In den folgenden Dateien, die einen Datenbank-Connect benötigen, brauchen Sie nach dem Einschließen der `CONNECT.INC.PHP` nur noch den SQL-Query an die Datenbank zu senden und schon können Sie mit den Daten arbeiten (mehr dazu später).

► Testdatenbank anlegen

Wir legen uns zu Testzwecken nun eine Datenbank an, mit der wir später arbeiten werden. Es soll ein kleines Adressbuch für private Zwecke werden. Die Tabelle soll folgende Spalten enthalten:

- Name
- Vorname
- Straße
- PLZ
- Ort
- Telefon
- Fax
- E-Mail

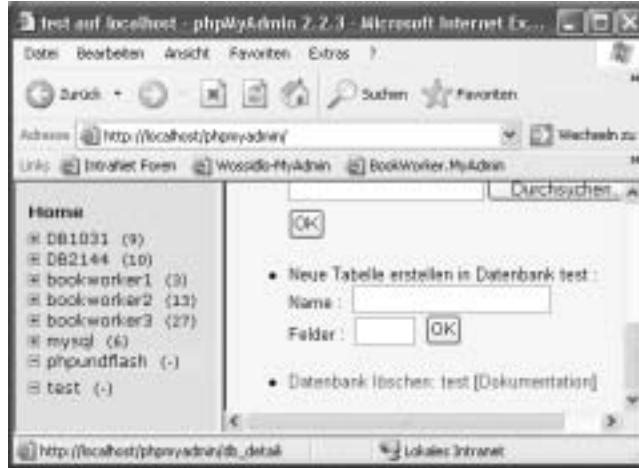
Vorerst brauchen wir nicht mehr als ca. fünf Einträge, die wir gleich in die Tabelle eingeben. Zuerst müssen wir sie aber natürlich erzeugen.

- 1 Sofern Sie es noch nicht getan haben, starten Sie bitte Ihren Apache-Webserver und den PHPmyAdmin (Hinweise dazu im Kapitel „PHP-Einführung – Installation“).

■ **Abbildung 3.4:**
Gestarteter
PHPmyAdmin 2.2.3

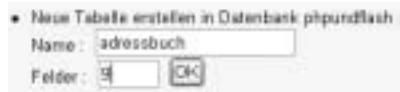


Erzeugen Sie nun eine neue Datenbank mit dem Namen „phpundFlash“.



■ ■ ■ **Abbildung 3.5: Datenbank wurde erzeugt**

- 2 Erzeugen Sie eine Tabelle mit dem Namen „adressbuch“. Sie benötigen gemäß unserer Vorgaben acht Felder (Spalten) plus ein Feld für den künstlichen Schlüssel („ID“).



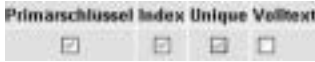
■ ■ ■ **Abbildung 3.6: Neue Tabelle erzeugen**

- 3 Legen Sie nun die Eigenschaften der Felder gemäß der Abbildung 3.7 fest.

Feld	Typ	Länge/Set*	Attribute	Null	Standard	Extra
id	INT	255		not null		auto_increment
name	TINYTEXT			not null		
vorname	TINYTEXT			not null		
strasse	TINYTEXT			not null		
ort	TINYTEXT			not null		
plz	TINYTEXT			not null		
telefon	TINYTEXT			not null		
fax	TINYTEXT			not null		
email	TINYTEXT			not null		

■ ■ ■ **Abbildung 3.7: Tabelleneigenschaften**

- 4 Von Bedeutung ist auch, dass Sie für die erste Spalte („ID“) das AUTO_INCREMENT und die Häkchen gemäß der Abbildung 3.8 setzen.



■ ■ ■ *Abbildung 3.8: Eigenschaften der Spalte ID*

- 5 Speichern Sie die Tabelle.
Alternativ dazu können Sie die Tabellenstruktur auch der Datei ADRESSBUCH.SQL auf der CD entnehmen.

► **Testdatenbank füllen**

- 1 Um nun ca. fünf Einträge in die Tabelle zu schreiben, klicken Sie nach dem Erzeugen der Tabelle auf die Option EINFÜGEN.



■ ■ ■ *Abbildung 3.9: Einträge einfügen*

- 2 Geben Sie folgende Angaben in die Tabelle ein:

ID	Name	Vorname	Straße	Ort	PLZ	Telefon	Fax	Email
1	Muster-mann	Max	Muster-straße 2	Musterort	12345	0123/ 456789	01234/ 456788	max@muster-mann.de
2	Klausen	Klaus	Zur Klausen 4	Klausdorf	54321	3210/ 987654	43210/ 887654	
3	Klaus	Hans	Berliner Ring 10	Saal	65425			
4	Mann	Daniel	Franz-burger 1a	Rostock	89745	8525/ 456852	8525/ 456851	dan@manns.de
5	Lange	Jürgen	Neuendorfer Str.1a	Saal	18317			buch@book-worker.de

■ ■ ■ *Tabelle 3.1: Einträge für die Tabelle „adressbuch“*

Um neue Datensätze einzufügen, aktivieren Sie vor dem Speichern des ersten Datensatzes die Option NEUEN DATENSATZ EINFÜGEN (siehe Abbildung 3.10).



■ ■ ■ *Abbildung 3.10: Speichern und neuen Datensatz einfügen*

► Testdatenbank auslesen

Nun geht's ans Eingemachte! Nachdem die Datenbank-Verbindungsdaten in der `CONNECT.INC.PHP` bereits eingetragen sind (Achtung! Alle Daten müssen stimmen, da ansonsten kein Connect zustande kommt!), brauchen Sie die `CONNECT.INC.PHP` nur noch per `include` in jede neue Datei (eine erstellen Sie jetzt bitte) einzubinden.

Listing 3.2: `connect.inc.php` in Datei einbinden

```
<?php
include("connect.inc.php");

?>
```

Wir möchten jetzt die Nachnamen aller in der Datenbanktabelle verzeichneten Adressbucheinträge auflisten und diese, sofern vorhanden, mit einem E-Mail-Link verbinden.

Listing 3.3: Daten aus der Datenbank auslesen und im Skript verarbeiten

```
<?php
// Datenbank-Connect
include("connect.inc.php");

?><?php

// SQL-Query an die Datenbank senden
// mit Hilfe von MySQL_query()
// das @-Zeichen unterdrückt Fehlermeldungen

$result = @MySQL_query("SELECT name,email,id FROM adressbuch");

// Mittels einer Schleife und der Funktion MySQL_fetch_array()
// ein Array bilden und dies
// mit den Werten der Tabelle füllen

while($row = MySQL_fetch_array($result)) {

    $id    = $row["id"];
    $name  = $row["name"];
    $email = $row["email"];

?><?php

// Wenn Email nicht leer ist, Name mit Email-Link verbinden
if($email != '') {

    echo "<a href=mailto:$email>$name</a><br>";

}
}
```

```
// Ansonsten nur Name anzeigen
else {

    echo "$name<br>";

}

?><?php

// While-Schleife beenden
}

?>
```

Ausgabe:

```
Mustermann
Klausen
Klaus
Mann
Lange
```

Dieses Skript muss im gleichen Verzeichnis wie die `CONNECT.INC.PHP` abgespeichert sein, da der Include sonst nicht funktionieren würde. Man kann natürlich auch mit relativen Pfaden die Adresse zur `CONNECT.INC.PHP` innerhalb der `INCLUDE()`-Funktion verändern. Dies würde beispielsweise so aussehen:

```
include("../Ordner/connect.inc.php");
```

In dem vorangehenden Skript finden zwei wichtige Funktionen Verwendung. Das ist zum einen die `MYSQL_QUERY()`-Funktion und zum anderen die `MYSQL_FETCH_ARRAY()`-Funktion. Erstere sendet eine Datenbank-„Query“ (Anfrage) an die Datenbank und liefert ein Resultat zurück. Mittels der Funktion `MYSQL_FETCH_ARRAY()` wird eine Variable erzeugt, in der die Ergebnisse, Zeile für Zeile gespeichert werden. Diese Variable bekommt als Index die Spaltennamen der Tabelle. Alternativ kann man natürlich auch wieder die numerische Indizierung verwenden (ist aber ungünstiger, da sich Spaltennummern auch einmal verändern können, wenn Sie nachträglich beispielsweise Spalten hinzufügen).

Mein Tipp hierzu: Modifizieren Sie das Skript nach der Übung nach Ihren Wünschen, erzeugen Sie neue Tabellen und versuchen Sie, diese auszulesen. Nach ein wenig Übung können Sie bald neue und eigene Skripte zum Auslesen von MySQL-Datenbanken anfertigen.

► **Neue Datensätze erzeugen**

Nachdem wir nun Datensätze sozusagen „per Hand“ erzeugt, Daten ausgelesen und vor allem Tabellen erzeugt haben, kommen wir nun zum letzten Schritt, dem Eintragen von Daten in die Datenbank mittels PHP.

Wir behalten dabei die Struktur der Testdatenbank, mit

- Name
- Vorname
- Straße
- PLZ
- Ort
- Telefon
- Fax
- E-Mail

Das heißt, bevor mit der Programmierung an sich begonnen wird, muss die Möglichkeit geschaffen werden, die Daten irgendwo einzutragen. Wie einige von Ihnen bereits wissen, geschieht das per HTML-Formular (per Flash ist das auch möglich, aber dazu in späteren Kapiteln noch mehr).



■ ■ ■ *Abbildung 3.11: formular.php auf der CD zum Buch*

Erzeugen Sie also ein HTML-Formular mit folgenden Textfeldernamen:
NAME, VORNAME, STRASSE, PLZ, ORT, TELEFON, FAX, EMAIL

Der Code hierzu lautet:

```

<!-- Formularbeginn -->
<form method="post" action="<?php echo"$PHP_SELF"; ?>"
      name=formular>
<!-- Hier beginnt die Tabelle -->
<table width="50%" border="0" cellspacing="0" cellpadding="0"
align="center">
  <tr>
    <td colspan="2">
      <!-- Überschrift -->
      <div align="center">
        <h2>Formular zur Erzeugung von neuen
          Datensätzen</h2>
      </div>
      <!-- Überschrift Ende -->
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile Name -->
      <div align="center">Name</div>
    </td>
    <td width="50%">
      <input type="text" name="name">
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile Vorname -->
      <div align="center">Vorname</div>
    </td>
    <td width="50%">
      <input type="text" name="vorname">
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile Straße -->
      <div align="center">Straße</div>
    </td>
    <td width="50%">
      <input type="text" name="strasse">
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile PLZ -->
      <div align="center">PLZ</div>
    </td>

```

■ ■ ■ *Listing 3.4:*
formular.php –
Formular zur
Erzeugung von neuen
Datensätzen

```

        <td width="50%">
            <input type="text" name="plz">
        </td>
    </tr>
    <tr>
        <td width="50%">
            <!-- Zeile Ort -->
            <div align="center">Ort</div>
        </td>
        <td width="50%">
            <input type="text" name="ort">
        </td>
    </tr>
    <tr>
        <td width="50%">
            <!-- Zeile Telefon -->
            <div align="center">Telefon</div>
        </td>
        <td width="50%">
            <input type="text" name="telefon">
        </td>
    </tr>
    <tr>
        <td width="50%">
            <!-- Zeile Fax -->
            <div align="center">Fax</div>
        </td>
        <td width="50%">
            <input type="text" name="fax">
        </td>
    </tr>
    <tr>
        <td width="50%">
            <!-- Zeile Email -->
            <div align="center">Email</div>
        </td>
        <td width="50%">
            <input type="text" name="email">
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <div align="center">
                <!-- Absendebuttton -->
                <input type="submit" name="submit" value="Eintragen">
            </div>
        </td>
    </tr>
</table><!-- Hier endet die Tabelle -->
</form><!-- Formularende -->

```

Wer sich die HTML-Arbeit ersparen will, schaut auf der CD zum Buch nach und öffnet dort die Datei FORMULAR.PHP.



Das Wichtigste bei diesem Listing ist die Angabe der Form-Action:

```
<?php echo"$PHP_SELF"; ?>
```

Dieses kleine Skript bewirkt das erneute Laden der Seite nach dem Klicken des EINTRAGEN-Buttons und ermöglicht somit die Verarbeitung des Formulars innerhalb einer Datei.

Der nächste Schritt erfordert etwas Überlegung. Nach der Eingabe der Daten wird der EINTRAGEN-Button gedrückt und die Seite mit den gesetzten Variablen erneut geladen. Die Daten sollen beim zweiten Laden der Seite in die Datenbank eingetragen und es soll eine entsprechende Meldung ausgegeben werden. Das schließt ein, dass das Formular nicht automatisch wieder angezeigt wird. Eindeutig ein Fall für eine If-Struktur. Der Schlüssel hierzu ist die Variable submit, die erst nach dem Absenden des Formulars einen Wert erhält:

```
<?php

if($submit) {

    // Datenbankeintrag und Ausgabe der Meldung

}

if($submit == '') {

    // Formular anzeigen

}

?>
```

Hierbei ist wieder zu beachten, dass, sofern Sie den ganzen HTML-Code innerhalb der ELSE-Struktur anzeigen wollen, die Anführungszeichen mit einem Slash („/“) zu versehen sind. Da das aber etwas umständlich wäre, beenden wir das PHP-Skript vor jeder HTML-Ausgabe und geben die Meldungen an den entsprechenden Stellen aus.

```
<?php if($submit) {

    // Datenbankeintrag

?>

<!-- HTML-Code für Meldungen -->

<?php } ?>
```

```

<?php if($submit == '') { ?>

    <!-- HTML-Code für Formular -->

<?php } ?>

```

Eingebaut in die Datei sieht das nun etwa so aus:

Listing 3.5: ■■■
formular2.php

```

<?php if($submit) {
// Datenbankeintrag

?>
<?php } ?>
<?php if($submit == '') { ?>
<!-- Formularbeginn -->
<form method="post" action="<?php echo"$PHP_SELF"; ?>"
      name=formular>
<!-- Hier beginnt die Tabelle -->
<table width="50%" border="0" cellspacing="0" cellpadding="0"
align="center">
  <tr>
    <td colspan="2">
      <!-- Überschrift -->
      <div align="center">
        <h2>Formular zur Erzeugung von neuen
          Datens&auml;tzen</h2>
      </div>
      <!-- Überschrift Ende -->
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile Name -->
      <div align="center">Name</div>
    </td>
    <td width="50%">
      <input type="text" name="name">
    </td>
  </tr>
  <tr>
    <td width="50%">
      <!-- Zeile Vorname -->
      <div align="center">Vorname</div>
    </td>
    <td width="50%">
      <input type="text" name="vorname">
    </td>
  </tr>
</tr>

```

```
<td width="50%">
  <!-- Zeile Straße -->
  <div align="center">Straße</div>
</td>
<td width="50%">
  <input type="text" name="strasse">
</td>
</tr>
<tr>
  <td width="50%">
    <!-- Zeile PLZ -->
    <div align="center">PLZ</div>
  </td>
  <td width="50%">
    <input type="text" name="plz">
  </td>
</tr>
<tr>
  <td width="50%">
    <!-- Zeile Ort -->
    <div align="center">Ort</div>
  </td>
  <td width="50%">
    <input type="text" name="ort">
  </td>
</tr>
<tr>
  <td width="50%">
    <!-- Zeile Telefon -->
    <div align="center">Telefon</div>
  </td>
  <td width="50%">
    <input type="text" name="telefon">
  </td>
</tr>
<tr>
  <td width="50%">
    <!-- Zeile Fax -->
    <div align="center">Fax</div>
  </td>
  <td width="50%">
    <input type="text" name="fax">
  </td>
</tr>
<tr>
  <td width="50%">
    <!-- Zeile Email -->
    <div align="center">Email</div>
  </td>
  <td width="50%">
```

```

        <input type="text" name="email">
    </td>
</tr>
<tr>
    <td colspan="2">
        <div align="center">
            <!-- Absendebutton -->
            <input type="submit" name="submit" value="Eintragen">
        </div>
    </td>
</tr>
</table><!-- Hier endet die Tabelle -->
</form><!-- Formularende --><?php } ?>

```



Den kompletten Code hierfür finden Sie wiederum auf der CD zum Buch in der Datei FORMULAR2.PHP. Wenn Sie nun auf den EINTRAGEN-Button klicken, sollte sich die gleiche Datei ohne weiteren Inhalt öffnen.

Bleibt nun nur noch der eigentliche Eintragungscode, den es jetzt zu erstellen gilt. Wichtig hierbei ist der INSERT-Befehl von SQL. Er ermöglicht das Einfügen von neuen Datensätzen in die Tabelle.

Ergänzen Sie den Code an den entsprechenden Stellen wie folgt:

Listing 3.6: ■■■
Ergänzung des
Codes zur Datensatz-
erzeugung –
formular3.php

```

<?php
if($submit) {
    // Datenbankeintrag
    // connect.inc.php in die Datei einschließen
    include ("connect.inc.php");
    // Query an die Datenbank senden
    MySQL_db_query("$dbName","INSERT INTO adressbuch
    (name,vorname,strasse,ort,plz,telefon,fax,email)
    VALUES ('$name','$vorname','$strasse','$ort','$plz',
    '$telefon','$fax','$email')");
    // Anzahl der erzeugten Datensätze wird geprüft
    $anzahl = MySQL_affected_rows();
    // Bei mehr als 0 Datensätzen erzeugt > status = positive
    if ($anzahl > 0) {
        $status="Datensatz hinzugefügt !";
    }
}

```

```

// wenn kein Datensatz erzeugt wurde > status = Fehler

else {
    $status="Eintragung fehlgeschlagen !";
}

?>
<?php

// Statusmeldung wird ausgegeben

echo"$status";
?>
<?php

// IF-Struktur schließen
}
?>

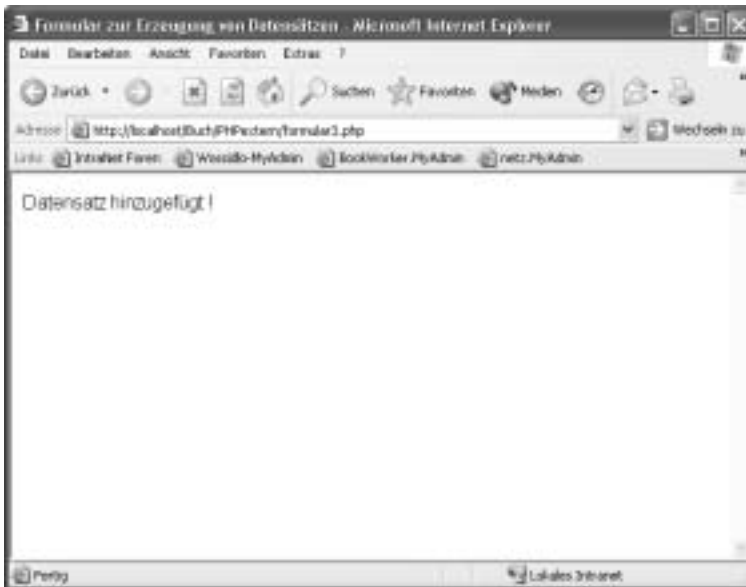
```

Auch hierzu ist die fertige Datei für Sie vorbereitet. Sie finden sie auf der CD zum Buch unter dem Namen `FORMULAR3.PHP`.



Die hierfür wichtige Funktion ist `MySQL_db_query()`. Sie verlangt unbedingt den Parameter „dbName“, den wir bereits in der `CONNECT.INC.PHP` festgelegt haben. Sie können natürlich auch den direkten Datenbanknamen angeben (in diesem Fall „phpundFlash“).

Testen Sie das Formular und erzeugen Sie nun einen neuen Datensatz.



■ ■ ■ **Abbildung 3.12: Erfolgreiche Statusmeldung**

id	name	vorname	strasse	ort	plz	telefon	fax	email
1	Mustermann	Max	Musterstrasse 3	Musterort	12345	0123/456789	0123/456789	max@mustermann.de
2	Klausen	Klaus	Zur Klausur 4	Klausdorf	54321	3210/987654	43210/887664	
3	Klaus	Hans	Berliner Ring 10	Saal	65425			
4	Mann	Daniel	Franzburger 1a	Rostock	89745	8525/456852	8525/456851	dann@manns.de
5	Lange	Jürgen	Denkstuehle 5	Nimmerland	98765	3578/6542654	3578/6542655	buch@bookworker.de
6	Klein	Tim	IndenWolken 7	Laabnitz	18999	32145/65432	32145/65433	tim@timskleinesete.de

■ **Abbildung 3.13:** Neu erzeugter Datensatz markiert



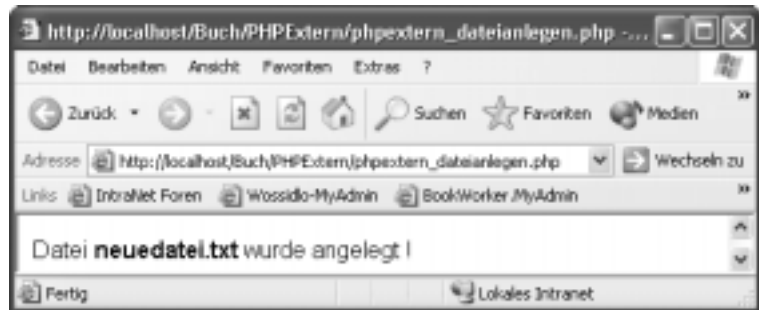
Dies alles ist sehr allgemein gehalten, aber alle Möglichkeiten im Einzelnen zu erläutern, würde hier den Rahmen sprengen. Testen Sie sich selbst anhand von praktischen Übungen und studieren Sie andere Skripte, die Sie beispielsweise unter www.hotscripts.com/PHP/ finden.

3.3 Das Dateisystem

Mit PHP können Sie nicht nur mit Datenbanken, sondern auch mit normalen (textbasierenden) Dateien kommunizieren. Dies kann sehr hilfreich sein, wenn Sie beispielsweise Ordnerstrukturen ausgeben, Daten Ihrer Besucher in Textform abspeichern wollen oder wenn Sie eine Ausweichmöglichkeit für Datenbanken suchen.

Um Informationen in Dateien speichern zu können, müssen Sie Dateien sowohl erstellen, auslesen als auch löschen können. Das alles ermöglicht PHP Ihnen auf ganz einfache Weise.

Dateien anlegen und löschen



■ **Abbildung 3.14:** Neue Dateien anlegen

Lokale Dateien lassen sich recht einfach erzeugen. Mit der Funktion `touch()` bietet Ihnen PHP eine recht komfortable Lösung, um Textdateien aller Art anzulegen:

```
<?php
// Dateinamen bestimmen
$neuedatei = "neuedatei.txt";

// Datei anlegen
if(touch($neuedatei)) {

    // Meldung ausgeben !
    echo "Datei <b>$neuedatei</b> wurde angelegt !";
}
?>
```

■ ■ ■ Listing 3.7:
Neue Dateien anlegen

Ausgabe:

Datei **neuedatei.txt** wurde angelegt !

Die Datei `NEUEDATEI.TXT` wurde erzeugt, enthält aber noch keine Informationen. Bevor wir sie jedoch mit Informationen füllen, möchte ich noch kurz auf das Löschen von Dateien eingehen.

Zum Löschen von Dateien bietet PHP ebenfalls eine bequeme Funktion: `unlink()`.

Voriges Beispiel diesmal mit Löschfunktion:

```
<?php
// Dateinamen bestimmen
$neuedatei = "neuedatei.txt";

// Datei löschen
if(unlink($neuedatei)) {
    // Meldung ausgeben !
    echo "Datei <b>$neuedatei</b> wurde gelöscht !";
}

?>
```

■ ■ ■ Listing 3.8:
Dateien löschen

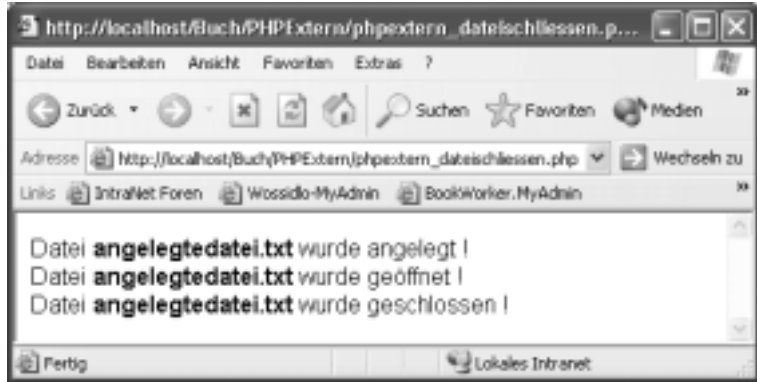
Ausgabe:

Datei **neuedatei.txt** wurde gelöscht !

Letztendlich ist es aber wichtig, die Dateien mit Informationen zu füllen und diese später auszulesen, wozu ich im nächsten Absatz kommen werde.

Dateien öffnen und schließen

Wenn die Datei angelegt wurde, muss sie anschließend auch geöffnet werden, damit man sie lesen kann. Dazu, wie könnte es anders sein, gibt es in PHP ebenfalls eine Funktion: `fopen()`; und ihr Pendant `fclose()`;, welches die Datei wieder schließt. Dazu ein kurzes Beispiel:



■ ■ ■ *Abbildung 3.15: Dateien anlegen, öffnen und schließen*

Listing 3.9: ■ ■ ■
Datei anlegen, auslesen und schließen

```
<?php
// Dateinamen bestimmen
$neuedatei = "angelegtedatei.txt";

// Datei anlegen
if(touch($neuedatei)) {
    // Meldung ausgeben !
    echo "Datei <b>$neuedatei</b> wurde angelegt !<br>";
}

// Datei öffnen
if($dateioffen = fopen($neuedatei,'r')) {
    // Meldung ausgeben
    echo "Datei <b>$neuedatei</b> wurde geöffnet !<br>";
}
if(fclose($dateioffen)) {
    echo "Datei <b>$neuedatei</b> wurde geschlossen !";
}
?>
```

Ausgabe:

```
Datei angelegtedatei.txt wurde angelegt !
Datei angelegtedatei.txt wurde geöffnet !
Datei angelegtedatei.txt wurde geschlossen !
```

Der zweite Parameter der Funktion `fopen()`; bestimmt die Art des Öffnens der Datei. Man unterscheidet dabei folgende Methoden:

- r öffnet die Datei zum Lesen (Dateizeiger auf Anfang)
- r+ öffnet die Datei zum Lesen und Schreiben (Dateizeiger auf Anfang)
- w öffnet die Datei zum Schreiben. Die Datei wird neu angelegt, wenn sie noch nicht existiert. Wenn die Datei bereits vorhanden ist und Daten enthält, werden diese gelöscht und überschrieben (Dateizeiger auf Anfang).
- w+ öffnet die Datei zum Schreiben und Lesen. Die Datei wird neu angelegt, wenn sie noch nicht existiert. Ist Datei bereits vorhanden, werden die Daten gelöscht und überschrieben (Dateizeiger auf Anfang).
- a öffnet die Datei zum Schreiben. Die Datei wird neu angelegt, wenn sie noch nicht existiert. Daten werden aber nicht gelöscht, wenn bereits vorhanden (Dateizeiger am Ende).
- a+ öffnet die Datei zum Schreiben und Lesen. Die Datei wird neu angelegt, wenn sie noch nicht existiert. Ist die Datei bereits vorhanden, werden die Daten nicht gelöscht und bleiben somit erhalten (Dateizeiger am Ende).

Nachdem Sie nun auch Dateien öffnen und schließen können, möchte ich Sie nicht länger auf die Folter spannen und Ihnen erläutern, wie man denn nun Informationen in die Datei schreibt ☺.

Dateien lesen und schreiben



■ ■ ■ *Abbildung 3.16:*
Dateien zeilenweise
auslesen

Da das Lesen und Schreiben von Dateien zwei der elementarsten Vorgänge sind, bietet PHP hier nun ein paar mehr Funktionen. Neben dem häufig verwendeten `file()`; (zum Einlesen der Datei in ein Array) sind weitere wichtige Funktionen, wie z.B. `fgets()`; (zum Auslesen der Zeilen) und `fwrite()`; (zum Schreiben von Binärdaten) unerlässlich.

Wir wollen nun eine Datei auslesen, die zehn Zeilen mit je einer Ziffer (0–9) enthält.

Listing 3.10: ■■■
*Datei zeilenweise
auslesen*

```
<?php
// Datei, die auszulesen ist, angeben
$meinedatei = "zahlen.txt";

// Datei öffnen
if($datei = fopen($meinedatei, "r")) {

    // Solange der Zeiger nicht am Ende der Datei ist
    //Zeile auslesen und ausgeben

    while(! feof($datei)) {
        $zeile = fgets($datei, 1024);
        print("$zeile<BR>\n");
    }
}
else {
    // Ansonsten Fehlermeldung ausgeben
    print("Datei konnte nicht geöffnet werden");
}

?>
```

Ausgabe:

```
0
1
2
3
4
5
6
7
8
9
```

Die Funktion `feof()`; überprüft, ob der Zeiger bereits am Ende (end of file) angekommen ist und die Funktion `fgets()`; liest pro Zeile so viele Zeichen wie im zweiten Parameter angegeben. Sollte vorher ein Zeilenumbbruch vorkommen, wird das Auslesen der Zeile dort beendet.

Wichtig ist, dass man beim Programmieren Fehlerüberprüfungsroutinen einbaut. Auch wenn es sehr unwahrscheinlich ist, dass beispielsweise beim Test eine Datei nicht geöffnet werden kann, sollte man (auch im Allgemeinen) die andere Möglichkeit in der Programmierung mit einbeziehen.



Des Weiteren ist es nach wie vor sinnvoll, Kommentare und Einrückungen zu setzen, damit der Code auch für andere Programmierer bzw. zu anderen Zeiten immer verständlich bleibt.

Versuchen Sie nun einmal, die Vorgehensweise für das Schreiben in eine Datei zu konstruieren. Dazu ein etwas einfacherer Vorschlag: Schreiben Sie einfach die Zahlenreihe 1 bis 10 in eine Zeile in eine Textdatei. Nutzen Sie dazu die Funktion `fwrite()`; und verwenden Sie zum Zählen der bereits geschriebenen Ziffern eine `While()`-Schleife!

Die Lösung habe ich nachfolgend einmal vorbereitet; versuchen Sie aber, sie nicht gleich zu verwenden:

```
<?php
// Dateiname festlegen
$datei      = "eigenerversuch.txt";

// Datei öffnen
$dateioffen = fopen($datei, 'w+');

// While-Schleife
while ($zaehler < 10) {

    // Zähler hochzählen
    $zaehler = $zaehler +1;

    // Werte in die Zeile schreiben
    fwrite($dateioffen, $zaehler);

    // While-Schleife beenden
}

?>
```

■ ■ Listing 3.11:
*Werte in eine Datei
schreiben*

Ich hoffe, dass Sie das Problem zu Ihrer Zufriedenstellung lösen konnten und dass mein Skript nur eine kleine Hilfe dazu war.

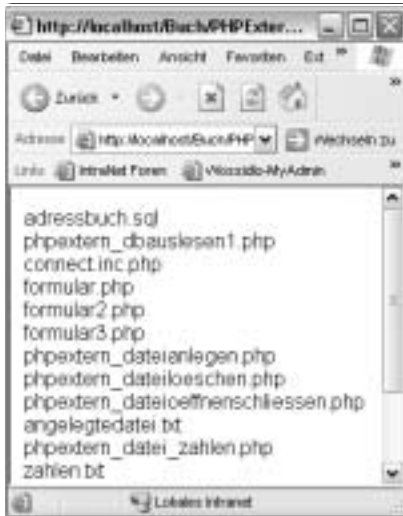
Dies ist leider auch nur ein kleiner Ausschnitt aus dem, was mit PHP möglich ist. Um vielleicht eine Anregung zu geben, was im Zusammenhang mit Dateien noch alles machbar ist, erhalten Sie eine kurze Funktionsübersicht der Dateifunktionen in PHP.

Übersicht der Dateifunktionen

Funktionsname	Beschreibung
<code>fwrite(Zeiger, Zeichenkette);</code>	Schreibt den Inhalt der Zeichenkette an die Position, an der sich der Zeiger der Datei befindet.
<code>fputs(Zeiger, Zeichenkette);</code>	Hängt den Inhalt der Zeichenkette an die Datei des Zeigers an.
<code>fgets(Zeiger, Zeichen);</code>	Liest den Inhalt der aktuellen Zeile aus, an der sich der Zeiger befindet, und zwar nur so weit, wie Zeichen angegeben sind oder bis ein Zeilenumbruch kommt. Springt dann in die nächste Zeile.
<code>fgetss(Zeiger, Zeichen);</code>	Wie <code>fgets()</code> , entfernt außerdem PHP und HTML-Tags.
<code>fread(Zeiger, Bytes);</code>	Liest vom aktuellen Zeichen, auf das der Zeiger verweist, die Anzahl an Bytes aus.
<code>fgetscsv();</code>	Liest die aktuelle Zeile, auf die der Zeiger verweist, in ein Array aus. Das Besondere an dieser Funktion ist, dass sie zusätzlich eine CSV-Datei (durch Trennzeichen getrennte Spalten) in das Array ausliest. Die Zeilenlänge muss hier größer sein als die maximale Länge aller in der Datei vorkommenden Zeilen. Die Angabe des Trennzeichens ist optional, der Standard ist ein Komma.
<code>ftell(Zeiger);</code>	Liefert die aktuelle Position des Zeigers zurück.
<code>fseek(Zeiger, Offset, [Startpunkt]);</code>	Bewegt den Zeiger an die Stelle <code>Offset</code> (in Byte). Optional kann ein Startpunkt festgelegt werden: <code>>Seek_SET</code> – setzt den Startpunkt auf Anfang <code>>Seek_CUR</code> – setzt den Startpunkt auf die aktuelle Position <code>>Seek_END</code> – setzt den Startpunkt auf das Dateiende
<code>feof(Zeiger);</code>	Liefert „true“ zurück, sobald der Zeiger am Ende der Datei angelangt ist.

■ **Tabelle 3.2: Übersicht der Dateifunktionen (Auswahl)**

Dateimanipulation und Verzeichnisse



■ ■ ■ **Abbildung 3.17:**
Elementare Ausgabe von
Verzeichnissen mit PHP

Funktion	Beschreibung
Dateimanipulation	
Filegroup(Dateiname);	Liefert die Gruppenzugehörigkeit einer Datei.
Fileperms(Dateiname);	Liefert Zugriffsrechte einer Datei.
Fileowner(Dateiname);	Liefert den Eigentümer einer Datei.
Chgrp(Dateiname, Gruppe);	Ändert die Zugehörigkeit einer Datei zu einer Gruppe.
Chmod(Dateiname, Mode);	Ändert die Zugriffsrechte einer Datei durch Mode.
Chown(Dateiname, Eigentümer);	Ändert den Eigentümer einer Datei.
Copy(Quelle, Ziel);	Kopiert eine Datei (Quelle) in die Datei (Ziel).
Rename(Quelle, Ziel);	Benennt eine Datei (Quelle) in die Datei Ziel um.
Verzeichnisse	
Is_dir(Verzeichnisname);	Gibt „true“ zurück, wenn es sich um ein Verzeichnis handelt.
Mkdir(Verzeichnisname);	Erstellt ein neues Verzeichnis.
Rmdir(Verzeichnisname);	Löscht ein Verzeichnis, welches leer sein muss.
Chdir(Verzeichnisname);	Wechselt in das Verzeichnis, welches unter Verzeichnisname angegeben ist.
Opendir(Verzeichnisname);	Erzeugt einen Handle (Zeiger) auf ein Verzeichnis.
Readdir(Verzeichnishandle);	Liefert den nächsten Dateinamen im Verzeichnis.
Rewinddir(Verzeichnishandle);	Setzt den Verzeichnishandle auf die erste Datei zurück.
Closedir(Verzeichnishandle);	Schließt den Handle auf ein Verzeichnis.

■ ■ ■ **Tabelle 3.3:** Dateimanipulations-/Verzeichnisfunktionen

Ein weiterer Vorteil von PHP ist, dass man mit Hilfe bestimmter Funktionen Dateien und Verzeichnisse auslesen, ändern oder löschen kann. Im Grunde genommen kann man also konkrete Dateimanagementsysteme mit PHP erstellen (z.B. Windows-Explorer). Da das Ganze dann webbasiert ist, birgt diese Möglichkeit natürlich Risiken, die es zu beachten gilt. Man sollte also immer darauf achten, dass die CHMOD-Rechte vernünftig vergeben sind und gegebenenfalls einzelne Verzeichnisse gesichert werden.

Auch hier gibt es wieder einige Funktionen, bei denen es sich lohnt, sie kurz vorzustellen. Testen Sie doch einfach einmal ein paar dieser Funktionen anhand von kleinen Skripten. Später werden noch einige Beispiele folgen.

Hierzu nun ein kleines Beispielskript:

Listing 3.12: ■■■
fileperms()

```
<?php
fileperms());

$datei = "eigenerversuch.txt";

// Zugriffsrechte abfragen
if($ausgabe1 = fileperms($datei)) {

    echo "Zugriffsrechte für $datei: $ausgabe1<br>";

}

?>
```

Ausgabe:

33206



Eine wirklich empfehlenswerte, exakte Übersicht über alle Verzeichnis- und Dateifunktionen gibt es auf Deutsch unter <http://www.php.net/manual/de/ref.filesystem.php>.

Ein weiteres Beispiel liefert alle Dateien im Root-Verzeichnis zurück:

Listing 3.13: ■■■
*Daten aus dem Root-
Ordner ausgeben*

```
<?php

// Verzeichnis relativ zur Datei festlegen
$verzeichnishandle = opendir('/_Projekte/Buch/PHPEextern/');

// Während das Verzeichnis ausgelesen wird, Dateien ausgeben
while($datei = readdir($verzeichnishandle)) {
    if($datei != "." && $datei != "..") {
        print("$datei\n");
    }
}
}
```

```
// Verzeichnis schließen
closedir($verzeichnishandle);
```

```
?>
```

Ausgabe (in diesem Fall immer im Verhältnis zum ausgewählten Verzeichnis):

```
adressbuch.sql
phpextern_dbauslesen1.php
connect.inc.php
formular.php
formular2.php
formular3.php
phpextern_dateianlegen.php
phpextern_datei_loeschen.php
phpextern_dateioeffnenschliessen.php
...
```

Solche Skripte kann man dann später noch weiter ausbauen und beispielsweise je nach Dateityp (zum Beispiel .php, .txt, .doc, .jpg) entsprechende Bilder (Icons) vor dem Dateinamen ausgeben, wie Sie das aus dem Windows-Explorer kennen.

3.4 PHP und FTP

Nachdem wir eine ganze Menge über die Kommunikation mit lokalen Dateien erfahren haben, wenden wir uns nun externen Systemen zu, wie z.B. FTP. Das File Transfer Protocol ist eines der wichtigsten Protokolle des Internets. Die Hauptanwendung des Datei-Transfer-Protokolls ist das „Hochladen“ von Dateien auf einen FTP-Server.

Zitat de.comp.lang.php FAQ:

»PHP hat eingebaute FTP-Funktionen, mittels derer man Dateien von oder zu einem FTP-Server übertragen kann. Es ist damit aber nicht möglich, Daten von einem Browser zu empfangen! Dies ist also keine Alternative zum HTTP-Upload per HTML-Formular (siehe hierzu "Wie funktioniert ein Datei-Upload über HTML-Formulare?").

PHP kann nur einen FTP-Client darstellen, der Verbindung zu einem FTP-Server aufnimmt, d.h., der Verbindungsaufbau und die -steuerung müssen immer vom PHP-Skript aus erfolgen und die Daten für den Upload müssen sich bereits auf dem Webserver befinden. Vollständige Beispiele für eine FTP-Verbindung mit PHP finden sich im Manual.«

Quelle: http://www.koehntopp.de/php/open_exec.html#datei-upload-ftp



Auch PHP, bietet die Möglichkeit, mit FTP-Servern zu kommunizieren. Optimal ist das, wie Sie im obigen Tipp bereits gelesen haben, allerdings nicht. Deshalb dazu auch nur ein kleines Skript, welches die Verbindung zu einem FTP-Server aufbaut, eine Textdatei vom Server lädt und diese dann ausgibt:

Listing 3.14: ■■■
**Dateien von einem
 FTP-Server down-
 loaden und im Browser
 ausgeben**

```
<?php
// herunterzuladende Datei
$ftpdatei = "textdatei.txt";

//Eine Datei vom FTP-Server laden
$ftpserver = ftp_connect("ftp.deinserver.de");
if (!$ftpserver) {
    die("Verbindung fehlgeschlagen");
}
$benutzername = "ftpbenutzername";
$passwort = "ftppasswort";
$einloggen = ftp_login($ftpserver, $benutzername, $passwort);
if (!$einloggen) {
    die("Login fehlgeschlagen");
}
ftp_pwd($ftpserver, "Verzeichnis");
ftp_get($ftpserver, $ftpdatei, $ftpdatei, ftp_ascii);
ftp_quit($ftpserver);

//Die Datei im Browser ausgeben
if($datei = fopen($ftpdatei, "r")) {
    $inhalt = fread ($datei, filesize ($ftpdatei));
    print($inhalt);
    fclose ($datei);
} else {
    die("Datei konnte zum Lesen nicht geöffnet werden");
}
?>
```

Dieses Skript ist etwas schwieriger und funktioniert höchstwahrscheinlich auch nicht auf Anhieb bei jedem. Es geht aber auch nur darum, die Funktion des FTP im Zusammenhang mit PHP zu verstehen. Wichtig ist, dass immer erst eine Verbindung zum FTP-Server aufgebaut werden muss und dann erst Transaktionen stattfinden können.

Im Zusammenspiel mit Flash finden die FTP-Funktionen bei mir weniger Verwendung, da meiner Meinung nach bei solchen Systemen eher Funktionalität als Ästhetik eine Rolle spielt. Solch ein System mit Flash umzusetzen, ist meines Erachtens nach Zeitverschwendung.

Kommunikation zwischen Flash und PHP



Azad Adsay

*„Dass wir miteinander reden können,
macht uns zu Menschen“.*

Karl Jaspers

Gegenstand dieses Buches ist die Kommunikation von Flash mit PHP. Damit es eine solche Kommunikation geben kann, müssen die Daten für Flash „übersetzt“ werden, sie werden kodiert. Dies ist prinzipiell nicht notwendig, aber für das Benutzen von Sonderzeichen wie „&“ oder „+“ muss so verfahren werden.

4.1 Textdateien einladen

Wie in der Einführung schon beschrieben, lassen sich Textdaten ganz einfach laden. Die Textdatei DATA.TXT hat folgenden Inhalt:

```
input1=Hallo Welt!
```

Mit Flash werden die Daten dann wie folgt geladen:

```
loadVariablesNum("data.txt",0);
```

In Flash wird „Hallo Welt!“ wiedergegeben. In PHP funktioniert das so jedoch nicht.

4.2 Daten aus PHP-Dokumenten einladen

Bei PHP-Dokumenten werden nur Daten angezeigt, welche der Programmierer auch ausgeben will. Das heißt – wenn der Programmierer die Variable \$a wiedergeben will, aber nicht \$b, schreibt er Folgendes in sein Projekt:

*Listing 4.1: ■■■
Lediglich Variable a
kann ausgelesen werden,
Variable b bleibt für jeden (außer den
Entwickler natürlich)
unsichtbar.*

```
<?
$a = „Text“;
$b = „geheimer Text“
echo $a; ?>
```

Für Besucher ist es unmöglich, Variable \$b herauszufinden. Auch mit Flash geht es nicht anders. Möchten Sie also den Inhalt der Variablen \$a lesen, muss sie erst mit echo ausgegeben werden. Doch damit ist es noch nicht getan. In dem echo-Befehl muss wiedergegeben werden, welche Variable in Flash angezeigt werden soll. Das könnte dann so aussehen:

*Listing 4.2: ■■■
Die Variablenübergabe
wird in PHP vorbereitet*

```
<?
$a = "Hallo Welt!";
echo "input1=$a";
?>
```

Hier müsste das Textfeld „input1“ heißen.

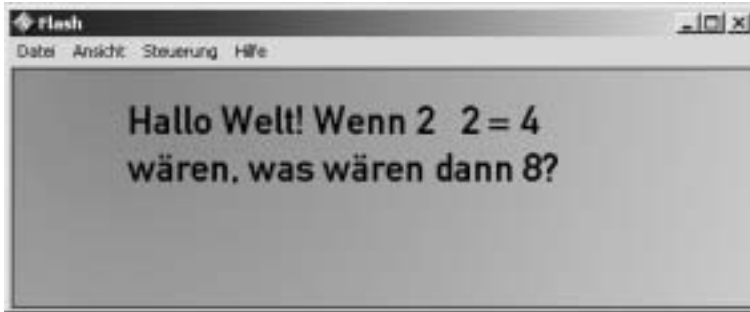
4.3 Sonderzeichen kodieren

Wenn wir aber in der Variablen \$a Sonderzeichen einsetzen, wie in dem nachfolgenden Beispiel, erhalten wir keine korrekte Ausgabe.

Der Input:

```
input1="Hallo Welt! Wenn 2 + 2 = 4% wären, was wären dann 8% ?";
```

Die Ausgabe:



■ ■ ■ *Abbildung 4.1:*
Das Plus- und das Prozent-Zeichen fehlen

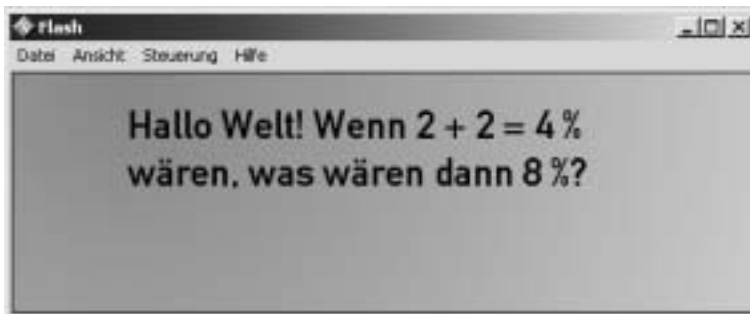
Sie sehen, die Ausgabe ist massiv verfälscht. Es gibt hier zwei denkbare Lösungsstrategien. Die erste Möglichkeit wäre, alle Sonderzeichen per Hand auszutauschen. Das ist nur möglich, wenn die Daten lediglich von einem Entwickler bearbeitet werden. Diese Prozedur ist jedoch dem Entwickler kaum zuzumuten, dem Kunden schon gar nicht. Ersetzt werden könnten die Zeichen wie folgt:

- %2B – Plus Zeichen – +
- %25 – Prozentzeichen – %
- %26 – Und-Zeichen – &

Das könnte dann so aussehen:

```
input1="Hallo Welt! Wenn 2 %2B 2 = 4 %25 wären,  
was wären dann 8 %25?";
```

■ ■ ■ *Listing 4.3:*
Manuell formatierte Sonderzeichen



■ ■ ■ *Abbildung 4.2:*
Nun ist die Ausgabe korrekt.

Wenn jedoch ein Kunde Daten eingeben soll oder die Daten aus einer Datenbank kommen und diese dynamische Inhalte hat, ist ein solche Art der Bearbeitung noch schlechter – speziell bei Eingaben durch Besucher bzw. nicht Programmierbewanderte in einem Gästebuch oder einem Forum ist diese Methodik nahezu unmöglich.

Jetzt setzen wir die PHP-Funktion `rawurlencode()`; ein:

```

<?
$a = "Hallo Welt! Wenn 2 + 2 = 4% wären, was wären dann 8%?";
$output = rawurlencode($a);

echo "input1=$output";
?>

```

PHP gibt also aus:

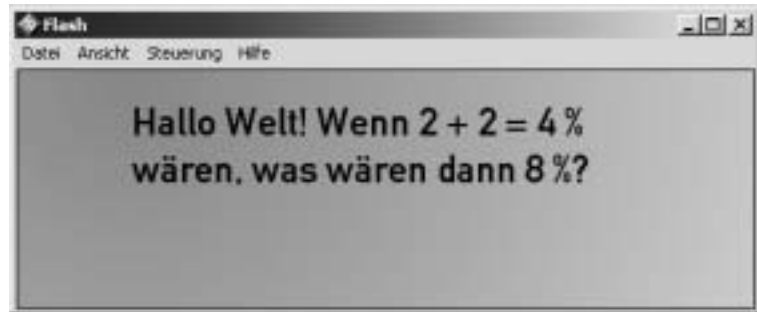
```

input1=Hallo%20Welt%21%20Wenn%202%20%2B%202%203D%204%25%20%20w%
E4ren%2C%20was%20w%E4ren%20dann%208%3F

```

Wie Sie sehen können, wurden die Leerzeichen auch schon mit %20 ersetzt.

Abbildung 4.3: ■■■
Nun ist die Ausgabe
perfekt.



4.4 Überprüfung des Ladevorgangs

Um zu gewährleisten, dass die Daten wirklich angekommen sind, kann mit Flash eine Schleife eingebaut werden, die den Wert einer Variablen überprüft, welche eigens dafür vorhanden ist. Diese Variable kann einfach an das Ende der eigentlichen Variablen gesetzt werden. Das sähe zum Beispiel so aus:

Listing 4.4: ■■■
Übergabe zweier
Variablen

```

input1=Hallo Welt! Wenn 2 + 2 = 4% wären, was wären
dann 8%?&loaded=1

```

In Flash wird dann nach einer gewissen Zeit überprüft, ob loaded auch gleich 1 ist, andernfalls muss ein Fehler vorliegen.

Listing 4.5: ■■■
Überprüfung des
Ladevorgangs

```

loadVariablesNum("data.txt", 0);
waiting_time=50;
if (loaded != 1 && timeout >= waiting_time) {
    trace ("Timeout überschritten!");
} else {
    timeout = Number((timeout)+1);
}

```

Als zu wartende Zeit wird 50 eingegeben. Wenn die Zeit verstrichen ist und loaded nicht den Wert 1 hat, gibt Flash eine Fehlermeldung für den Entwickler aus.



Das Cache-Verhalten steuern

Azad Adsay

*„Freunde, vergeßt nie diese goldene
Regel: Nichts kann schaden in der Kunst,
solange ihr frech seid“.*

James Whistler

Moderne Internet Browser speichern Daten jeder Website auf die lokale Festplatte, um Ladezeit bei einem späteren Besuch zu verkürzen, sofern sich die Daten nicht geändert haben. Dies ist ein prinzipiell sehr cleveres Extra. Für normale Surfer ein Segen, für Entwickler eine Mischung aus Segen und Fluch.

Das Problem ist nur, dass der Browser nicht zwangsläufig die richtige Datei erkennt. Spielt der Entwickler ein neues Bild auf den Server, merkt der Browser das. Aber was ist, wenn dynamische Inhalte integriert werden? Bei HTML-Seiten geht das auch ohne Probleme, nicht jedoch bei Flash-Filmen. Dies werden Sie vielleicht schon im dynamischen Datenaustausch mit Flash bemerkt haben.

Das Problem kann sehr leicht mit Flash, aber auch mit PHP gelöst werden. Wir beginnen mit einer kleinen Flash-internen Lösung, zuvor erfolgt jedoch eine Erklärung des Prinzips.

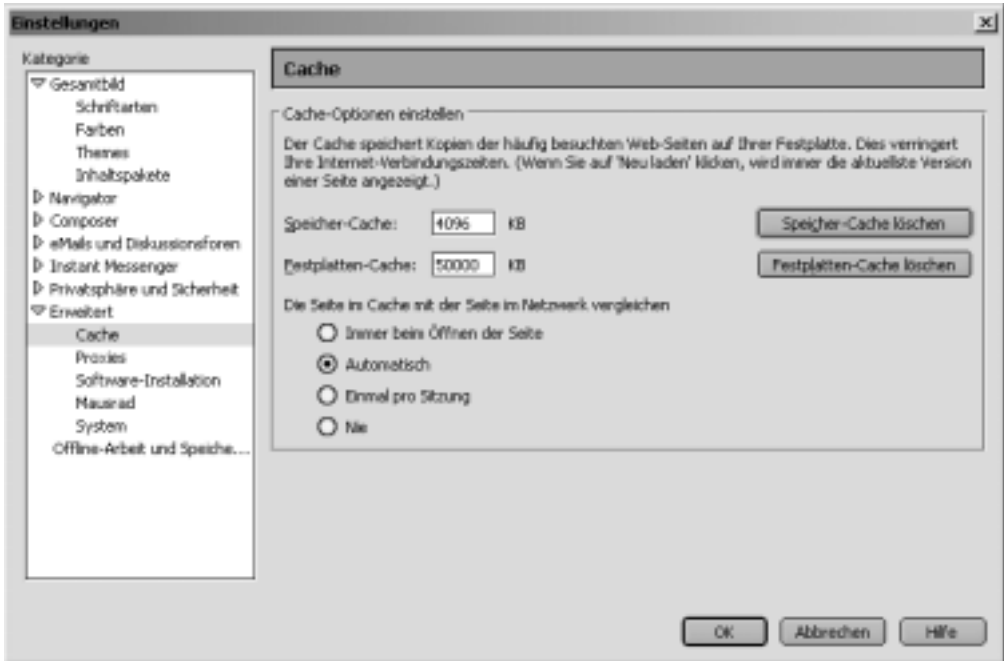
5.1 Leeren des Cache

Wenn Sie einen Flash-Film laden, wird er, sofern es nicht das erste Mal ist, aus dem Cache geholt. Werden Daten dynamisch geladen, ignoriert der Browser das. Das Drücken der **[F5]**-Taste, was eigentlich alle Dateien aktualisieren soll, hilft hier auch nicht. Zwar ist es möglich, den Inhalt des Cache wie in dem nachfolgenden Bild zu leeren, aber man kann von keinem Besucher erwarten, dass er dies tut. Des Weiteren ist es auch für Entwickler kein Spaß, ein und denselben Klickvorgang immer und immer wieder zu wiederholen.



■ ■ ■ *Abbildung 5.1: Cache leeren im Internet Explorer 6*

Im Internet Explorer kann der Cache über **EXTRAS > INTERNETOPTIONEN > TEMPORÄRE DATEIEN** wie in *Abbildung 5.1* geleert werden.



■■■ *Abbildung 5.2: Cache leeren im Netscape Communicator 6*

5.2 Den Cache austricksen

Wird eine Datei jedoch mit einer Variablen geladen, reagieren die Browser anders.

Die Datei KRITIK.SWF wird mit der Variablen `no-cache` geladen. Das würde so aussehen: `http://www.azadadsay.com/kritik.swf?no-cache=1`

Wenn aber der Wert der Variablen `no-cache` variieren würde, würde der Browser einen Wechsel eines Attributs der Datei registrieren und die Datei nicht aus dem Cache laden.

Den Cache mit Flash austricksen

Flash bietet zwei Lösungen an: einen Flash-Film mit `GETURL` zu öffnen oder vorzugsweise den Flash-Film zu laden. Wir behandeln nur die zweite Methode.

```
nc = Number(random(9999));
go = "http://www.work4.de/buch/kritik/kritik.swf?ac=" + nc;
loadMovieNum (go, 0);
```

■■■ *Listing 5.1: Cache mit Flash austricksen*

Mit der Funktion `random` wird eine zufällige Zahl ausgewählt. Das ergibt jedoch nur im Bereich der angegebenen Zahl, hier 9999, 0 mit eingeschlossen, einen realen Zahlen-Output im Bereich von 0–9998.

Den zufällig generierten Wert speichern wir in die Variable `nc`. In die Variable `go` wird die Adresse zu dem Film mit der angehängten Variablen `nc` gespeichert. Ergibt der Zufallsgenerator also zum Beispiel 1234, wäre der Wert in der Variablen `go`:

```
http://www.work4.de/buch/kritik/kritik.swf?ac=1234
```

Erfolgt ein erneuter Besuch der Seite und `nc` beinhaltet die Zahl 6666, dann sieht die Ausgabe von `go` wie folgt aus:

```
http://www.work4.de/buch/kritik/kritik.swf?ac=6666
```

Schon wird die Datei nicht mehr aus dem Cache geladen.

Den Cache in PHP austricksen

Auch mit PHP kann eine Variable zufällig generiert werden. Wir verwenden hier eine Kombination aus Datum und Zeit:

Listing 5.2: ■■■

*Laden der Flash-Datei
cb.swf*

```
<html>
<title>no cache</title>
<body bgcolor="#CCCC66">

<?
    $send_date = (date("d.m.Y"));
    $send_time = (date("H.i.s"));

    $nc = "$send_date.$send_time";
    ?>

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
Flash/swFlash.cab#version=5,0,0,0" width="500" height="350">
    <param name=movie value="cb.swf?cache=<?php echo $nc ?>">
    <param name=quality value=high>
    <embed src="cb.swf?cache=<?php echo $nc ?>" quality=high
    Plug-Inspage="http://www.macromedia.com/shockwave/download/
    index.cgi?P1_Prod_Version=ShockwaveFlash" type="application
    /x-shockwave-Flash" width="500" height="350">
    </embed>
</object>

</body>
</html>
```

Zuerst wird das Datum, danach die Zeit ermittelt. Handelt es sich also um den 13. April 2002 und 02:15:43 Uhr, haben die Variablen folgende Werte:

```
$send_date: 13.04.2002  
$send_time: 02.15.43
```

In der Variablen `$ac` werden die beiden anderen Variablen vermischt. Die Ausgabe lautet: `13.04.2002.02.15.43`, das Auftreten dieses Werts ist einzigartig.

Im HTML-Code wird der Flash-Film mit der Variablen `ac` geladen, ein Auszug daraus:

```
cb.swf?cache=<?php echo $nc ?>
```

■■■ *Listing 5.3:*
cb.swf mit der Variablen cache laden

	1	Eine Site komplett in Flash aufbauen	...123
		Das Intro	...124
		Hauptseite und Navigation	...135
		Text- und Bildelemente ein- und ausblenden, bewegen und skalieren	...143
	2	ECards versenden (Flash und PHP)	...163
		Der Flash-Film	...164
		Die PHP- und Datenbankprogrammierung	...188
	3	Flash-Spiel mit Highscore (Flash und PHP)	...209
		Das Spiel programmieren	...210
		Der Highscore auf Datenbankbasis	...214
		Highscore-Ausgabe	...216
	4	Kleines News-System (Flash und PHP)	...219
		Die Datenbank-Tabelle	...220
		Daten auslesen	...221
		Daten eintragen	...222
		Komplexes System mit erweiterter Tabelle	...224
	5	Newsletter (PHP)	...229
		Mailadressen eintragen	...230
		Newsletter senden	...232
		Datenbankanbindung	...234
	6	Feedback-Protokoll (Flash und PHP)	...235
		Tabelle anlegen	...236
		PHP-Programmierung	...236
		PHP-Dateien	...237
		Flash	...242
	7	Texte ausleuchten (Flash und PHP)	...251
	8	Bilder dynamisch mit PHP in Flash laden	...253
	9	Ausblick Dreamweaver MX	...261
		Software-Review	...262
		Workshop: Dreamweaver richtig konfigurieren	...264



Teil II: Anwendungen

Eine Site komplett in Flash aufbauen



Azad Adsay

„Ich denke, also bin ich“.

René Descartes



■ ■ ■ **Abbildung 1.1:** Das fertige Projekt, die Homepage von Claudia Black

In diesem Kapitel werden Sie eine vollständige Flash-Seite entwickeln und dabei erfahren, wie man das Macromedia-Programm in der Praxis einsetzen kann.

Als Beispiel für die Entwicklung dient uns die frei erfundene Figur der selbständigen Online-Journalistin Claudia Black. Folgendes Szenario gilt für dieses Kapitel:

Die Homepage von Claudia Black soll schlicht, aber modern und dabei klassisch sein, jedoch nicht farblos gestaltet werden. Claudia Blacks

Logo wurde bereits entwickelt und soll aktiv in die Seite eingebunden werden. Die Seite soll eine digitale Visitenkarte darstellen. Ein News-System oder Administrierfähigkeit für die Kunden ist dementsprechend nicht notwendig.

In diesem Kapitel werden wir äußerst wenig ActionScript einsetzen. Ich denke, man muss die Entwicklung einer Seite beobachten, damit man ihren Aufbau richtig verstehen kann. Für mich ist es immer ein zufriedenes Gefühl, wenn ich jedes Detail der Seite kenne, wenn ich sehe, wie sich die Seite entwickelt und aus einzelnen Bestandteilen ein Gesamtwerk entsteht. Sicherlich lässt sich durch ActionScript auch Zeit einsparen, weshalb ich es an einer Stelle einsetzen werden, an der das Entwickeln per Hand immens eintönig sein würde. Grundsätzlich jedoch begnüge ich mich hier mit sehr simplen Skripten.

Bevor die Entwicklung beginnt, noch ein kleiner Hinweis: Es soll nicht Ihre Aufgabe sein, pixelgenau eine Seite nachzubauen. Zeichnen Sie die Objekte frei, verändern Sie kleine Effekte nach Ihrem Belieben – es geht hier um das Erlernen des Grundaufbaus einer gesamten Seite und kleiner grafischer Tricks von Flash.



■■■ *Abbildung 1.2: Das Logo von Claudia Black*



Auf der CD finden Sie unter anderem das Logo bereits im Vektorformat. Öffnen Sie die Datei LOGO.FLA im Ordner des Kapitels 1/Teil II.

Bevor wir die Seite entwickeln, planen wir das Projekt genau durch. Die Seite wird mit einem Intro eingeleitet. Danach erhält der Besucher die Möglichkeit, aus verschiedenen Themen zu wählen.

1.1 Das Intro

In dem Intro wird sich das Logo langsam zusammensetzen. Dazu werden wir die gesamte Größe des Films von 500 Pixel in der Breite und 350 Pixel in der Höhe ausnutzen. Als Hintergrundfarbe nehmen wir den Grauton #CECFCE. Stellen Sie dazu die Bildrate auf 20 Bps (bzw. engl. fps).

Das C und das B aus dem Hintergrund von Claudia Blacks Logo sollen sich wie in den nachfolgenden Bildern eindreihen:

■■■ *Abbildung 1.3:
Das Intro setzt sich
wie in diesem Bild
zusammen*





Öffnen Sie die Datei LOGO.FLA der CD und kopieren Sie das Logo heraus. Fügen Sie es in unser Projekt ein. Sie müssen nun das C und das B in zwei verschiedene Ebenen bringen. Markieren Sie dazu zunächst das zusammengesetzte CB und fügen Sie es in eine neue Ebene wieder ein. Skalieren Sie es wie in der folgenden Abbildung, so dass es einen Großteil der Bildfläche füllt, jedoch noch Ränder zu allen Außenseiten aufweist. Dazu markieren Sie das Objekt und ziehen es mit Hilfe des Skalierwerkzeugs in die gewünschte Größe.

Abbildung 1.4: ■■■
Zentrieren Sie das CB
mit Rändern wie in der
Abbildung.



Verschieben Sie das B jetzt, so dass es von dem C exakt abgetrennt ist, ohne dass Überreste abstehen (beachten Sie dazu Abbildung 1.5).

Abbildung 1.5: ■■■
Das C ist von dem B
abgetrennt.



Schneiden Sie das B mit **Strg** + **X** aus und fügen Sie es danach wieder mit **Strg** + **⇧** + **V** in eine neue Ebene ein. Durch die zweite Tastenkombination wird das B an seiner alten Position eingefügt, von dort, wo es zuvor ausgeschnitten wurde.

Formatieren Sie die beiden Buchstaben jeweils zu einer Filmsequenz. Markieren Sie dazu einen Buchstaben und drücken Sie **F8**. Führen Sie dasselbe mit dem zweiten durch.

Wieso setzen wir hier Filmsequenzen ein? Weder das C noch das B sollen weitergehend animiert werden. Dies geschieht jedoch nicht aus reiner Willkür. Flash hat den Fehler, dass Objekte, die in Bilder formatiert wurden, größer sind als Objekte, die zu Filmsequenzen formatiert wurden. Formatieren Sie also alle Objekte entweder als Filmsequenz oder als Schaltfläche, jedoch nie als simple Grafik.



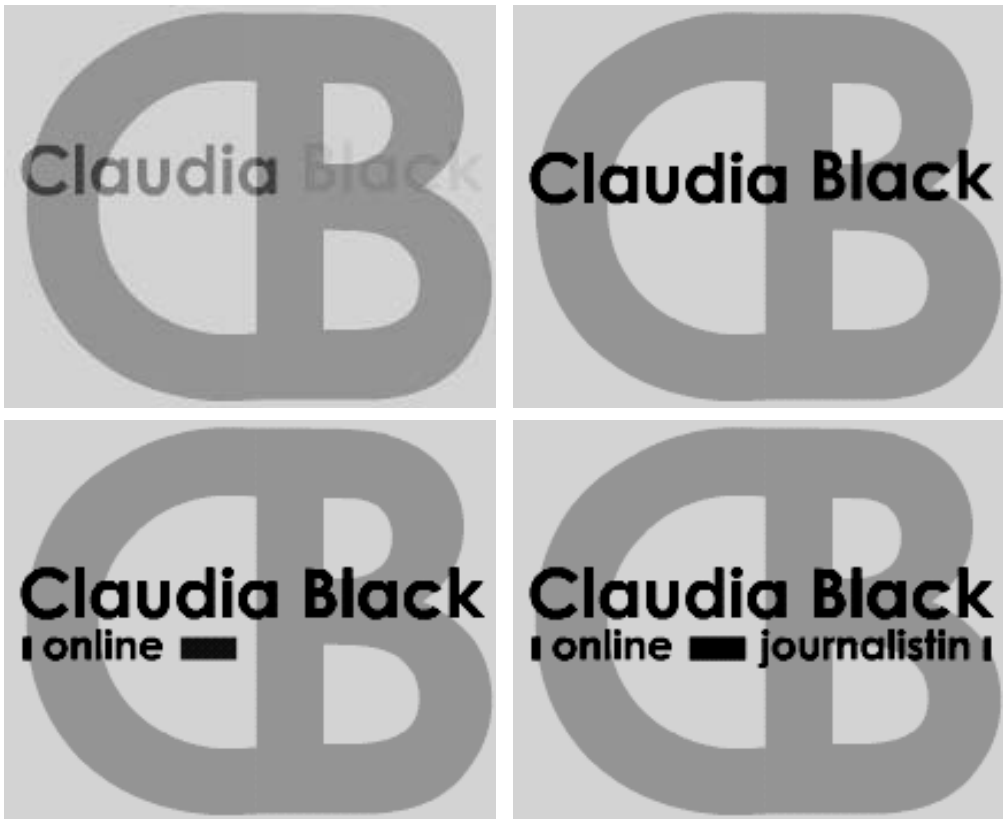
Durch das Drücken der **F5**-Taste kann die Zeitleiste mit „leeren Schlüsselbildern“ aufgefüllt werden. Fügen Sie 19 leere Schlüsselbilder hinzu und danach wieder ein Schlüsselbild in beide Ebenen ein (mit **F6**).

Im 20. Schlüsselbild sollen sich das C und das B fertig eingedreht haben. Zwanzig Schlüsselbilder bilden eine Sekunde, dies haben wir am Anfang der Entwicklung so gewählt (Bildrate: 20 Bps). Markieren Sie nun das C und das B in den ersten Schlüsselbildern. Mittels des Drehwerkzeugs drehen Sie das C und das B um 90°, so dass das C oben steht und nach unten hin geöffnet ist. Verschieben Sie das C jetzt noch ein Stück nach oben, das B nach unten. Fügen Sie danach ein Bewegungstween hinzu.



■ ■ ■ **Abbildung 1.6:**
Das CB zu Beginn der Animation

Nun soll noch der Untertitel „Online Journalistin“ hinzugefügt werden, ebenso wie die ausfüllenden Balken des Logos (Abbildung 1.2). Das Logo wird in sieben Elemente eingeteilt. Alle Objekte werden jeweils in eine Filmsequenz formatiert, jedes in seine eigene Ebene: Claudia und Black jeweils als eine Filmsequenz, wie auch die drei Balken und der Untertitel an sich. Das Claudia und das Black sollen nun langsam von oben eingeblendet werden. Die Balken kommen wie der Untertitel von links nach rechts langsam hinzu, alle Objekte werden von Alpha 0% zu Alpha 100% mittels Motiontweens eingeblendet. Entnehmen Sie den Verlauf der Animation den nachfolgenden Bildern.



■ **Abbildung 1.7:** Claudia Black und der Untertitel blenden sich ein.

Sie sollten sich jetzt bei Bildposition 110 befinden. Das heißt, dass Sie schon um die 110 Schlüsselbilder und leere Schlüsselbilder eingesetzt haben. Sie finden die aktuelle Bildposition oberhalb der Zeitleiste wie in der folgenden Abbildung.



■ ■ ■ *Abbildung 1.8: Die Zeitleiste von Flash MX*

Lassen Sie jetzt alle Elemente über 15 Bilder wieder auf Alpha 0% laufen. Sie können das sehr einfach bewerkstelligen, indem Sie in der obersten Ebene das letzte leere Schlüsselbild markieren und mit gedrückter **[⇩]**-Taste das letzte leere Schlüsselbild der untersten Ebene anwählen. Anschließend drücken Sie die Taste **[F6]** und in jeder Ebene sind neue Schlüsselbilder eingefügt. Nun markieren Sie wieder alle Schlüsselbilder des Anfangs dieser Animation und weisen Sie ihnen als Effekt Alpha 0%.

Nachdem auf der Hauptbühne kein Objekt mehr angezeigt wird, kann der Aufbau der Hauptseite beginnen. Die nachfolgende Abbildung zeigt die fertig gestellte Seite, ohne Texte. Die Entwicklung der Sequenz wird nun erfolgen.



■ ■ ■ *Abbildung 1.9: Die Hauptseite, unser nächstes Ziel auf dem Weg der Fertigstellung*

Die große graue Fläche wird nun in der Mitte von einer schwarzen Linie durchtrennt, diese fährt ganz simpel von Alpha 0% auf Alpha 100%. Danach erscheinen zwei weitere Elemente wie in der folgenden Abbildung.

Abbildung 1.10: ■■■
*Die graue Fläche wird
mit drei Objekten
gefüllt.*



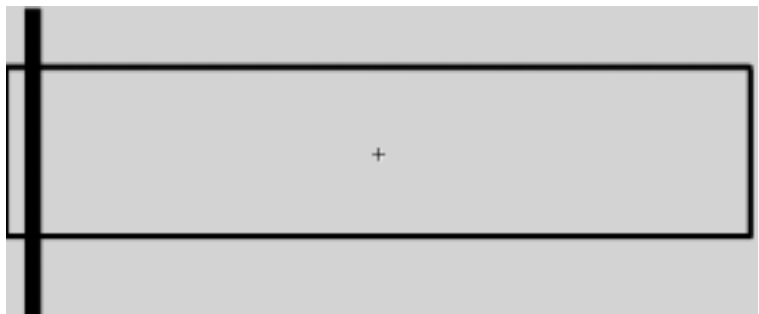
Die obere Linie wurde bereits in der Abbildung 1.9 angesprochen, die untere fährt von rechts außen in die Mitte und ebenfalls von Alpha 0% auf 100%. Währenddessen blendet sich die linke „Preisbox“ mit einem Alpha Motiontween ein; dieses Element hat lediglich eine gestalterische Funktion, die Entwicklung geschieht wie folgt:

Abbildung 1.11: ■■■
Die Preisbox



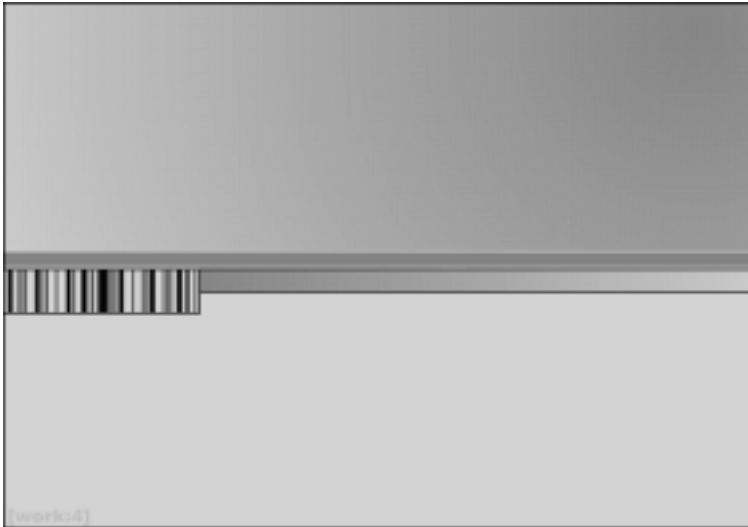
Zeichnen Sie einen ungefüllten Kasten in einer Form ähnlich der Abbildung. Nun zeichnen Sie viele einzelne „Vierecke“ ein. Wichtig ist, dass Sie keine einfachen Striche einsetzen und dass Sie alle Kästchen in eine neue Ebene zeichnen. Zeichnen Sie die Kästchen einfach wie in der folgenden Abbildung.

Abbildung 1.12: ■■■
*Ein Kasten ist links
außen bereits
ingezeichnet.*



Zeichnen Sie nun weitere Kästen. Wenn Sie in etwa die Hälfte ausgefüllt haben, kopieren Sie den Rest einfach mit in die Box. Alles, was über den Rahmen hinausragt, markieren und entfernen Sie.

Fügen Sie die Preisbox als eine Filmsequenz in etwa an der Position ein, die Sie in den vorangehenden Abbildungen sehen konnten. Nun soll die Textbox langsam von Alpha 0% auf Alpha 100% fahren.



■ ■ ■ *Abbildung 1.13: Drei weitere Elemente sind auf der Hauptseite hinzugekommen.*

In der Abbildung 1.13 sind drei Elemente hinzugekommen. Eine Ausfüllung für den Bereich oben, darunter ein Balken und die Ausfüllung der zwei schwarzen Linien.

Beginnen wir mit der obersten Ausfüllung. Zeichnen Sie ein Viereck, das den gesamten Bereich füllt, in eine neue Ebene. Füllen Sie den Kasten mit einem „Kreisförmigen Farbverlauf“ aus, klicken Sie aber nicht in die Mitte, wie Sie es wahrscheinlich normalerweise mit einem Farbverlauf machen, sondern oben rechts. Mischen Sie den Farbverlauf, mit Dunkelgrün innen (#9C9A00) und Hellgrün außen (#CECF63). Formatieren Sie das Feld zu einer Filmsequenz. Das Feld soll jetzt „hineingeblitzt werden“. Dazu fügen Sie fünf leere Schlüsselbilder nach dem ersten Erscheinen des Feldes ein. Im ersten Schlüsselbild stellen Sie den Wert des Feldes für den Effekt Helligkeit auf 100%. Anschließend fügen Sie ein Motiontween hinzu. Sehen Sie sich den Effekt an und Sie werden merken, das Bild kommt wie ein Blitz auf die Hauptbühne.

Der nun folgende mittlere Balken ist in einem einfachen Dunkelgrün (#9C9A00) gehalten und wird ebenfalls wie ein Blitz eingeblendet.

Der unterste Balken wechselt einfach per Formtween von Grau (#CECFCE) seine Farbe zu einem geraden Farbverlauf, der links dunkelgrün (#9C9A00) und rechts hellgrün (#CECF63) ist. Formatieren Sie dieses Viereck nicht zu einer Filmsequenz, sondern belassen Sie es als einfaches Objekt.

Abbildung 1.14: ■■■
Es sind sechs weitere
Elemente hinzu-
gekommen.



Nun sind sechs Elemente hinzugekommen. Zum einen oben rechts das Logo unserer Klientin sowie das große CB im Hintergrund, zum anderen die vier Punkte der Navigation.

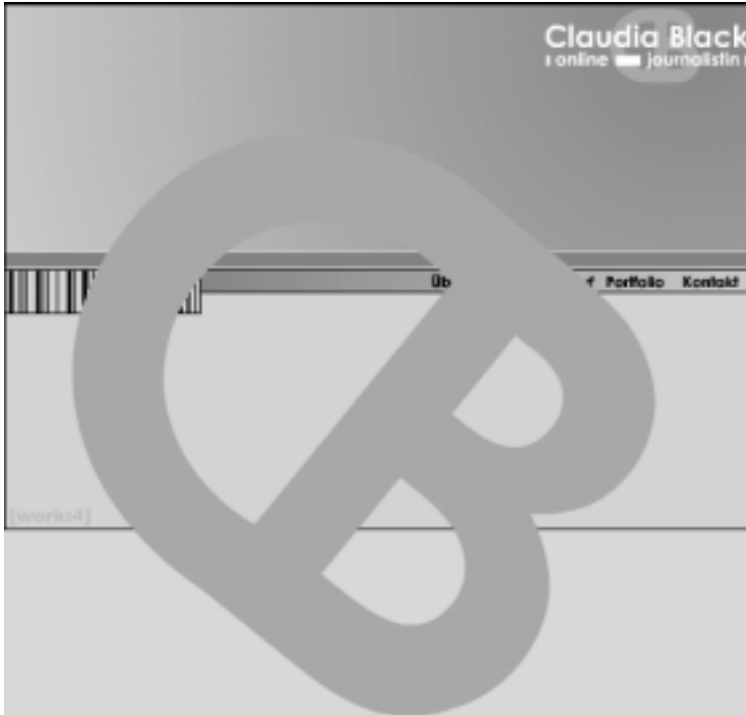
Beginnen wir mit dem Logo. Kopieren Sie dazu das Logo von der CD, teilen Sie es jedoch in zwei Bestandteile auf: Zum einen das CB, formatiert in eine Filmsequenz; die Sie in eine neue Ebene legen. Zum anderen der gesamte Rest; den Sie ebenfalls in eine Filmsequenz formatiert in eine neue Ebene legen.

Abbildung 1.15: ■■■
Der Aufbau von
Claudia Blacks Logo.



Zuerst wird das Claudia Black mit dem Untertitel „Online Journalistin“ mit einem Alphatween eingblendet. Das CB kommt danach wieder mit unserem Blitzeffekt auf die Seite.

Das CB für den Hintergrund nehmen wir uns jetzt aus dem Logo heraus. Vergrößern Sie das CB, so wie in der nachfolgenden Abbildung dargestellt.



■■■ **Abbildung 1.16:** Das CB ist jetzt so groß, es ragt sogar aus der aktiven Fläche heraus.

Sperren Sie nun alle Ebenen, außer derjenigen, welche das CB beinhaltet. Teilen Sie das CB. Das heißt, dass es keine Filmsequenz mehr ist. Dies können Sie mit der Tastenkombination `[Strg] + [B]` erledigen. Nun werden die unsichtbaren Teile des CB herausgeschnitten, wie im nachfolgenden Bild.

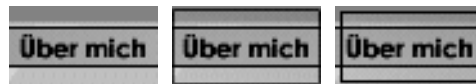


■ **Abbildung 1.17:** Die Überschnitte des CB wurden entfernt.

Kopieren Sie das C in eine neue Ebene und färben Sie den unteren Teil dunkelgrau (#9C9A9C), danach formatieren Sie beide Objekte jeweils zu Filmsequenzen. Weisen Sie beiden Teilen einen Alphawert von 40% zu.

Die beiden Teile kommen jetzt von außen auf die Hauptbühne. Der obere Teil von links, der untere von rechts. An ihrer Position angelangt kommt noch ein kurzer Blitzeffekt.

Beginnen wir nun mit der optischen Gestaltung der Navigation. Das Menü ist aufgeteilt in vier Knöpfe. Die Knöpfe werden in kleinen Animationen aktiviert:



■ **Abbildung 1.18:** Menübeispiel, erklärt an dem „Über mich“-Knopf

Hinter einem Schriftzug mit der Bezeichnung des Knopfs (hier „Über mich“) befindet sich unter „Maus Move“ eine Filmsequenz. Diese Filmsequenz beinhaltet einen Kasten, der zu Beginn groß ist und kleiner wird. In der Praxis sieht das so aus:

Der Besucher geht mit seiner Maus auf den „Über mich“-Knopf. Im Hintergrund ist die Box der Filmsequenz. Von Alpha 0% fährt sie in fünf

Schlüsselbildern auf Alpha 100%. Dabei wird die Box kleiner, so dass sie gerade um die Bezeichnung des Knopfs passt.

Zeichnen Sie nun eine Box wie in der Abbildung 1.19. Die Größe ist erst einmal nicht wichtig, das Aussehen sollte sich jedoch an der folgenden Abbildung orientieren.



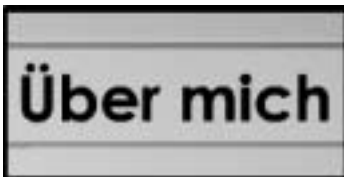
■ ■ ■ **Abbildung 1.19:**
Box für den Hintergrund eines Knopfs

Die Box wird mit einem grünen Farbverlauf ausgefüllt. Das dunkle Grün hat den Hex-Code #9C9A00 und das helle #CECF63.

Markieren Sie nur die gefüllte Fläche, ohne den Rahmen, und formatieren Sie die Fläche in eine Filmsequenz. Formatieren Sie danach auch den Rahmen in eine Filmsequenz. Weisen Sie der Filmsequenz mit der ausgefüllten Fläche den Alpha-Wert von 40% zu.

Jetzt werden beide Filmsequenzen in eine neue formatiert, diese soll mit einem Motiontween sichtbar werden.

Strecken Sie die Filmsequenz jetzt so, dass sie gerade um das „Über mich“ passt.



■ ■ ■ **Abbildung 1.20:**
Der „Über mich“-Knopf

Kopieren Sie die Filmsequenz auch für die anderen drei Knöpfe.

1.2 Hauptseite und Navigation

Die Seite soll drei wesentliche Inhalte haben: einen Text über Claudia Black, wer sie ist und vor allem, was sie beruflich macht, dazu ihren Lebenslauf und ein Portfolio mit einem Auszug ihrer Kunden. Daneben gibt es natürlich noch die Möglichkeit, eine E-Mail an Claudia Black zu schreiben. Dazu werden unter „Kontakt“ ihre E-Mail-Adresse sowie drei Textboxen eingeblendet.

Alle Inhalte sollen eingeblendet und später wieder ausgeblendet werden. Wechselt der Besucher also zwischen verschiedenen Texten, wird der alte Inhalt abgebaut und der neue danach aufgebaut.





■ ■ ■ Abbildung 1.21: Eine Kategorie wird ausgeblendet, bevor eine neue eingeblendet wird.

Folgende vier Kategorien soll es also geben:

- Über mich
Begonnen wird hier mit einem kleinen Intro, die Wörter „Ich denke, also...“ werden nacheinander mit ActionScript eingeblendet. Dazu wird jeder einzelne Buchstabe sichtbar und verschwindet nach kurzer Zeit. Danach blendet sich als Fortsetzung „muss ich sein.“ ein. Der Spruch ist an das Zitat *Ich denke, also bin ich* von René Descartes angelehnt. Darauf folgend werden Textzeilen eingeblendet, die allesamt Aussagen über Claudia Black beinhalten sollen.

Abbildung 1.22: ■■■
Inhalt „Über mich“

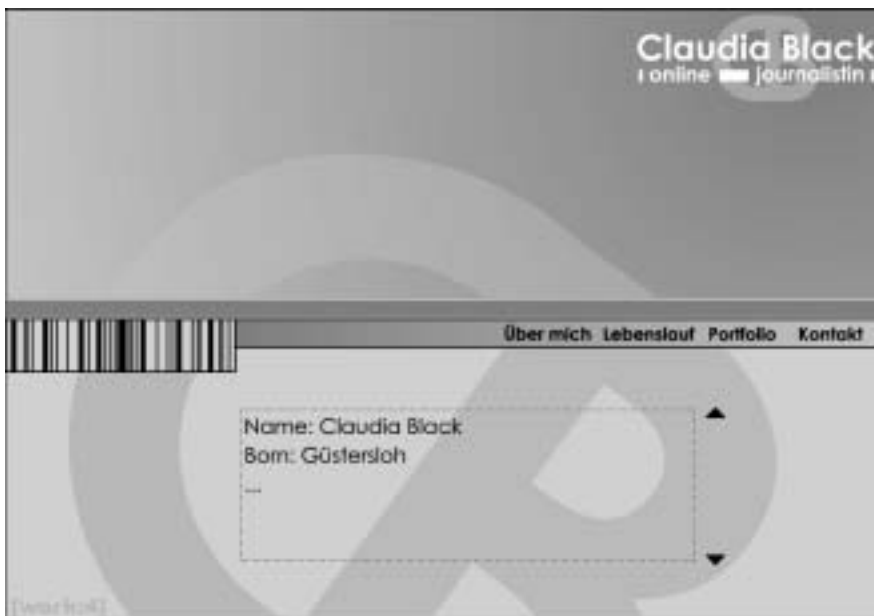


- Lebenslauf

Hier wird der Lebenslauf eingeblendet. Da wir davon ausgehen können, dass dieser recht lang sein wird, gibt es hier die Möglichkeit, den Text hinunterzuscrollen.



Born in Gütersloh



■ ■ ■ *Abbildung 1.23: Der Lebenslauf von Claudia Black*

- Portfolio
Im Portfolio werden die Logos dreier Kunden von Claudia Black eingeblendet. Für unser Beispiel nehmen wir [work:4] new media, Netzproduktion und den Addison-Wesley-Verlag.



■ ■ ■ Abbildung 1.24: Portfolio

Auf der CD finden Sie alle in diesem Kapitel verwendeten Logos als Vektoren, in der Datei LOGO.FLA.



- Kontakt

Hier wird Claudia Blacks E-Mail-Adresse eingeblendet. In der oberen Zeile steht: „Nehmen Sie Kontakt mit mir auf. Ich freue mich!“ Die E-Mail-Adresse darunter soll sich durch ihre Größe eindeutig abheben und gut einprägen. Darunter befindet sich ein Kontaktformular, das mit PHP-Unterstützung eine Mail an Claudia Black schickt. Die E-Mail-Adresse wird mit einem PHP-Skript überprüft und falls sie falsch scheint, wird eine Fehlermeldung zurückgegeben.



■ ■ ■ **Abbildung 1.25: Das Kontaktformular zu Claudia Black**

Damit der Inhalt eingeblendet und später wieder ausgeblendet werden kann, müssen wir ein etwas komplexeres System einsetzen. Wird z.B. der Inhalt von „Über mich“ gerade angezeigt und der Besucher möchte jetzt den Lebenslauf sehen, soll der aktuelle Inhalt ausgeblendet und dann zum Lebenslauf gesprungen werden. Dazu gibt es nach dem alten Inhalt, dem Text unter „Über mich“, einen Schalter, der zum neuen Inhalt wechselt. Was real nur ein Schlüsselbild ist, benennen wir „Switch“. Die Funktion entwickeln wir wie folgt:

Jeder „Clip“ bekommt einen Start, wo er aufgebaut wird, und einen Abbau, von wo an er wieder ausgeblendet wird. Danach wird zu dem neuen Clip gewechselt. In unserem Szenario sieht das so aus: Der Besucher hat den Inhalt der Kategorie „Über mich“ auf seinem Bildschirm und klickt auf „Lebenslauf“. Nun wird das Schlüsselbild mit der Bezeichnung „Über mich – Abbau“ abgespielt, bis der Inhalt komplett verschwunden ist. Danach kommt einer unserer Switch-Frames und springt zum Frame mit der Bezeichnung „Lebenslauf – Start“. Nachdem der neue Inhalt aufgebaut ist, wird gestoppt.

Damit diese Methode funktioniert, brauchen wir zwei Variablen: `target_left` und `target`. `target` bedeutet auf Deutsch „Ziel“ und gibt an, was aufgerufen werden soll (hier der Lebenslauf), `target_left` ist der aktuelle Clip, der abgebaut werden soll.

Die Funktion des Lebenslauf-Knopfs im Menü sieht wie folgt aus:

Listing 1.1: ■■■
Klick zum Switch

```
on (press) {
    target_left = target;
    target = "Lebenslauf";
    if (target_left != target) {
        gotoAndPlay (target_left add " - Abbau");
    }
}
```

Der aktuelle Wert von `target` ist „Über mich“, welchen `target_left` übernimmt. Der neue `target`-Wert ist dann „Lebenslauf“. In der `if`-Abfrage wird überprüft, ob `target_left` und `target` ungleich sind. Wenn dem so ist, wird zu dem Schlüsselbild „Über mich – Abbau“ gesprungen.

Der Switch ist sehr einfach aufgebaut:

Listing 1.2: ■■■
Der Switch

```
gotoAndPlay (target add " - Start");
```

Flash springt an dieser Stelle zu dem Schlüsselbild „Lebenslauf – Start“.

1.3 Text- und Bildelemente ein- und ausblenden, bewegen und skalieren

Nachdem Sie die Grundgrafiken fertig gestellt und das System kennen gelernt haben, beginnen wir mit der Entwicklung der Spots mit den Inhalten. Achten Sie darauf, dass Sie das vorhergehende Switch-System beibehalten.

Die Begrüßung

Im Gegensatz zu vielen anderen Webseiten begrüßt Claudia Black ihre Besucher. Dazu reicht ein schlichtes „Claudia Black welcomes you“, danach wird der Inhalt eingeblendet.

Legen Sie ein neues Textfeld an, in das Sie einen beliebigen Text eintragen. In der Beispielseite habe ich wie folgt begonnen: „Werbung ist mehr als die Anhäufung“, was natürlich in der zweiten Zeile fortgesetzt wird. Kopieren Sie dieses Textfeld mit `[Strg] + [C]` und fügen Sie es mit `[Strg] + [⇧] + [V]` in eine neue Ebene wieder ein. Jetzt ist es exakt an derselben Position wie das Original. Markieren Sie es und schieben Sie es mit der `[↓]`-Taste herunter. Nun wird der Text fortgesetzt: „bunter Bilder. Werbung machen ist“, was natürlich auch noch fortgesetzt werden muss. Es reicht aber, wenn Sie es jetzt bei diesen beiden Zeilen belassen.

Die Zeilen sollen nun langsam mit einem Motiontween von Alpha 0% auf Alpha 100% fahren.

Legen Sie eine Ebene an, die Sie „Kontrolle“ nennen. In diese Ebene legen wir alle Switches und weiteren ActionScripts. So bleibt es für uns übersichtlich. In welcher Ebene Sie das ActionScript aber einsetzen, ist für Flash ohne Bedeutung. Fügen Sie also in einem Schlüsselbild, das sich an derselben Position befindet wie die auf Alpha 100% stehende letzte Filmsequenz, ein leeres Schlüsselbild mit folgendem Skript ein:

```
target = "Home";  
stop();
```

Von hier an kann der Benutzer bei der fertigen Seite auf die Knöpfe klicken.

■■■ Listing 1.3:
Bestimmung des
Anfangszustands

Spot: Über mich





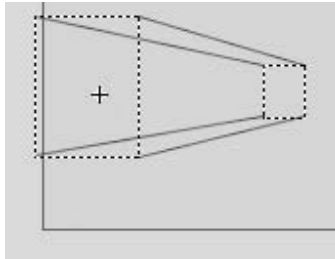
■ ■ ■ *Abbildung 1.26: Ich denke, also muss ich sein.*

Nachdem zuerst jeder Buchstabe der Einleitung „Ich denke, also...“ eingeblendet und auch wieder ausgeblendet wurde, blendet sich die Zeile „muss ich sein“ ein. Danach werden die einzelnen Zeilen des Textes eingeblendet. Beginnen wir mit der Einleitung.

Eine solche Animation einzelner Buchstaben benötigt natürlich kein ActionScript. Um aber Zeit einzusparen, setzen wir ActionScript ein. Zuerst legen Sie jedoch ein leeres dynamisches Textfeld mit der Variablen `text` an. Formatieren Sie dieses Objekt in ein Bild.

Nun formatieren Sie das Bild in eine Filmsequenz. Diese wird mit ActionScript dupliziert, das Textfeld wird immer einen Buchstaben beinhalten, der in der Filmsequenz animiert wird. Diese Animation entwickeln wir jetzt.

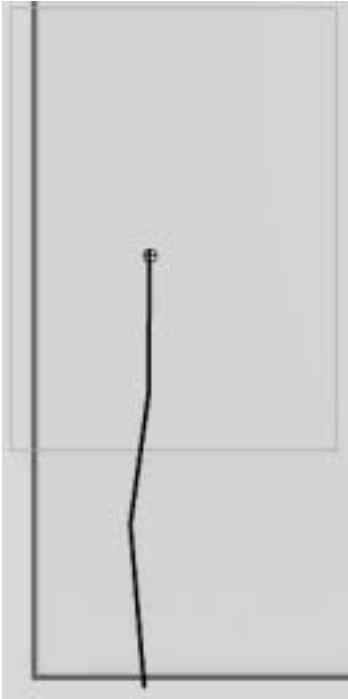
Der Effekt soll relativ langsam abgespielt werden, setzen wir also gut 100 Schlüsselbilder (knapp fünf Sekunden) für die Länge der Animation an. Zuerst wird das Textfeld über rund 15 Schlüsselbilder eingeblendet:



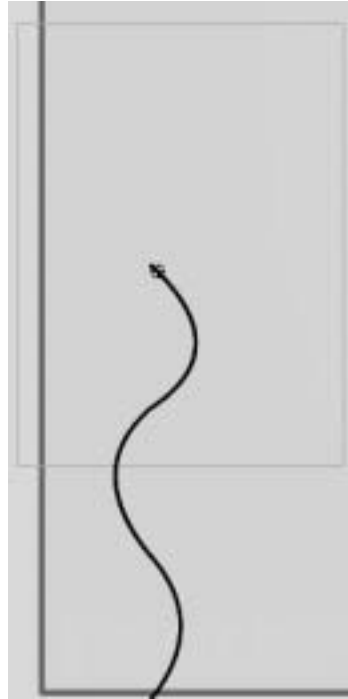
■ ■ ■ **Abbildung 1.27:** Rechts ist das Textfeld verkleinert, links vergrößert.

Nun soll eine Pause folgen, im Beispiel 45 Bilder. Das ist genug Zeit, um den Buchstaben zu lesen, der nach dem Verstreichen dieser Bilder wieder ausgeblendet wird.

An diesem Punkt beginnen wir die Arbeit mit einer Führungsebene. Legen Sie eine neue Ebene an und zeichnen Sie folgendes Linienkonstrukt.



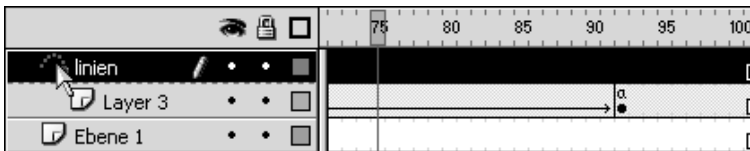
■ ■ ■ **Abbildung 1.28:**
Linienkonstrukt



■ ■ ■ **Abbildung 1.29:**
Geschwungene Linien

Orientieren Sie sich bei dem Beginn der Zeichnung am Mittelpunkt der Grafik mit dem Textfeld. Die letzte Linie sollte knapp aus dem Gesamtbild heraus führen.

In der zweiten Abbildung sind die Linien schon geschwungen, machen Sie es ebenso. Die Ebene mit den geschwungenen Linien wird nun als eine Führungsebene bestimmt (auf Englisch: Guide). Um zu den Eigenschaften der Ebenen (Layer Properties) zu gelangen, klicken Sie doppelt auf das Symbol vor dem Namen der Ebene.



■ ■ ■ **Abbildung 1.30:** Das Symbol vor der Ebene „Linien“ zeigt, dass es sich um eine Führungsebene handelt.

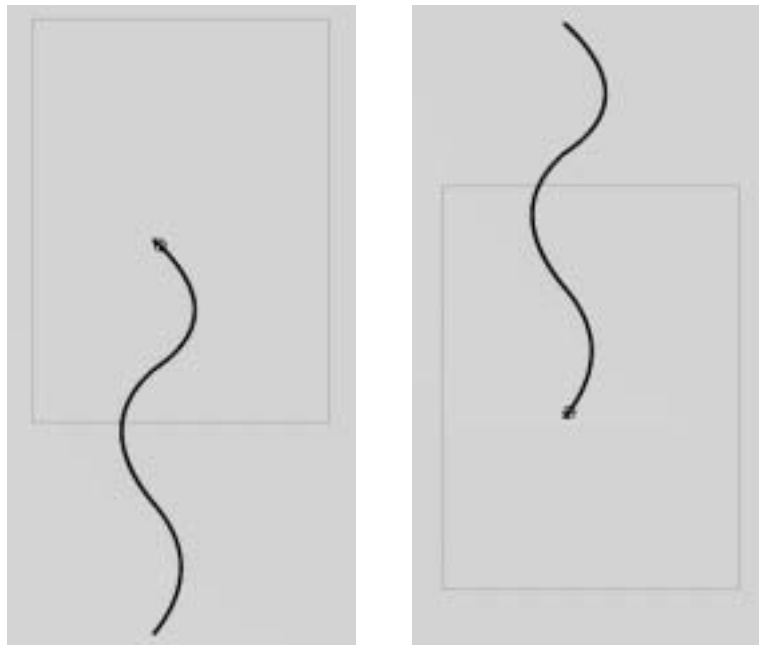
Nachdem Sie doppelt geklickt haben, erscheint folgendes Fenster:



■■■ *Abbildung 1.31: Die Layer Properties sind nun erschienen.*

Durch die Aktivierung der „Guide-Option“ wird unsere Ebene zu einer Führungsebene.

Die Filmsequenz des Buchstabens soll in Schlangenlinien ablaufen, diese Linien werden in der Führungsebene festgelegt. Durch die von uns ausgeführten Schritte wird der Buchstabe an der Schlangenlinie ausgerichtet, orientiert sich beim Ablauf der Animation also an ihrer Linienführung. Damit dies aber für unseren Buchstaben auch aktiviert wird, muss die Ebene mit Buchstaben als „Guided“ aktiviert werden.



■■■ *Abbildung 1.32: Das Textfeld ist mit seiner Mitte an den Schlangenlinien ausgerichtet.*

Fügen Sie in das erste und das letzte Bild dieser Filmsequenz ein `stop()`; ein. Diese Sequenz soll nun mit ActionScript kopiert und aneinander gereiht werden. Damit das gleich folgende ActionScript greifen kann, muss die Filmsequenz einen Instanznamen bekommen. Wählen Sie als Namen „letter“ – zu Deutsch: Buchstabe. Positionieren Sie den Buchstaben nahe dem linken Rand. Wie genau Sie ihn am besten positionieren, erfahren Sie durch Testversuche.

Die Animation ist in fünf Schlüsselbildern entwickelt:

1 Schlüsselbild 1, Text definieren

```
text = "Ich denke, also...";
```

Hier wird einfach die Variable `text` gesetzt. Diese Variable beinhaltet, wie der Name schon sagt, den Text, der animiert werden soll.

■ ■ ■ *Listing 1.4:
Festlegen einer
Variablen*

2 Schlüsselbild 2, ActionScript programmieren

Das nun folgende ActionScript wird schon etwas komplexer.

```
textlength = length(text);
i = "1";
while (Number(i)<=Number(textlength)) {
    duplicateMovieClip ("letter", "letter" add i, i);
    setProperty ("letter", _x, Number(getProperty("letter",
_x))+28);
    tellTarget ("letter" add i) {
        txt = substring(..:text, ..:i, 1);
    }
    i++;
}
```

■ ■ ■ *Listing 1.5:
Das Skript dupliziert
und positioniert Movie-
Clips.*

Die Variable `textlength`, welche zuerst besetzt wurde, liest die Länge (`length`) der Variablen `text` aus. „Ich denke, also...“ umfasst 18 Zeichen (Leerzeichen sind mitberechnet). Es werden dementsprechend später auch 18 Kopien der Filmsequenz angelegt.

```
i = „1“;
```

Die Variable `i` wird mit 1 besetzt, sie ist sehr wichtig für die Namensgebung der Filmsequenzen – mehr dazu später.

```
while (Number(i)<=Number(textlength)) {
    ...
}
```

Eine While-Schleife wiederholt den Inhalt, der in geschweiften Klammern steht, bis zum Erreichen der eben genannten Endbedingung. In diesem Skript initiiert `while` eine Schleife, die läuft, so lange der Wert der Variable `i` kleiner ist als der Wert der Variablen `textlength`.

```
duplicateMovieClip ("letter", "letter" add i, i);
```

`duplicateMovieClip` dupliziert eine Filmsequenz. Die Funktion ist wie folgt aufgebaut:

```
duplicateMovieClip (Target, New Name, Depth)
```

Für uns sind eigentlich nur `Target` und `New Name` interessant. Der Wert `Target` gibt an, welche Filmsequenz kopiert werden soll (in diesem Fall "letter"). `New Name` gibt den neuen Namen der Filmsequenz an, hier: "letter" add `i`. An das Wort „letter“ wird der Wert der Variablen `i` mit angehängt, z.B. `letter8`.

Diese Art der Bezeichnung ist notwendig, da alle Filmsequenzen im Nachhinein von uns wieder angesprochen werden sollen. Die Bezeichnung `letter8` ist hier die neue Instanzbezeichnung.

`Depth` ist für uns ohne Belang, wir geben auch hier einfach `i` an.

```
setProperty ("letter", _x, Number(getProperty("letter",
_x))+28);
```

Die Funktion `setProperty` beeinflusst gezielt eine Filmsequenz, hier jene mit dem Namen des Wertes der Variablen: "letter" wird hier in seiner X-Koordinate verschoben. Mit `getProperty` wird der aktuelle Wert ausgelesen, dem dann 28 hinzuaddiert werden. Die 28 sollten Sie durch Testversuche, nach Ihrem Belieben, ergänzen.

```
tellTarget ("letter" add i) {
    txt = substring(..:text, ..:i, 1);
}
```

Nun wird die aktuelle Filmsequenz angesprochen und das enthaltene Textfeld mit dem Namen „txt“ wird mit dem aktuellen Buchstaben gefüllt. Sind wir bei `i = 8`, also `letter8`, wird der achte Buchstabe eingesetzt, hier „k“.

```
i++;
```

Die bereits erwähnte Variable `i` wird um eins erhöht.

3 Schlüsselbild 3

```
i = 1;
```

4 Schlüsselbild 4 und los geht's

Listing 1.6: ■■■
Die Variable `i` setzen
wir wieder auf eins.

Listing 1.7: ■■■
Es wird `letter + i`
angesprochen.

```
tellTarget ("letter" add i) {
    play ();
}
i++;
```

Jetzt wird jede Filmsequenz angesprochen und mittels `play()`; abgespielt. Begonnen wird mit `letter1`, danach folgt `letter2`, die Instanznamen setzen sich aus „letter“ und dem Wert der Variablen `i` zusammen.

Benennen Sie dieses Schlüsselbild „Text-Start“.

5 Schlüsselbild 5, Kontrolle

```
if (Number(i)<=Number(textlength)) {
    gotoandplay("Text-Start");
}
```

■ ■ ■ *Listing 1.8:*
Kontrolle

Mit diesem Skript wird sichergestellt, dass `i` kleiner oder genauso groß ist wie `textlength`, was uns garantiert, dass jeder Buchstabe auch angesprochen wird. Ist dies der Fall, springt Flash zurück zum Schlüsselbild 4 (Text-Start).

6 Schlüsselbild 6, kurze Pause

Nennen Sie dieses Schlüsselbild „check-up“, denn nun soll noch eine kurze Pause folgen. In Schlüsselbild 6 wird dazu die Variable `a` erhöht:

```
a++;
```

■ ■ ■ *Listing 1.9:*
Erhöhung der Variable
a um 1

7 Schlüsselbild 7

In diesem Schlüsselbild wird die Variable `a` überprüft:

```
if (a eq 100) {
    nextFrame;
} else {
    gotoandplay("check-up");
}
```

■ ■ ■ *Listing 1.10:*
Überprüfen von „a“

Ist der Wert von `a` gleich 100 ist, springt Flash zum nächsten Schlüsselbild. Ist dies jedoch nicht der Fall, wird zu Schlüsselbild 6 (check-up) gesprungen.

Nachdem die Textanimation abgeschlossen ist, wird „muss ich sein“ eingeblendet. Ziehen Sie dazu einfach ein großes Textfeld in die Hauptbühne, das langsam eingeblendet und wieder ausgeblendet wird.

Der nachfolgende Text blendet sich Zeile für Zeile ein und später wieder aus. Die einzelnen Linien verschieben sich nach außen, die erste nach links, die zweite nach rechts usw.

Spot: Lebenslauf





■ ■ ■ *Abbildung 1.33: Lebenslauf*

Der folgende Spot beginnt mit einem sich einblendenden „Born in Gütersloh“.

Zuerst zeichnet sich das „Born“, danach folgt das Einfliegen der Buchstaben i und n. Den Abschluss bilden dann die Einblendung von „Gütersloh“ und die Einfärbung des gesamten Schriftzugs.

Nachdem die Animation fertig ist, wird eine dynamische Textbox mit dem Lebenslauf eingeblendet, das Textfeld wird scrollbar sein.



■ **Abbildung 1.34: Born in Gütersloh**

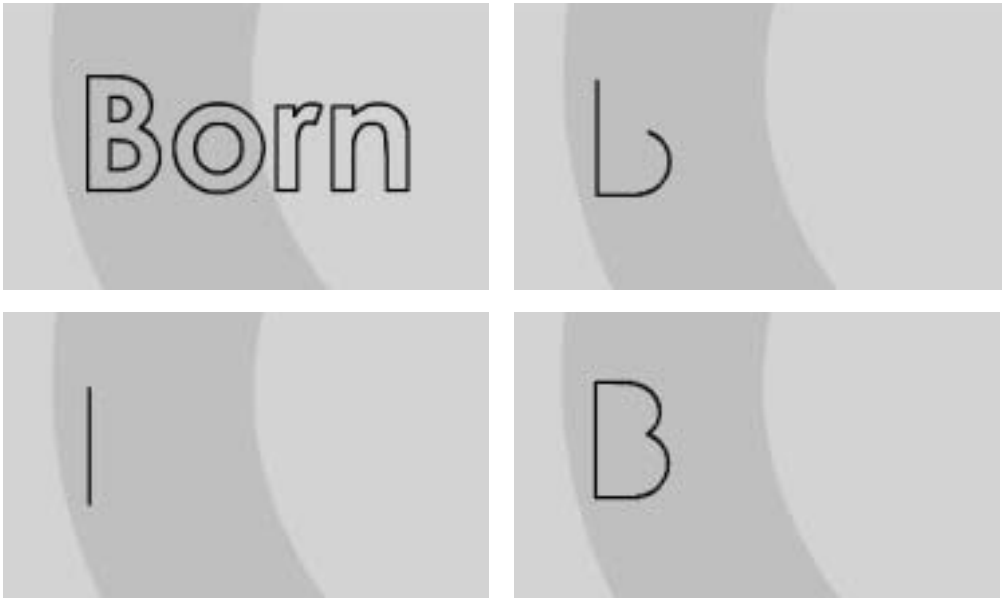
Legen Sie ein Textfeld auf die Hauptbühne und schreiben Sie „Born in Gütersloh“ hinein. Markieren Sie das Textfeld und drücken Sie `[Strg] + [B]`. Dadurch wird es geteilt, die einzelnen Buchstaben werden zu einzelnen Vektoren.



Mit dem Tintenfasswerkzeug (Icon in der Abbildung 1.35) können Sie jeden Vektor umranden. Dazu müssen Sie jedoch jeden einzelnen Rahmen malen. Klicken Sie dazu in den äußeren und – je nachdem – in den inneren Bereich des Vektors (nicht auf den Hintergrund).

■ **Abbildung 1.35: Das Tintenfasswerkzeug**

Nachfolgend werden wir zuerst das „Born“ umsetzen:



■ ■ ■ **Abbildung 1.36: Born**

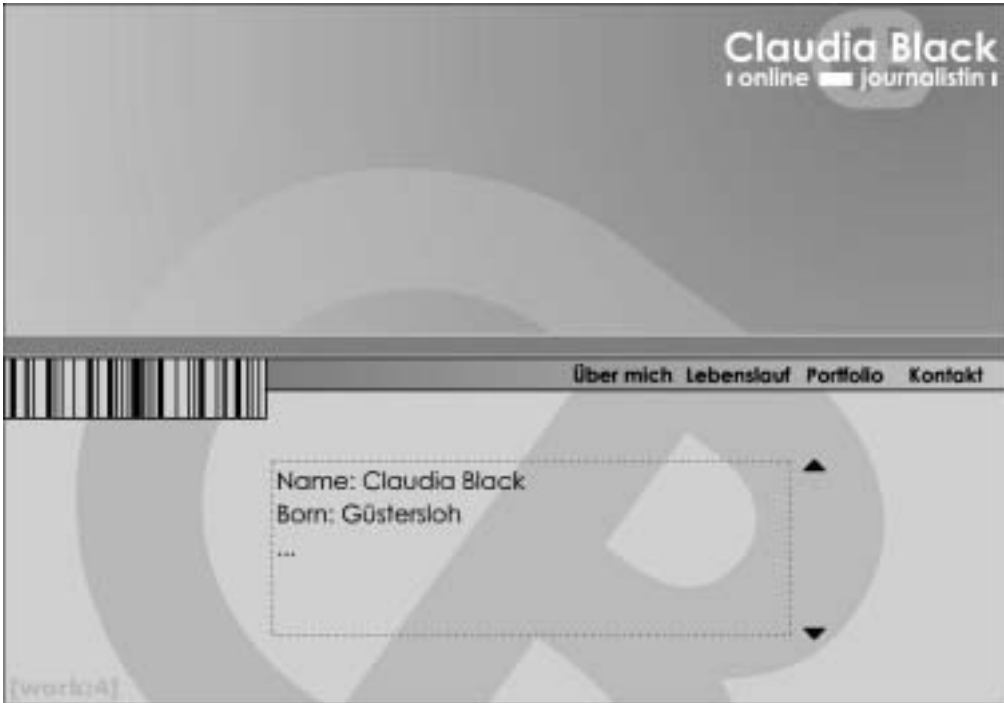
Kopieren Sie das „Born“ in ein Schlüsselbild und duplizieren Sie es einfach ein paar Mal. Danach schneiden Sie zuerst die letzten drei Buchstaben weg und dann einen Teil des B. Im nächsten Schlüsselbild verfahren Sie ebenso, jedoch nur mit einem kleineren Teil des B, so dass sich im Verlauf das B aufbaut.

Das i und das n formatieren Sie einfach in Filmsequenzen und lassen sie von links außen einfliegen, während sich die Umrandung von „Gütersloh“ ganz simpel einblendet.

Die Füllung erfolgt einfach wieder mit den Wörtern – legen Sie ein Textfeld an mit „Gütersloh“ als Inhalt und platzieren Sie es eine Ebene unter die Umrandung von Gütersloh. Per Alphasweeps kommt es in das Bild, genauso wie bei dem „Born“ und dem „in“.

Legen Sie nach einer kurzen Pause in jeder Ebene ein neues Schlüsselbild an, formatieren Sie dann den gesamten Schriftzug zu einer Filmsequenz und blenden Sie ihn aus.

Für das Feld, das letztendlich den Lebenslauf enthält, legen Sie ein dynamisches Textfeld mit mehreren Zeilen und automatischem Wortumbruch an. Im Weiteren muss HTML aktiviert werden. Weisen Sie dem Textfeld die Variable `text` zu.



■ **Abbildung 1.37: Lebenslauf**

Wie in der vorangegangenen Abbildung legen Sie das Textfeld etwa in der Mitte der grauen Ebene an. Zeichnen Sie rechts daneben zwei Objekte, die die Scrollfähigkeit symbolisieren, eins nach oben und eins nach unten ausgerichtet, beispielsweise Pfeile.

Formatieren Sie das Textfeld in eine Grafik (nicht in eine Filmsequenz!) und die zwei Pfeile zusammen in eine Filmsequenz!

Öffnen Sie die Filmsequenz und wählen Sie den Pfeil, der nach oben zeigt. Weisen Sie folgendes ActionScript zu:

```
on (press, rollover, dragOver, dragOut, keyPress "<Up>") {
    _root.text.scroll = Number(_root.text.scroll)-1;
}
```

■■■ *Listing 1.11:*
Scrollfunktion nach oben

Durch diese Funktion wird nach oben gescrollt, der Text bewegt sich also nach unten. Um nach unten zu scrollen, wird im Gegensatz dazu einfach die Eigenschaftsvariable `scroll` wieder erhöht. Das sieht so aus:

```
on (press, rollover, dragOver, dragOut, keyPress "<Down>") {
    _root.text.scroll = Number(_root.text.scroll)+1;
}
```

■■■ *Listing 1.12:*
Scrollfunktion nach unten

Wenn Sie nun eine Kopie dieses Schlüsselbilds in dieselbe Ebene legen, erreichen Sie das Scrollen ohne Pause, wenn man mit der Maus auf einen Pfeil klickt.

Dank `keyPress` können auch die Pfeiltasten der Tastatur eingesetzt werden.

Um nun das Textfeld zu füllen, geben Sie folgendes ActionScript in ein Schlüsselbild ein:

```
text = "<u>Name:</u> Claudia Black";
```

■■■ *Listing 1.13:*
HTML-Syntax in Flash

Die Syntax `<u></u>` stammt aus HTML und wird benutzt, um etwas zu unterstreichen. Seit Flash 5 wird diese Art der Textformatierung unterstützt, aber nur bei Textfeldern, bei denen HTML aktiviert wurde.

Um eine weitere Zeile hinzuzufügen, wird `
` verwendet, im Beispiel:

```
text = "<u>Name:</u> Claudia Black<br><u>Wohnort:</u> Hamburg";
```

■■■ *Listing 1.14:*
Neue Zeile mit der HTML-Syntax einfügen

Spot: Portfolio



■ ■ ■ Abbildung 1.38: Portfolio

Im Portfolio werden einfach drei Filmsequenzen eingeblendet, jedes Logo kommt von oben nach unten. Die gesamte Animation ist eher langsam zu halten.

Da diese Animation für Sie keine besonderen Neuerungen beinhaltet, geschweige denn neues Fachwissen voraussetzt, welches wir in diesem Kapitel noch nicht behandelt haben, schließe ich diesen Teil des Kapitels jetzt schon ab.

Spot: Kontakt



■ ■ ■ **Abbildung 1.39: Das Kontaktformular**

Mit diesem Kontaktformular werden wir über PHP eine E-Mail an Claudia Black versenden. Des Weiteren wird es möglich sein, im normalen E-Mail-Programm eine Mail zu verfassen.

Dazu legen Sie einfach einen Knopf mit dem Text „Claudia-Black@work4.de“ an und fügen ihm folgendes ActionScript hinzu:

```
On (press) {
    getURL ("mailto:ClaudiaBlack@work4.de");
}
```

■ ■ ■ **Listing 1.15:**
E-Mail senden

Für das PHP-Formular wird die Entwicklung jedoch etwas komplexer.

Wir verwenden zwei Filmsequenzen: einmal die in der vorangegangenen Abbildung, welche den Senden-Knopf sowie die drei Textfelder enthält, zum anderen eine Nachricht, dass die E-Mail erfolgreich abgeschickt wurde.

Legen Sie drei Eingabetextfelder (Input-Text) an, mit folgenden Variablen: `user_name`, `user_email` und `message`. Zeichnen Sie zudem einen Knopf neben die Textfelder. Fassen Sie die Textfelder und den Knopf zu einer Filmsequenz mit der Instanz „`mail_block`“ zusammen.

Der Senden-Knopf überprüft nun zuerst, ob alle Textfelder einen Inhalt haben. Wenn dem so ist, geht es weiter.

Listing 1.16: ■■■
*Überprüfung der
Eingaben*

```
on (release) {
    if ((user_name.length>0) && (user_email.length>0) &&
        message.length>0)) {
        _root.play();
    } else {
        _root.send_status = "Nicht alle Felder wurden ausgefüllt!";
    }
}
```

Wenn also die Länge von `user_name`, `user_email` und `message` länger ist als 0, wird der Film fortgesetzt. Wenn nicht, meldet das Textfeld `send_status`, dass nicht alle Felder ausgefüllt sind. Legen Sie nun das entsprechende Textfeld in der Hauptebene an. Es sollte natürlich mehrere Zeilen unterstützen können.

Wenn nun also eine Mail geschrieben wurde und abgeschickt werden soll, muss in einem Schlüsselbild folgendes ActionScript stehen. Das ActionScript muss nach dem `Stop()`; eingesetzt werden, von wo an die E-Mail geschrieben werden kann:

Listing 1.17: ■■■
Aufruf der mail.php

```
email = "ClaudiaBlack@work4.de";
title = "ClaudiaBlack Website - Kontakt";
user_name = mail_block.user_name;
user_email = mail_block.user_email;
message = mail_block.message;
loadVariablesNum ("mail.php", 0, "GET");
mail_send = "true";
send_status = "";
```

Hier werden nur E-Mails ausgetauscht und dann an `MAIL.PHP` geschickt. `user_name`, `user_email` sowie `message` müssen zuerst wieder aus der Filmsequenz geholt werden, bevor sie abgeschickt werden können.

Das PHP-Skript wird die E-Mail an den Inhalt der Variablen `email` senden, mit dem Wert von `title` als Betreff. `mail_send` wird für das Ausblenden des späteren Switch benötigt. `send_status`, das Textfeld, wird wieder geleert, sollte etwas darin stehen.

Im nächsten Schlüsselbild wird die Hauptebene mit `stop()`; pausiert. Ab hier gibt es dann die zweite bereits angesprochene Filmsequenz.

Hier ist eine Filmsequenz, die eine Nachricht in der Art von „Ihre E-Mail wurde erfolgreich abgeschickt.“ vermeldet. Weisen Sie dieser Filmsequenz die Instanz `mail_erfolg` zu.

Ab dem Ausblendausschnitt muss nun folgendes ActionScript eingesetzt sein:

```
if (mail_send == "true") {
    mail_block._visible = false;
    mail_erfolg._visible = true;
    mail_send = "false";
} else {
    mail_block._visible = true;
    mail_erfolg._visible = false;
}
```

■ ■ ■ *Listing 1.18:*
Ausblendensequenz bestimmen

Wurde also die Mail gesendet, wird `mail_block` unsichtbar (`visible = false`) und `mail_erfolg` ist zu sehen. Falls keine E-Mail versendet wurde, bleibt der `mail_block` sichtbar.

Wurde eine E-Mail gesendet, aber der Besucher klickt wieder auf KONTAKT und danach woanders hin, wird der `mail_block` wieder ausgeblendet.

Die Datei `MAIL.PHP` hat letztlich noch folgenden Inhalt:

```
<?php
    mail($email, $title, $message, "From: $user_name
    <$user_email>\nReplyTo:$user_email");
?>
```

■ ■ ■ *Listing 1.19:*
Mailformular mail.php

Die Funktion `mail()`; sendet eine Mail an `$email` mit dem Betreff `$title` und der Nachricht `$message`.



ECards versenden (Flash und PHP)

Azad Adsay und Jürgen Lange

*„Alles Denken ist wesentlich optimistisch.
Der vollendete Pessimist würde verstummen
und – sterben.“*

*Christian Morgenstern
(1871–1914), dt. Lyriker*

In diesem Kapitel erstellen wir ein sehr einfaches, auf einer MySQL-Datenbank basierendes ECard-System. Dabei nehmen wir eine Unterteilung in den Flash- und in den PHP- und Datenbankprogrammierungsteil vor.

2.1 Der Flash-Film

Azad Adsay

Abbildung 2.1: ■■■
ECard, Ablauf



Unser Ziel ist es wie gesagt, eine Ecard-Anwendung zu entwickeln. Dabei handelt es sich um eine Art digitale Postkarte, die sich zu vielen Anlässen einsetzen lässt. Der Vorteil für den Betreiber einer Site mit ECard-Feature ist, dass es mehr Besucher gibt, da die ECard nur vom entsprechenden Server abgerufen werden kann. Ein zusätzliches Plus sind die geringen Erstellungskosten der ECards. Außerdem muss das System nicht gepflegt werden.

Der Zweck einer ECard ist schnell erklärt. Wenn zum Beispiel ein Freund Geburtstag hat, kann man ihm eine ECard schicken. Das geht schneller als eine normale Karte und ist meistens auch noch individueller.

Unsere ECard zeigt zwei Fische im Meer, die sich einen banalen Witz erzählen. Vorher werden der Gruß und eine private Nachricht angezeigt. Abschließend kann der Empfänger der Karte mit einer E-Mail oder auch einer anderen ECard reagieren oder unsere Seite besuchen.

Das Storyboard

Zuerst bedarf es einer gründlichen Planung. Für ein ECard-System benötigen wir eine Tabelle, in der Name und E-Mail-Adresse sowohl des Absenders als auch des Empfängers gespeichert werden. Hinzu kommt natürlich der Inhalt des Grußes und die ID, mit der die Einträge ausgelesen werden.

Die Werte der Tabelle werden über Parameter an Flash übergeben und sind anschließend für den Benutzer einsehbar. Nach der Begrüßung folgt der Inhalt der Karte und danach der Witz. Der Inhalt wird in kleinen Sprechbläschen der Fische wiedergegeben. Nachdem die zwei Fische gründlich gelacht haben, verschwinden beide wieder und der Spruch „Lach’ mal wieder“ erscheint, nachdem das Bild verdeckt worden ist.

Die gesamte Animation findet unter dem Meeresspiegel statt, wir werden also eine komplette Unterwasserwelt aufbauen und diese auch noch mit Sound hinterlegen.

Nach dem Szenario im Meer wird Werbung für BookWorker eingeblendet und der Besucher kann mit einer ECard oder einer E-Mail antworten.



■ ■ ■ *Abbildung 2.2:*
Mögliche Werbung
von BookWorker

Die Handlung

Bevor wir die Grafik des Flash entwickeln können, benötigen wir eine Handlung. In unserer ECard treffen sich zwei Fische im Meer. Sie kommen langsam nacheinander in das Bild, begleitet von Hintergrundmusik. Nach einer kurzen Pause beginnt der erste Fisch mit dem Text „Sie haben eine ECard von ... bekommen“. Der Text erscheint in einer kleinen Sprechblase, der Inhalt kommt aus einer Datenbank.

Abbildung 2.3: ■■■
Begrüßung des
Empfängers



Nun entfernt sich die Sprechblase und bei dem anderen Fisch öffnet sich eine neue Blase mit den Grüßen des Absenders. Die ECard pausiert hier so lange, bis der Empfänger auf einen Weiterknopf klickt.

Abbildung 2.4: ■■■
Die individuellen Grüße



Die Blase entfernt sich und die Fische rücken ein Stück näher. Der große Fisch erzählt nun einen Witz:

„Wie lauten die letzten Worte von Steven Spielberg bei den Dreharbeiten vom ‚Weißen Hai‘?“

Es folgt dann eine kurze Pause ohne Regung der Fische. Die Sprechblase entfernt sich und eine neue öffnet sich:

„Tolles Modell! Sieht aus wie echt...“



■ ■ ■ *Abbildung 2.5:
Der Witz der ECard*

Wieder eine kurze Pause. Dann lachen die Fische laut und schießen aus dem Bild, während wieder die Musik vom Anfang abgespielt wird. Es erscheint noch der Spruch „Lach’ mal wieder!“, während das Bild verdunkelt wird. Abschließend erfolgt die Werbung des Betreibers des ECard-Service, hier BookWorker.

Prozent-Preloader



■ ■ ■ *Abbildung 2.6:
Preloader*

Da wir zur Untermalung der schönen Grafik auch noch Sound einsetzen, entwickeln wir, zur Überbrückung der Ladezeiten, einen Prozent-Preloader, der dem Besucher den exakten Status der Ladezeit anzeigt. Dazu wird die Zahl der geladenen Bytes durch die Gesamtzahl der Bytes dividiert und dann mit 100 multipliziert. Dieser Vorgang lässt sich in Flash sehr leicht realisieren. Öffnen Sie das Bildaktionsfenster beim ersten Schlüsselbild und fügen Sie folgendes ActionScript ein:

Listing 2.1: ■■■
Kleiner Preloader

```
if (_framesloaded>=_totalframes && inhalt != "") {
    gotoAndPlay (3);
}
geladen = this.getBytesLoaded();
insgesamt = this.getBytesTotal();
prozent = Math.round((geladen/insgesamt)*100);
loaded = prozent+" % geladen";
```

In einer ersten If-Abfrage wird überprüft, ob die Anzahl der geladenen Frames größer oder gleich der Gesamtzahl aller Frames ist und ob der Wert der Variablen `inhalt` gefüllt, also nicht leer ist. Ist dies der Fall, springt Flash an Position 3, wo unsere Animation beginnt.

Da die meisten Besucher etwas länger laden müssen, wird eine Anzeige der bereits geladenen Prozente angezeigt. Dazu benötigen wir drei Variablen: eine für die Gesamtzahl der Bytes des Flash-Films, eine zur Ermittlung der bereits geladenen Bytes und eine, in der alles verrechnet wird.

In der Variable `geladen` werden die geladenen Bytes mit der Funktion `getBytesLoaded()` ermittelt und gespeichert. Die Variable `insgesamt` ermittelt die Summe aller Bytes in dem Flash-Film.

In der Variable `prozent` verrechnet Flash die Variablen und teilt `geladen` durch `insgesamt`. Die Summe wird mit 100 multipliziert.

`loaded`, die vierte Variable, enthält den Inhalt von `prozent` und fügt außerdem den Schriftzug „ % geladen“ an. In Flash zeichnen wir dann ein dynamisches Textfeld in der Variable `loaded`.

Fügen Sie nun noch ein Textfeld über der Prozentanzeige hinzu (wie in Abbildung 2.6) mit einem Preload-Hinweis.

Die grafische Gestaltung der ECard



■ ■ ■ *Abbildung 2.7: Der Hintergrund und die zwei Fische*

Beginnen wir mit der Erstellung eines Hintergrundbildes, in dem dann die beiden Akteure – also die Fische – auftreten werden. Wie in Abbildung 2.7 benötigen wir erst einen Hintergrund, der sich aus ein paar Pflanzen und Gesteinen zusammensetzt. Durch den Einsatz verschiedener Farben und Größen erzeugen wir eine Tiefenwirkung, so dass das Bild dreidimensional wirkt. Im Hintergrund geschieht nichts weiter. Der Fokus soll klar bei den Fischen und später dann bei den Blasen bleiben.

Während die Blasen mit Inhalt erscheinen, halten die Fische still. Alle Texte, mit Ausnahme des Hauptinhalts, werden in Blasen wiedergegeben. Zudem benötigen wir für den Inhalt einen Scrollbalken.

► Zeichnen des Hintergrunds

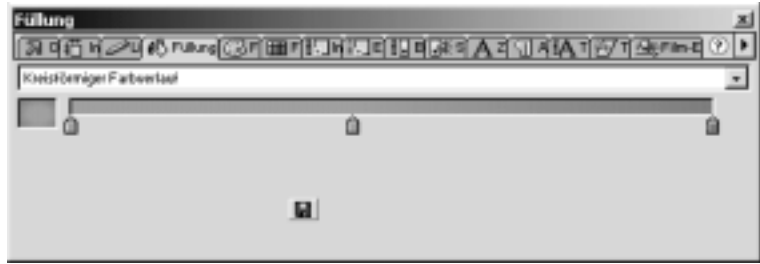
Legen Sie einen neuen Film mit 260 Pixeln in der Breite und 200 Pixeln in der Höhe an. Stellen Sie die Bildrate auf 20 Bps. Zeichnen Sie ein Kästchen über die gesamte Bühne mit der Farbe #1879D6 und sperren Sie die Ebene.

Zeichnen Sie nun einen ovalen Kreis wie in Abbildung 2.8, gefüllt mit einem kreisförmigen Farbverlauf wie in Abbildung 2.9. Er stellt den Schimmer des Ozeans dar.



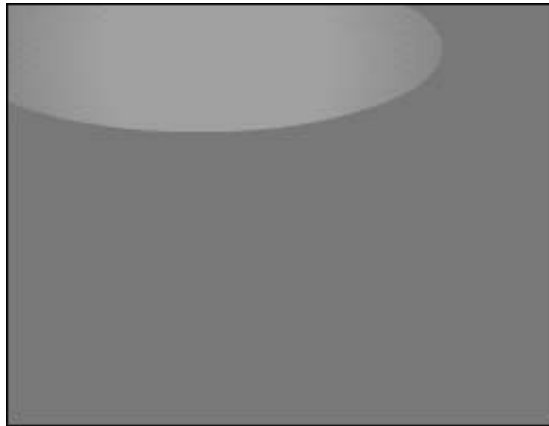
■ ■ ■ *Abbildung 2.8: Dieser Kreis wird der Meeresspiegel.*

Abbildung 2.9: ■■■
Farbverlauf für den
Kreis



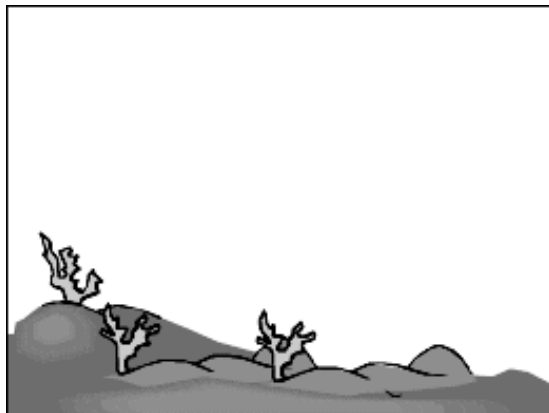
Schneiden und positionieren Sie den Kreis wie in Abbildung 2.10.

Abbildung 2.10: ■■■
Positionierter
Meeresspiegel



Nun zeichnen wir den Boden mit drei Pflanzen darauf. Der Boden hat kleine Erhebungen und durch das Einsetzen von kleinen Feldern in einer helleren Farbe entsteht ein räumlicher Effekt.

Abbildung 2.11: ■■■
Der Boden



In Abbildung 2.11 sehen Sie den fertig gestellten Boden. Damit uns die Farbeinwirkungen nicht irritieren können, zeichnen wir zuerst eine Oberfläche mit schwarzen Linien. Außerdem müssen ein paar Hügelchen ge malt werden.

Nehmen Sie das Linienwerkzeug und zeichnen Sie eine Linie, die die Oberfläche darstellen soll. Wenn Sie eine gerade Linie gezeichnet haben, können Sie sie biegen. Klicken Sie dazu auf die Linie, wobei Sie den Mauszeiger gedrückt halten. Anschließend können Sie die Linie frei verbiegen.



■ ■ ■ *Abbildung 2.12:
Biegen einer Geraden*

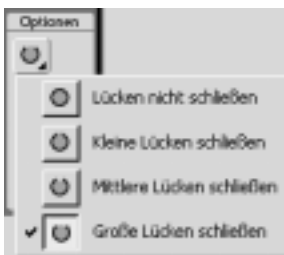
Nun füllen Sie den Bereich unten in Grün mit der Hex-Farbe #528552.



■ ■ ■ *Abbildung 2.13: Hügeluntergrund*

Sie können einen Bereich nur füllen, wenn er „geschlossen“ ist. Das heißt, es sind keine Lücken mehr in dem Linienkonstrukt. Zeichnen Sie deswegen auch unten herum drei Linien an den Kanten, bevor Sie das Bild füllen.

Unter Umständen klappt es noch nicht mit dem Füllen. Wählen Sie dann die Einstellung GROSSE LÜCKEN SCHLIESSEN beim Füllen (Abbildung 2.14).



■ ■ ■ *Abbildung 2.14: Fülloptionen*

Wenn das Füllen nun noch immer nicht funktioniert, überprüfen Sie genauestens Ihre Zeichnung, ob nicht vielleicht noch eine kleine Lücke darin ist. Benutzen Sie die Lupe zur Überprüfung.

Abbildung 2.15: ■■■
Eine kleine Lücke
kann das Ausfüllen
verhindern.



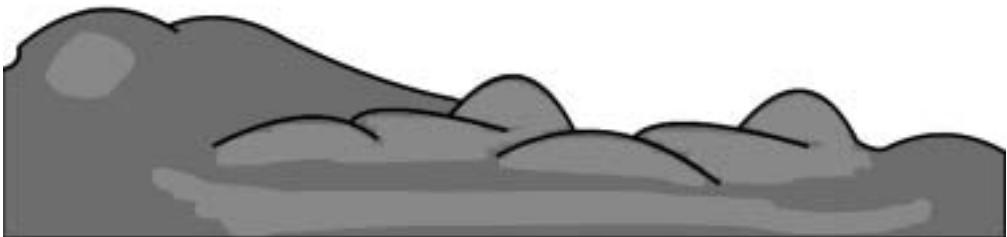
Nachdem die Füllung nun vollständig ist, zeichnen wir kleine hellere Flecke hinein, damit der Boden nicht so flach aussieht, wobei die hinteren etwas dunkler sind. Die mittlere Reihe der Hügel wird ebenfalls hellgrün, ganz vorne kommt der größte und hellste Fleck hin.

Zeichnen Sie auf den Hügel oben rechts ein ähnliches Linienkonstrukt wie in Abbildung 2.16, welches wir durch das Verbiegen der Linien in einen Fleck verwandeln.



■■■ Abbildung 2.16: Flecken

Nachdem der Fleck mit der Farbe #639663 gefüllt ist, löschen Sie die umliegenden Linien einfach mit einem Doppelklick auf einen Teil der Linie und durch Drücken der `Entf`-Taste.




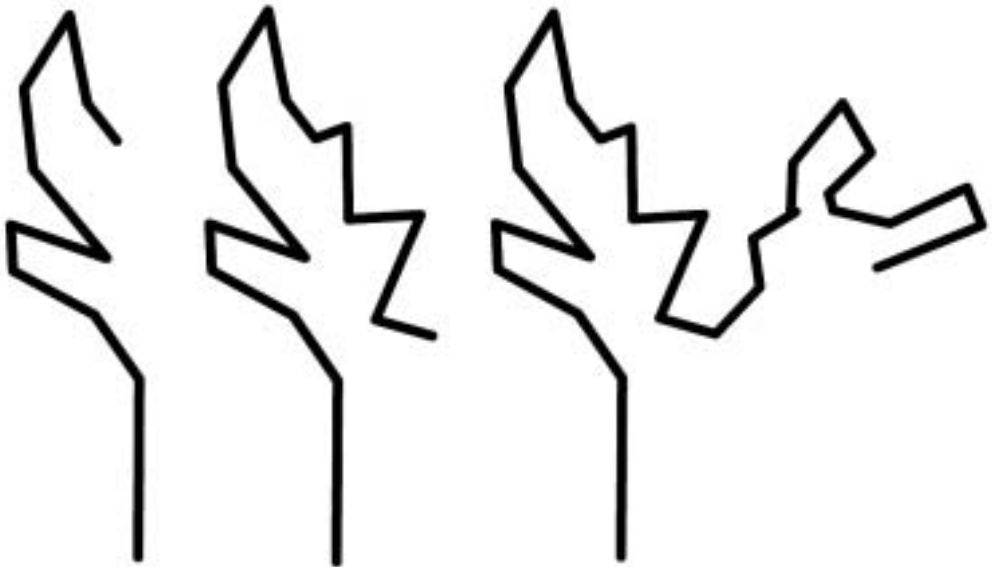
■■■ Abbildung 2.17: Mögliche Füllung

Nachdem der Boden so weit fertig gestellt ist, setzen wir nun drei Pflanzen ein.



■ ■ ■ *Abbildung 2.18:*
Eine Pflanze

Zeichnen Sie mit dem Linienwerkzeug eine Pflanze wie in Abbildung 2.18. Sie kann ruhig eckig bleiben. Bei Bedarf drücken Sie während des Zeichnens die -Taste und die Linie wird gerade gezeichnet. Füllen Sie die Pflanze mit der Hex-Farbe #9CCF9C. Diese Pflanze können wir auch ein zweites Mal in die ECard integrieren.



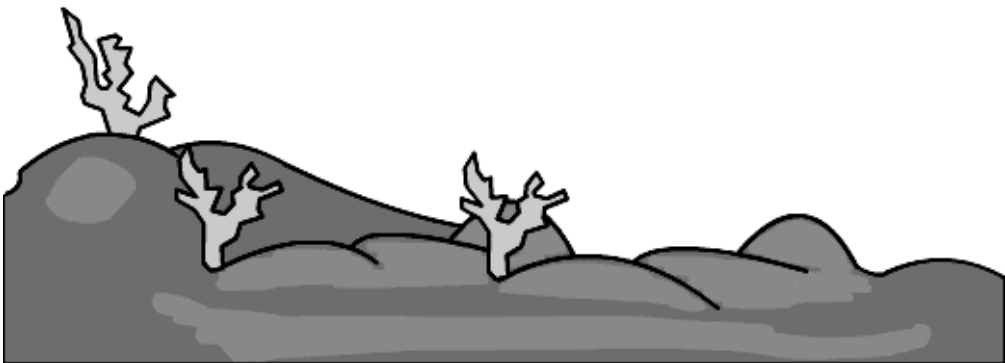
■ ■ ■ *Abbildung 2.19: Die Pflanze*

Außerdem zeichnen wir noch eine andere Pflanze (siehe Abbildung 2.20).

Abbildung 2.20: ■■■
Zweite Pflanze



Gruppieren Sie nun alle vorhandenen Teile der Zeichnung: einmal den Boden, dann die Pflanzen. Positionieren Sie die Elemente wie in Abbildung 2.21.



■■■ *Abbildung 2.21: Die Pflanzen sind hier bereits auf den Boden gesetzt.*

Der Boden ist damit fertig gestellt. Nun bauen wir noch Hintergrundgewölbe, die dunkel und zum Teil durch das Wasser nur verschwommen sichtbar sind.

Beginnen wir bei den zwei großen Pflanzen an den Außenseiten. An der Kante oben links sollen nur noch ein paar Blätter einer Pflanze zu sehen sein. Abbildung 2.22 zeigt eine mögliche Version.



■■■ *Abbildung 2.22:*
Eine weitere Pflanze

Nachdem Sie die Zeichnung fertig gestellt haben, folgt jetzt eine Pflanze für die rechte Seite, die Sie in der nachfolgenden Abbildung sehen.



■■■ *Abbildung 2.23:*
Rechts ist noch eine Pflanze zu sehen.

Fügen Sie nun alle Elemente zusammen. Beachten Sie bitte, dass Sie für jeden Gegenstand eine eigene Ebene anlegen.



■■■ *Abbildung 2.24:*
Der aktuelle Stand

Beachten Sie beim Zusammenfügen, dass sich der Boden in der obersten Ebene befindet. Dahinter folgt die Ebene mit den Pflanzen links und rechts außen. Ganz unten in der Reihenfolge der Ebenen liegt der Ozeanschimmer.

Um den Hintergrund noch realistischer wirken zu lassen, zeichnen wir eine weitere Pflanze und zwei Gesteine ein: eines in den Hintergrund, eines unten links in die Ecke. Die Pflanze kommt links hinter den Boden, noch teilweise sichtbar.

Abbildung 2.25: ■■■
Eine weitere Pflanze



Abbildung 2.26: ■■■
Eine Lagune gibt es
auch.



Abbildung 2.27: ■■■
Steine



Fügen Sie nun jedes Objekt in die Bühne ein und positionieren Sie die Ebenen.



■■■ *Abbildung 2.28: So weit sind wir jetzt.*

Das so weit entwickelte Bild weist durchaus schon eine gelungene Unterwasseratmosphäre auf. Zur Perfektionierung werden jetzt noch zwei Pflanzen in den unmittelbaren Vordergrund gestellt.



■■■ *Abbildung 2.29: Eine weitere Pflanze*



■■■ *Abbildung 2.30: Meine persönliche Lieblingspflanze*

Das Zeichnen dieser beiden Bilder dürfte sich als schwierig herausstellen. Zeichnen Sie zuerst immer das Außengerüst, danach die Schatteneffekte und abschließend erst die detailreicheren Sachen, wie die roten Stacheln in Abbildung 2.30.

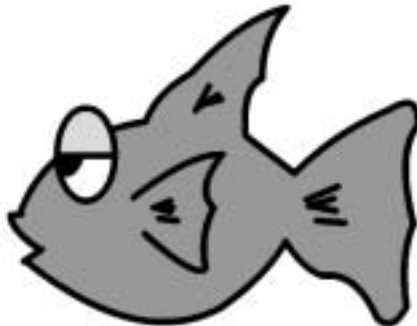
Sie müssen bei den Zeichnungen nicht zu sehr ins Detail gehen. Die Entwicklung dieser beiden Grafiken gehört sicher zu den schwierigsten Aktionen bei der Erstellung unserer ECard.

Abbildung 2.31: ■■■
Der Hintergrund ist
damit fertig.

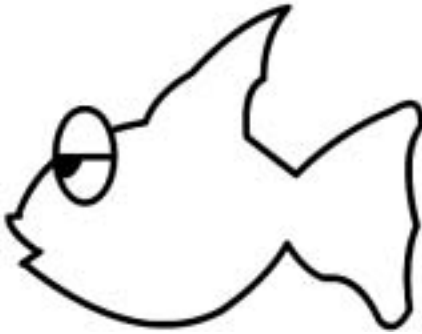


Nachdem nun der Hintergrund komplett ist, kann mit der Entwicklung der Fische begonnen werden. Die ECard zeigt zwei Fische. Jedoch entwickeln wir nur einen, der zweite wird später einfach kopiert und bekommt als Effekt eine andere Farbe.

Abbildung 2.32: ■■■
Ein Fisch



In Abbildung 2.33 sehen Sie den Fisch, den wir jetzt zeichnen. Beginnen Sie mit dem Körper und nehmen sie danach das Auge und die Schuppen in Angriff.



■ ■ ■ **Abbildung 2.33:**
Ein nackter Fisch

Nachdem Sie den Körper des Fisches gemalt haben, füllen Sie ihn mit dem Füllwerkzeug orange aus. Zeichnen Sie dann die Flosse im Zentrum des Körpers. Mit dem Linienwerkzeug folgen die Schuppen.



■ ■ ■ **Abbildung 2.34:**
Der Fisch ist fast fertig.

Der Fisch wird nun noch eine Schwimmbewegung und eine Lachbewegung bekommen. Formatieren Sie den Fisch zu einer Filmsequenz und bearbeiten Sie diese.

Beim Schwimmen bewegen sich nur die drei Flossen ein wenig. Dazu werden wir nun damit beginnen, den Fisch zu „zerlegen“. Klicken Sie zuerst auf die schwarze Linie der inneren Flosse. Schneiden Sie sie aus, indem Sie **[Strg]+[X]** drücken. Legen Sie eine neue Ebene an und fügen Sie sie an der entsprechenden Position mit **[Strg]+[⇧]+[V]** wieder ein. Formatieren Sie die Linie zu einer Filmsequenz.

Nun kopieren Sie die gesamte hintere Schwanzflosse wie in Abbildung 2.35 gezeigt.



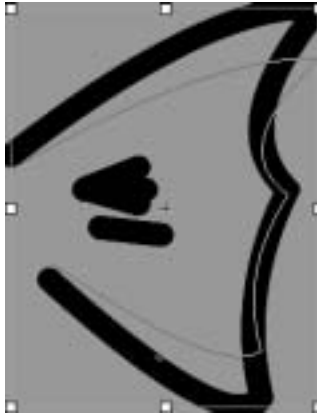
■ ■ ■ **Abbildung 2.35:**
Die Schwanzflosse des Fisches

Fügen Sie die Schwanzflosse ebenfalls an der Position in eine neue Ebene ein und formatieren Sie sie nun auch zu einer Filmsequenz.

Verfahren Sie genauso mit der obersten Flosse.

Beginnen wir nun mit der Animation der mittleren Flosse. Sie soll sich etwas zusammenziehen und dann wieder öffnen. Legen Sie dazu an den Positionen 10 und 20 in der Ebene weitere Schlüsselbilder an. In Bild 10 ist die Flosse zusammengezogen, in Bild 20 wieder geöffnet. Wählen Sie also Schlüsselbild 10 an und skalieren Sie es mit dem Skalierwerkzeug kleiner.

Abbildung 2.36: ■■■
Skalierung der
Fischflosse



Wählen Sie nun die Schwanzflosse an. Auch sie soll sich an Position 10 zusammenziehen und an Position 20 wieder öffnen. Skalieren Sie die Flosse aber nur ein kleines Stückchen, damit sie noch immer zum Rest des Fisches passt.

Die oberste Flosse wird nun etwas breiter und flacher skaliert. An Position 20 befinden sich genau dieselben Schlüsselbilder wie an Position 1.

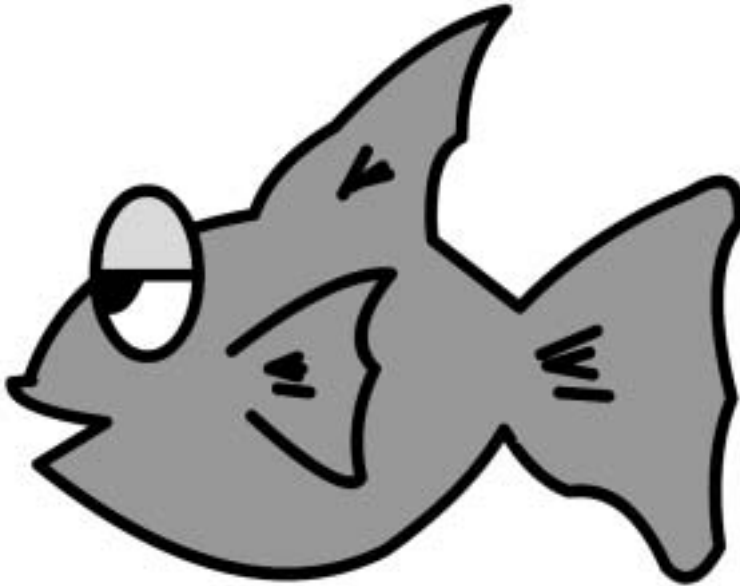
Fügen Sie bei Schlüsselbild 20 folgendes ActionScript ein:

```
gotoandplay (1);
```

So bleibt die Schwimmanimation immer in einer Schleife und die Fische bewegen sich die ganze Zeit.

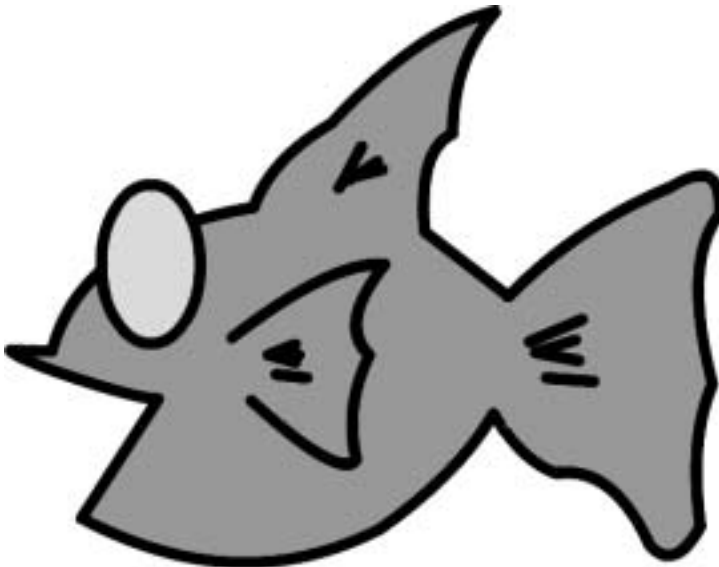
An Position 21 erfolgt noch eine kleine Lachanimation. Sie besteht aus zwei Schlüsselbildern, die sich wiederholen. Im ersten öffnet sich der Fischmund ein kleines Stückchen, im zweiten ist er weit aufgerissen.

Legen Sie an Position 21 in jeder Ebene ein Schlüsselbild an. Ziehen Sie den Mundwinkel des Fisches ein Stück nach hinten und nach oben:



■ ■ ■ *Abbildung 2.37: Der Fisch lacht – Teil 1*

Im nächsten Bild ist das Maul des Fisches weit aufgerissen. Dazu ziehen Sie den Mundwinkel noch etwas nach hinten und nach oben und die Nase sowie das Kinn höher.



■ ■ ■ *Abbildung 2.38: Der Fisch lacht – Teil 2*

Wiederholen Sie diese beiden Bilder über einen Zeitraum von 45 Bildern immer wieder. Im letzten Schlüsselbild setzen Sie folgendes Action-Script ein:

```
gotoandplay (1);
```

Dem Schlüsselbild an Position 21 weisen Sie noch als Bezeichnung „laughing“ zu.

Die Fische werden sich in kleinen Sprechblasen unterhalten. Diese Sprechblasen entwickeln wir nun.

Abbildung 2.39: ■■■
Die Sprechblase




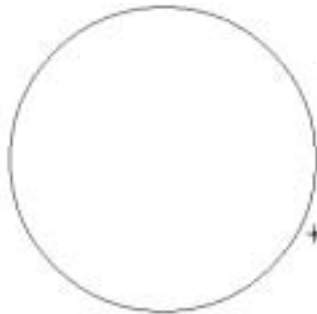
Die Sprechblase basiert auf einem kleinen Trick. Zuerst zeichnen wir einen Kreis mit gedrückter -Taste.

Abbildung 2.40: ■■■
Ein Kreis

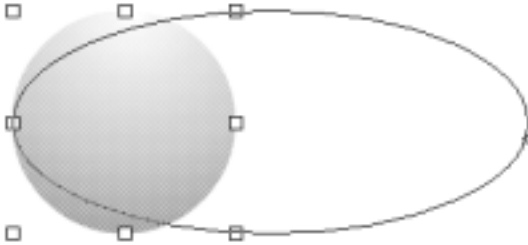


Füllen Sie den Kreis wie folgt:

Abbildung 2.41: ■■■
Füllung der
Sprechblase



Markieren Sie den ausgefüllten Kreis und wählen Sie erneut das Füllwerkzeug aus. Die Blase wird nun so geändert, dass der mittlere weiße Punkt nun an der oberen Kante ist. Klicken Sie dazu einfach oben hin. Strecken Sie den Kreis nun mit der Skalierfunktion wie in Abbildung 2.39.



■ ■ ■ *Abbildung 2.42: Strecken des Kreises*

Nachdem Sie den Kreis in eine ovale Form gebracht haben, kopieren Sie ihn durch Drücken von `[Strg]+[C]` und fügen ihn an derselben Position wieder mit `[Strg]+[⇩]+[V]` ein. Sie haben nun eine Kopie des Kreises an genau derselben Stelle. Formatieren Sie diese Kopie zu einer Filmsequenz und skalieren Sie sie ein Stück kleiner.

Drücken Sie danach `[Strg]+[B]`. Damit teilen Sie die Filmsequenz und der Kreis ist in keiner Filmsequenz mehr, sondern direkt in der Hauptebene. Jetzt färben Sie den inneren Kreis weiß.



■ ■ ■ *Abbildung 2.43: Fertige Blase*

Formatieren Sie die fertige Blase zu einer Filmsequenz und erstellen Sie eine Kopie. Diese skalieren Sie nun wieder richtig rund, wie zu Beginn, aber erheblich kleiner. Schon sind die kleinen Blasen fertig.

Das Innere der Blasen ist durchsichtig. In diesem Bereich erscheint später der Text. Damit dieser nicht direkt auf dem Meereshintergrund liegt und sich ein bisschen abhebt, setzen Sie den Alpha-Wert auf 12%. Wählen Sie dazu den inneren Kreis und den Farbmischer an. Stellen Sie bei Alpha nun 12% ein.

Die Grafik ist damit fertig.

► Die Animation

Nun animieren wir alle bereits gezeichneten Elemente und fügen die Sounds sowie alle Variablen, die aus der Datenbank kommen, ein. Nach diesem Abschnitt erfolgt die Variablenübergabe über Parameter an Flash.

Zu Beginn fügen wir einen zweiten Fisch ein. Kopieren Sie den Fisch und legen Sie eine neue Ebene an, in die Sie die Kopie mit `[Strg]+[V]` einfügen. Wählen Sie im Menü **MODIFIZIEREN > TRANSFORMIEREN** und dann **HORIZONTAL SPIEGELN**. Skalieren Sie den Fisch auf etwa 50% der alten Größe. Wählen Sie bei den Effekten **FARBTON** und **ROT** mit einer Intensität von 40%. Der kleine Fisch guckt nun nach rechts, der große nach links.

Lassen Sie die beiden Fische in die Hauptbühne schwimmen. Der kleine kommt von links außen, der große von rechts außen, wobei der kleine gut zehn Schlüsselbilder später auf die Hauptbühne kommt. Beide treffen sich bei Schlüsselbild 160. Beide brauchen also für ihren Weg rund acht Sekunden (bei 20 Bps). Ordnen Sie die Fische wie in der folgenden Abbildung an.

Abbildung 2.44: ■■■
Die Fische in Position



Falls die zwei Pflanzen nicht im Vordergrund sind, schieben Sie einfach die Ebenen nach oben. Sie sollten ganz im Vordergrund stehen, wenn Sie sie verwenden.

Es öffnet sich nun die erste Blase mit Inhalt. Sie dient der Begrüßung: „Du hast eine ECard von " + Absender + " bekommen.“

Der Wert von „Absender“ wird aus der Datenbank geladen. Der Einbau der Variablen sowie das Laden aus der Datenbank erfolgt später.

Lassen Sie die Blasen nacheinander erscheinen, erst eine kleine am Mund des Fisches, dann eine kleine etwas weiter oben und weiter rechts sowie anschließend die große Blase (Abbildung 2.45).



■■■ *Abbildung 2.45: Die Blase öffnet sich*

Zeichnen Sie ein dynamisches Textfeld in die Sprechblase. Wählen Sie als Schriftart Arial und als Farbe Weiß. Das Textfeld darf nicht anwählbar sein, keinen Rand haben und nicht HTML unterstützen. Betten Sie alle Schriftarten ein und weisen Sie als Variable `send_by` zu. Formatieren Sie das Textfeld zu einer Filmsequenz, die Sie nun bearbeiten. Die Schrift soll mit Schatten wiedergegeben werden. Also markieren Sie das Textfeld und kopieren Sie es. Ändern Sie dann die Farbe in Schwarz. Fügen Sie die Kopie mit `Strg` + `⇧` + `V` wieder ein und verschieben Sie das neue Textfeld mit den Pfeiltasten um einen Schritt nach oben und einen nach links. Sie sehen – der weiße Schriftzug hat nun einen Schatten.

Lassen Sie den Schriftzug eine kurze Weile stehen und dann lassen Sie alles mit einem Motiontween von `Alpha = 100` auf `Alpha = 0` fahren.

Die nächste Luftblase bei dem großen orangen Fisch wird geöffnet. Diese Luftblase muss größer sein, um die individuellen GrüÙe aufzunehmen. Erstellen Sie ein großes dynamisches Textfeld mit den Eigenschaften wie bei der vorhergehenden Luftblase. Erstellen Sie auch den Schatten. Formatieren Sie das Textfeld als Filmsequenz und legen Sie es in eine eigene Ebene.

Positionieren Sie rechts außen zwei Scrollknöpfe und bauen Sie daraus einen Scroller. Damit das Scrollen funktioniert, müssen Sie dem zweiten Textfeld eine neue Bezeichnung wie zum Beispiel „text2“ geben. Wenn Sie jetzt als Textfeld „text“ und „text2“ haben, müssen Sie auch den Scrollknopf so einstellen, dass er beides scrollt. Das kann so aussehen:

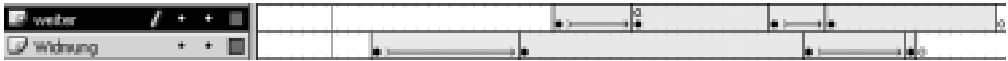
Listing 2.2: ■■■ `on (press, dragOver, dragOut, keyPress "<up>") {`
Eine kleine `_parent.text.scroll = _parent.text.scroll-1;`
Scrollfunktion `_parent.text2.scroll = _parent.text2.scroll-1;`
`}`

Legen Sie ein normales Textfeld an, mit der Beschriftung „Weiter >“. Der Ablauf der ECard wird unterbrochen, so dass dem Benutzer genug Zeit bleibt, um den Text in Ruhe zu lesen. Anschließend klickt er auf WEITER und die Animation wird weitergespielt.

Legen Sie für den WEITER-Schriftzug einen Schatten an und formatieren Sie beide Textfelder zu einem Knopf. Öffnen Sie das Objektaktionsfenster des Knopfs und geben Sie folgendes ActionScript ein:

Listing 2.3: ■■■ `on (press) {`
gotoandplay(); `gotoAndPlay (285);`
`}`

Dabei ersetzen Sie die Zahl (hier 285) durch die Nummer des Schlüsselbilds, das nach der Blase folgt, wo also eine Animation zum Ausblenden der Blase gebaut werden kann.



■■■ **Abbildung 2.46: Die Zeitleiste**

In Abbildung 2.46 sehen Sie den Aufbau der ECard. Wenn die Blase geöffnet ist, stoppt die Handlung. Nach einem Klick auf WEITER wird die Blase ausgeblendet und es geht ganz normal weiter.

Die Fische pausieren kurz, um sich anschließend einander zu nähern, so dass der orange Fisch jetzt den Witz erzählen kann. Es öffnet sich eine Luftblase mit dem Text:

„Wie lauteten die letzten Worte von Steven Spielberg bei den Dreharbeiten vom ‚Weißen Hai?‘“

Es folgt eine kurze Pause und die Fische schauen sich regungslos an. Dann öffnet sich beim orangen Fisch eine neue kleine Luftblase:

“Tolles Model! Sieht aus wie echt ...“

Wieder sind die Fische regungslos. Dann lachen sie laut und schwimmen weg.

Um die Fische zum Lachen zu bringen, weisen Sie ihnen Instanzen zu – dem roten Fisch „fish1“ und dem orangen Fisch „fish2“. Fügen Sie danach in das erste Schlüsselbild, von wo auch die Fische wegschwimmen, folgendes ActionScript ein:

```
fish1.gotoAndPlay("laughing");
```

Fünf Schlüsselbilder später folgt

```
fish2.gotoAndPlay("laughing");
```

Damit wird in den Filmsequenzen zur Lachanimation gesprungen. Weisen Sie dem Schlüsselbild, wo der erste Fisch verschwindet, den Sound „laugh“ (finden Sie auf der CD) zu. Nach gut 45 Bildern ohne Aktion lassen Sie die Fische schnell aus dem Wasser schießen und dazu jeweils den Song „Cartoon48“ abspielen. Im Hintergrund setzen Sie noch das „Rock1-Interned-5“ ein.

Nun wird der Schriftzug „Lach mal wieder“ eingeblendet. Danach wird der sichtbare Bereich verdunkelt und die Werbung angezeigt. Die grafische Gestaltung ist damit abgeschlossen.

■■■ **Listing 2.4:**
Den ersten Fisch zum Lachen bringen

■■■ **Listing 2.5:**
Dann darf auch der zweite lachen

2.2 Die PHP- und Datenbankprogrammierung

Jürgen Lange

Aufbau der Datenbank



■ ■ ■ **Abbildung 2.47:**
Das MySQL-Logo

Da viele User zur gleichen Zeit ECards verschicken und die Empfänger die Karten vielleicht öfter als ein Mal und zu jeder erdenklichen Zeit abholen möchten, benötigen wir für unser ECard-System eine Datenbank.

► Anlegen der Datenbank

Bevor es mit der Programmierung losgehen kann, müssen wir mit Hilfe des PHPmyAdmin die Datenbank für das ECard-System anlegen. Dazu bedarf es zunächst der Überlegung, welche Daten in der Datenbank gespeichert werden müssen.

Zuallererst sollte immer eine Spalte angelegt werden, die die Anzahl der Datensätze anzeigt, die so genannte ID-Spalte. Diese Spalte ist die erste in jeder Datenbanktabelle und auch verantwortlich für die genaue Identifizierung des Datensatzes, in unserem Fall, welche ECard zu welchem Empfänger gehört.

Als Nächstes richten wir Spalten für den Absender, die Absender-E-Mail-Adresse, den Empfänger und die Empfänger-E-Mail-Adresse ein. Es sind momentan also fünf Spalten. Nun gibt der Absender ja noch einen speziellen Text für den Empfänger ein, wofür wiederum eine Spalte benötigt wird. Auch das Datum und das Passwort müssen in der Datenbank gespeichert werden. Wir brauchen also genau acht Spalten in unserer Tabelle.

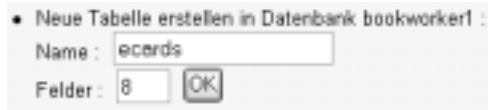
Zum Anlegen der Tabellen haben wir nun zwei Möglichkeiten für Sie vorbereitet. Zum einen wäre es möglich, den folgenden SQL-Code für das Erstellen der Tabelle einfach in das dafür vorgesehene Feld im PHPmyAdmin zu kopieren und auf OK zu klicken. Der Code dafür lautet:

Listing 2.6: ■ ■ ■
Tabelle erzeugen

```
CREATE TABLE ecards (
  id int(255) NOT NULL auto_increment,
  passwort tinytext NOT NULL,
  absender text NOT NULL,
  empfaenger text NOT NULL,
  tomail text NOT NULL,
  frommail text NOT NULL,
  datum text NOT NULL,
  inhalt text NOT NULL,
  PRIMARY KEY (id, id),
  UNIQUE id (id)
);
```

Die andere Möglichkeit wäre, die Tabelle Schritt für Schritt im PHPmy-Admin zu erstellen. Gehen Sie dazu wie folgt vor:

Öffnen Sie Ihren PHPmyAdmin in Ihrem Browser und wählen Sie die gewünschte Datenbank aus.



■ ■ ■ **Abbildung 2.48: Tabelle in Datenbank bookworker1 erstellen**



■ ■ ■ **Abbildung 2.49: Tabelleneigenschaften**

Tragen Sie im Feld „Name“ im Abschnitt „Neue Tabelle erstellen in Datenbank...“ das Wort „ecards“ ein. Geben Sie als Anzahl der Spalten bei „Felder“ die schon besprochene 8 ein und drücken Sie OK.

Geben Sie die Eigenschaften wie in der Abbildung in den dafür vorgesehenen Feldern ein. Achten Sie darauf, dass Sie als Extra für die Spalte ID „auto_increment“ belegen. Dies bewirkt, dass bei einem neuen Eintrag durch den Benutzer über das Formular automatisch eine neue Zeile und damit ein neuer Datensatz in der Datenbank erzeugt wird. Weiterhin sollten Sie den Umlaut bei Empfänger mit ae umschreiben und außerdem alle Bezeichnungen klein schreiben.

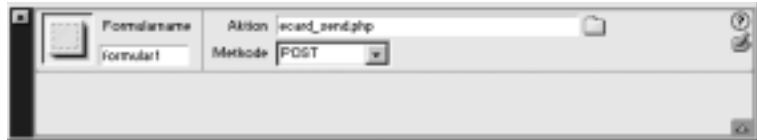
Aktivieren Sie für die Spalte ID die Optionen für PRIMÄRSCHLÜSSEL, INDEX und UNIQUE und klicken Sie auf SPEICHERN.

Sie erhalten eine Bestätigung und haben somit die Tabelle erfolgreich erstellt.

Eingabe der Daten



Wir erstellen dieses Formular nun mit Macromedias Dreamweaver 4, auf dem auch die Anleitung basiert. Sollten Sie den Dreamweaver nicht besitzen, können Sie sich alternativ den HTML-Code in der zugehörigen Datei ECARD.PHP auf der CD zum Buch anschauen.

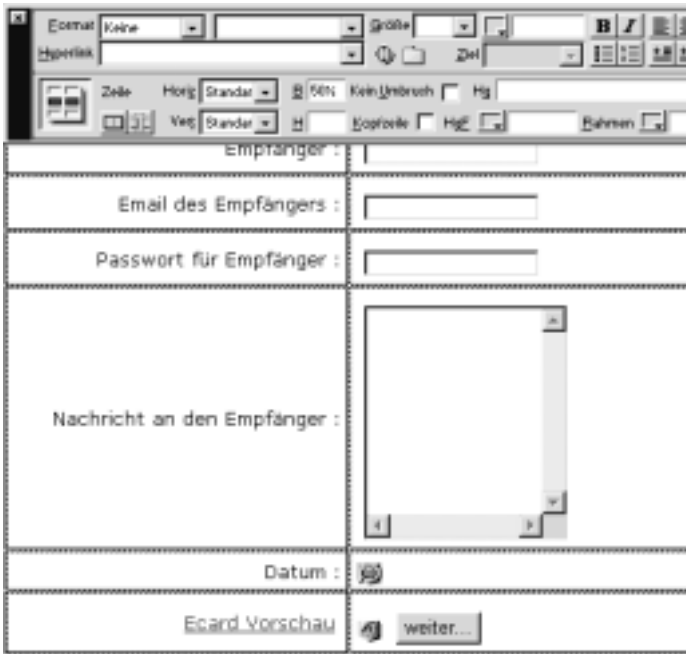


■ ■ ■ *Abbildung 2.50: Formulareigenschaften*

In das erste Formular gibt der Absender die benötigten Daten zur Versendung der ECard ein. Neben dem Namen und der E-Mail-Adresse des Empfängers benötigen wir selbiges für den Absender, das Passwort für den Empfänger, den gewünschten Text und das Datum. Das Datum wird hierbei automatisch durch ein kleines PHP-Skript generiert. Das Formular enthält außer der Datumsgenerierung keinen weiteren PHP-Code.

Erstellen Sie durch einen Klick in das Menü EINFÜGEN > FORMULAR ein neues Formular und benennen Sie es „formular1“. Als Aktion legen Sie den Wert „ecard2.php“ fest.

Im nächsten Schritt fügen Sie eine neue Tabelle mit acht Zeilen und zwei Spalten ein (EINFÜGEN – TABELLE) und modifizieren die Spaltenbreite mit 50%. In der rechten Spalte fügen Sie der Reihe nach von oben nach unten Textfelder mit den Bezeichnungen „absender“, „frommail“, „empfaenger“, „tomail“, „pass“ und „inhalt“ ein (EINFÜGEN > FORMULAROBJEKTE > TEXTFELDER). Aktivieren Sie für das Textfeld „pass“ die Option KENNWORT; dies hat zur Folge, dass der dort eingetippte Text in Form von Sternchen dargestellt, aber trotzdem richtig übergeben wird. Für das Textfeld „inhalt“ aktivieren Sie die Option MEHRZEILIG, um dem Benutzer die Möglichkeit zu geben, einen längeren Text einzutippen. Setzen Sie die Anzahl der Zeichen pro Zeile auf „10“.



■ ■ ■ *Abbildung 2.51: Spaltenbreite festlegen*



■ ■ ■ *Abbildung 2.52: Textfeldeigenschaften*

Sie sind inzwischen in der siebten Zeile (immer noch rechte Spalte) gelangt. Fügen Sie dort nun folgendes PHP-Skript ein, um das aktuelle Datum zu generieren:



■ ■ ■ *Abbildung 2.53: Skripte einfügen*

```
<?php $datum = date("d.m.Y "); echo"$datum"; ?>
```

Dieses Skript erzeugt mit Hilfe der PHP-Funktion `DATE()`; und den Attributen `d` (Tag), `m` (Monat) und `Y` (Jahr) das aktuelle Datum im deutschen Format und gibt es an selbiger Stelle aus.

Abbildung 2.54: ■■■
**Verstecktes Feld mit
 PHP-Skript als Wert**



In die letzte Zeile kommen zwei Elemente. Zum einen benötigen wir ein verstecktes Textfeld, welches das vom PHP-Skript erzeugte Datum an die Auswertung weitergibt. Dazu fügen Sie selbiges über das Menü **EINFÜGEN > FORMULAROBJEKTE > VERSTECKTES FELD** ein und geben als Wert folgenden PHP-Ausdruck an:

```
<?php echo"$datum"; ?>
```

Abbildung 2.55: ■■■
Absende-Button

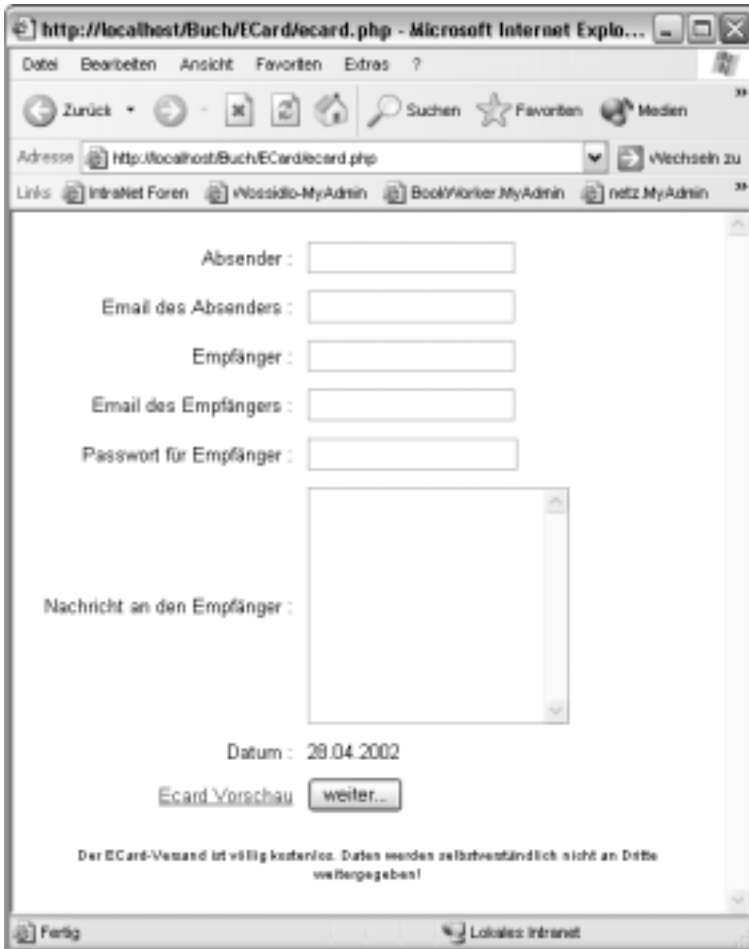


Zum anderen benötigen wir abschließend natürlich den Absende-Button, mit dem der Absender die Eingabe bestätigen kann. Diesen fügen Sie über das Menü **EINFÜGEN > FORMULAROBJEKTE > SCHALTFLÄCHE** in das Formular ein. Beschriften Sie den Button mit „weiter...“ und nennen Sie ihn „submit“.

Zu guter Letzt fügen Sie in die linke Spalte die Bezeichnungen für die Textfelder und Formularobjekte ein (von oben nach unten: Absender, E-Mail des Absenders, Empfänger, E-Mail des Empfängers, Passwort, Nachricht für den Empfänger und Datum). Speichern Sie das Ganze unter dem Namen **ECARD.PHP** ab.

Abbildung 2.56: ■■■
Linke Spalte

Damit ist dem Absender nun die Möglichkeit gegeben, die Daten für die ECard einzugeben.

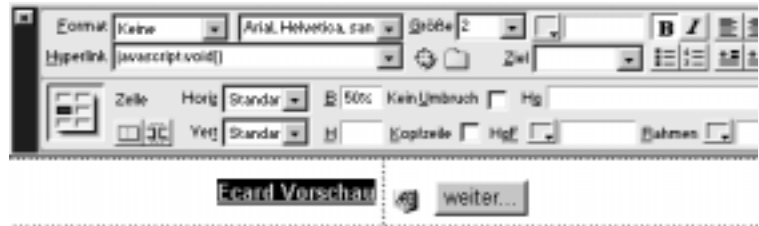


■■■ *Abbildung 2.57: Die fertige ecard.php*

Die Vorschaumöglichkeit

Als Absender möchte man natürlich gerne sehen, wie die Karte aussieht, die man dem Empfänger schickt. Auch unser ECard-System soll dem Absender diese Möglichkeit geben. Und das natürlich noch bevor irgendwelche Daten abgeschickt wurden. Diese Funktion realisieren wir mit Hilfe eines einfachen PopUp-Fensters, das die fertige Karte mit einem allgemeinen Text enthält.

Öffnen Sie die Datei ECARD.PHP in Ihrem Dreamweaver und beschriften Sie die letzte Zeile in der linken Spalte der Tabelle mit „ECard-Vorschau“.



■ ■ ■ **Abbildung 2.58: Hyperlink belegen**

Markieren Sie den Text und fügen Sie folgenden Code als Hyperlink in den Eigenschaftenspektor ein:

```
javaSkript:void()
```



■ ■ ■ **Abbildung 2.59: Das PopUp-Fenster**

Wechseln Sie in die Code-Ansicht und fügen Sie folgenden Code in den Abschnitt nach `javaSkript:void(0)` ein:

```
onClick="window.open( 'vorschauptopup.php?id=1' , 'ECard' ,  
'top=0, left=0, width=280, height=220' )"
```

Der komplette Code sollte nun so aussehen:

```
<a href="javaSkript:void(0)" onClick="window.open('vorschauptopup.php?id=1' , 'ECard' , 'top=0, left=0, width=280, height=220' )" >ECard-Vorschau</a>
```

Damit öffnet sich per Klick auf den Link ein neues Fenster mit den Maßen 280x220 Pixel, genau links oben in der Ecke.

Kommen wir nun zum Inhalt dieses Fensters. Der Absender möchte die Vorschau mit einem allgemeinen Text, beispielsweise mit „Hier könnte Ihre Nachricht an den Empfänger stehen“, betrachten. Da das System auf einer Datenbank basiert, lesen wir einfach nur den ersten Eintrag der Datenbank aus, welcher den allgemeinen Text enthält. Das hat den Vorteil, dass man den Text im Nachhinein leicht ändern kann und er immer sofort und aktuell in der Vorschau erscheint. Um das Ganze so zu realisieren, benötigen wir das erste PHP-Skript, welches den Eintrag aus der Datenbank liest.

Die Skripte zum Auslesen der Datenbank verändern sich im Weiteren nicht sonderlich, sie folgen alle einem bestimmten Muster: Verbindung zur Datenbank aufbauen, Daten auslesen und ausgeben, Verbindung zur Datenbank schließen. Ich werde kurz eines dieser Skripte erläutern, um so den Einstieg zu erleichtern.

```
<?php
require ("connect.inc.php");
```

Hier wird die im Kapitel „PHP und externe Datenquellen“ besprochene Datei `CONNECT.INC.PHP` eingelesen und die Variablen werden auch für diese Datei verwendet.

```
$result = MySQL_query("SELECT * FROM ecards WHERE id=$id")
or die("SQL-Anweisung fehlerhaft !");
```

Die Variable `$result` wird geschrieben und mit der SQL-Anweisung (Datenbankabfrage-Code) belegt. Weiterhin wird überprüft, ob der String einwandfrei ist. Sollte dies nicht der Fall sein, wird eine Fehlermeldung ausgegeben („SQL-Anweisung fehlerhaft!“). Die SQL-Anweisung liest alle Daten aus der Tabelle „Ecards“ (`SELECT * FROM ecards`), welche sich in der Zeile mit der ID-Nummer befinden, die die Variable `$id` enthält. Wenn also die Variable `$id` die Ziffer 1 enthält, liest der String alle Daten aus der ersten Zeile aus. Wenn sie die Zahl 10 enthält, werden alle Daten aus der zehnten Zeile gelesen usw.

```
while($row = MySQL_fetch_array($result)) {

    $id=$row["id"];
    $absender=$row["absender"];
    $empfaenger=$row["empfaenger"];
    $inhalt=$row["inhalt"];
    $tomail=$row["tomail"];
    $frommail=$row["frommail"];
    $datum=$row["datum"];
```

Hier werden in einer Schleife alle zutreffenden Daten mit Variablen belegt, d.h., es wird für die aktuelle Zeile zum Beispiel der Inhalt der Spalte „absender“ ausgelesen und in die Variable `$absender` geschrieben.

```

echo" Heute am $datum hast du, liebe(r) $empfänger, von
$absender ($frommail) eine ECard bekommen. Der Link zu der
Karte wurde auch an deine Emailadresse ($tomail) gesendet.
Die Karte enthält folgende Nachricht: $inhalt" ;
}
echo"<br><br>";
?>

```

Hier werden die Variablen in einem zusammenhängenden Beispieltext ausgegeben. Wenn die Daten also aus der Datenbank ausgelesen werden, wird der entsprechende Text für die Variablen eingesetzt. Der Text könnte beispielsweise lauten:

```

Heute am 02.08.2001 hast du, liebe(r) Azad Adsay, von Jürgen
Lange (lange@netzproduktion.com) eine ECard bekommen. Der Link
zu der Karte wurde auch an deine Emailadresse (azad@work4.de)
gesendet. Die Karte enthält folgende Nachricht: Hallo Azad, bin
heute Abend wieder mal in Berlin, hoffe du schreibst fleißig wei-
ter. Ich melde mich sobald ich wieder da bin. Bis dann, Jürgen !

```

Wenn die SQL-Anweisung anders gelaute hätte (z.B. `SELECT * FROM ecards WHERE id BETWEEN 1 AND 3`), wären möglicherweise mehrere Zeilen betroffen gewesen. Dies würde bedeuten, dass der obige Text aufgrund der beschriebenen Schleife mehrmals, jeweils mit den entsprechenden Inhalten der Zeile, angezeigt worden wäre.

Man kann die Ausgabe der Variablen auch innerhalb des normalen HTML-Codes realisieren. Dazu gibt es zwei Möglichkeiten. Die erste wäre, jeweils in den HTML-Code an die entsprechende Stelle ein kleines PHP-Skript zu schreiben, welches nur die Variable ausgibt. Ein Beispiel dafür wäre:

```

<a href="<?php echo "$url"; ?>"><?php echo "$text"; ?></a>

```

Die zweite Möglichkeit wäre, den HTML-Code direkt in PHP zu schreiben. Ein Beispiel dafür:

```

<?php echo "<a href=\"\$url \">$text</a>"; ?>

```

Der Nachteil hierbei ist, dass man den HTML-Code beispielsweise in Dreamweaver nur noch in der Codeansicht von Hand bearbeiten kann, da der Dreamweaver das Ganze als ein einziges PHP-Skript ansieht – bei komplexeren Skripten ein enormer Zeitaufwand.

Sie haben nun einen kurzen Einblick in das Auslesen aus Datenbanken gewonnen. Wenden Sie nun diese Kenntnisse an, um die Vorschau-möglichkeit für die ECard zu realisieren.

Erstellen Sie eine neue Seite in Ihrem Dreamweaver und wechseln Sie in die Codeansicht.

Fügen Sie folgenden Code vor dem <html>-Tag ein:

```
<?php
require ("connect.inc.php");
    $result = MySQL_query("SELECT * FROM ecards WHERE id=$id")
    or die("Query broke! ECard");
while($row = MySQL_fetch_array($result)) {

    $id=$row["id"];
    $absender=$row["absender"];
    $empfaenger=$row["empfaenger"];
    $inhalt=$row["inhalt"];
    $tomail=$row["tomail"];
    $frommail=$row["frommail"];
    $datum=$row["datum"];

?>
```

■ ■ ■ Listing 2.7:
Auslesen des
entsprechenden
Datensatzes

Um die Flash-Datei einzubetten, gehen Sie nicht wie gewohnt vor. Fügen Sie einfach folgenden Code zwischen dem <body>- und dem </body>-Tag ein:

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase=http://download.macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=5,0,0,0 WIDTH=280 HEIGHT=220>

<PARAM NAME=movie VALUE="e_card.swf?id=<? echo "$id" ?>& emp-
faenger=<? echo "$empfaenger" ;?>&absender=<? echo "$absender"
;?>&inhalt=<? echo "$inhalt" ;?>&tomail=<? echo "$tomail" ;?>&
frommail=<? echo "$frommail" ;?>&datum=<? echo "$datum" ;?>">
<PARAM NAME=quality VALUE=high><EMBED src="e_card.swf?id=<? echo
"$id" ;?>&empfaenger=<? echo "$empfaenger" ;?>&absender=<? echo
"$absender" ;?>&inhalt=<? echo "$inhalt" ;?>&tomail=<? echo
"$tomail" ;?>&frommail=<? echo "$frommail" ;?>&datum=<? echo
"$datum" ;?>%20"e_card.swf?id=<? echo "$id" ;?>&empfaenger=<?
echo "$empfaenger" ;?>&absender=<? echo "$absender"
;?>&inhalt=<? echo "$inhalt" ;?>&tomail=<? echo "$tomail"
;?>&frommail=<? echo "$frommail" ;?>&datum=<? echo "$datum" ;?>
"" quality=high WIDTH=280 HEIGHT=220 TYPE="application/
x-shockwave-Flash" PLUG-INSPAGE="http://www.macromedia.com/
shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">
</EMBED>

</OBJECT>
```

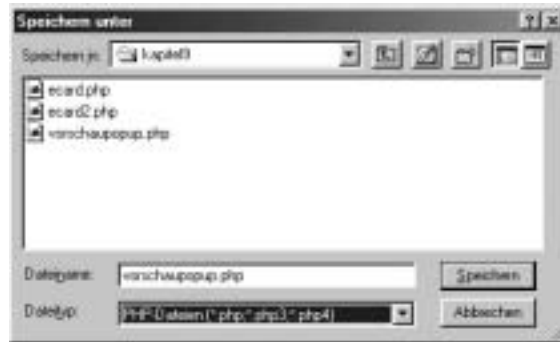
Das sieht erst einmal verwirrend aus, ist aber eigentlich der normale Einbettungscode von Flash-Filmen in HTML. Dieser wurde durch die PHP-Variablen ergänzt, indem die einzelnen Mini-PHP-Skripte an die Adresse angehängt werden ("e_card.swf?id=<? echo "\$id" ?>& empfaenger=<? echo "\$empfaenger" ;?>&absender=<? echo "\$absender" ;?>&inhalt=<? echo "\$inhalt" ;?>&tomail=<? echo "\$tomail" ;?>& frommail=<? echo "\$frommail" ;?>&datum=<? echo "\$datum" ;?>").

Ergänzen Sie hinter dem `</object>` noch folgenden wichtigen Code:

```
<?php  
    }  
?>
```

Dieses Skript schließt die Schleife des Skripts aus dem zweiten Schritt.

Abbildung 2.60: ■■■
Speicherdialog



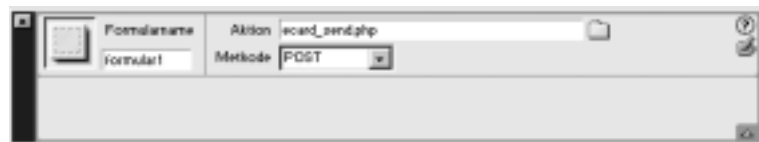
Speichern Sie die Datei unter dem Namen `VORSCHAUPOPUP.PHP` in Ihrem Ordner ab.

Wenn Sie nun von der Datei `CARD.PHP` aus das Vorschaufenster aufrufen, sollte sich das neue Fenster mit der ECard und dem Inhalt des ersten Eintrags aus der Datenbank (id 1) öffnen.

Überprüfung der Daten

Der User hat also seine Daten eingegeben und die Karte betrachtet. Er klickt nun auf den `WEITER...`-Button, um nochmals seine Daten überprüfen zu können. Wir listen ihm also die aus dem Formular übergebenen Variablen geordnet in einer Tabelle wieder auf.

Abbildung 2.61: ■■■
Formulareigenschaften



Erstellen Sie eine neue Seite und fügen Sie ein neues Formular (`EINFÜGEN > FORMULAR`) hinzu. Legen Sie als Aktion `ECARD_SEND.PHP` fest und vergeben Sie als Methode die Option `POST`.

Erstellen Sie eine Tabelle mit zwei Spalten und acht Zeilen.

Absender	<input type="text"/>
Email des Absenders	<input type="text"/>
Empfänger	<input type="text"/>
Email des Empfängers	<input type="text"/>
Passwort für Empfänger	<input type="text"/>
Datum	<input type="text"/>
Nachricht	<input type="text"/>
Eingaben erneuern	<input type="button" value="ECard versenden"/>

■ ■ ■ **Abbildung 2.62: Linke Spalte beschriften**



■ ■ ■ **Abbildung 2.63: Mini-Skripte**

Beschriften Sie die linke Spalte von oben nach unten mit folgenden Texten: Absender, E-Mail des Absenders, Empfänger, E-Mail des Empfängers, Passwort für Empfänger, Datum und Nachricht.

In der letzten Zeile in der linken Spalte geben Sie dem Absender die Möglichkeit, die Eingabe erneut zu tätigen, indem Sie einen Link zur vorherigen Seite einfügen. Schreiben Sie „Eingabe erneuern“ und belegen Sie den Text mit folgendem Hyperlink:

```
javaSkript:history.go(-1)
```

Als Nächstes fügen Sie die folgenden Mini-PHP-Skripte in der rechten Spalte von oben nach unten ein:

- Zeile 1: <?php echo"\$absender"; ?>
- Zeile 2: <?php echo"\$frommail"; ?>
- Zeile 3: <?php echo"\$empfaenger"; ?>
- Zeile 4: <?php echo"\$tomail"; ?>
- Zeile 5: <?php echo"\$pass"; ?>
- Zeile 6: <?php echo"\$datum"; ?>
- Zeile 7: <?php echo"\$inhalt"; ?>



■ ■ ■ **Abbildung 2.64: PHP-Skripte als Werte von versteckten Feldern**

In der letzten Zeile benötigen wir für jede Zeile und jede Variable ein verstecktes Formularfeld (EINFÜGEN > FORMULAROBJEKTE > VERSTECKTES FELD). Sie fügen im folgenden Schritt also sieben versteckte Felder ein und benennen diese der Reihe nach wie folgt:

absender, frommail, empfaenger, tomail, pass, datum, inhalt.

Anschließend fügen Sie wiederum von Anfang bis Ende folgende Werte für die versteckten Felder ein:

```
Feld 1: <?php echo"$absender"; ?>
Feld 2: <?php echo"$frommail"; ?>
Feld 3: <?php echo"$empfaenger"; ?>
Feld 4: <?php echo"$tomail"; ?>
Feld 5: <?php echo"$pass"; ?>
Feld 6: <?php echo"$datum"; ?>
Feld 7: <?php echo"$inhalt"; ?>
```

Abschließend fügen Sie erneut einen Button (EINFÜGEN > FORMULAROBJEKTE > SCHALTFLÄCHE) ein, belegen ihn mit dem Namen „submitz“ und beschriften ihn mit „ECard versenden“. Speichern Sie die Datei unter dem Namen ECARD2.PHP.

Sie haben dem Benutzer nun die Möglichkeit gegeben, seine Daten zu überprüfen und sie im Falle einer Fehleingabe erneut einzugeben. Die Daten werden nun an das dritte und letzte Skript im Absendeprozess übergeben und in die Datenbank eingetragen.

Eintragung in die Datenbank und E-Mail-Generierung

Bei der Erstellung des Eintrage-Skripts bzw. des entsprechenden SQL-Codes wenden wir die im Abschnitt „Datenbank auslesen“ gewonnenen Kenntnisse an. Folgende Teilschritte werden im letzten Skript enthalten sein:

Zuerst wird eine Verbindung zur Datenbank hergestellt, um die SQL-Anweisung zu übergeben (Eintragung der Daten in die Datenbank). Das Ganze wird in eine IF-Schleife gepackt, die kontrolliert, ob die Variablen tatsächlich übergeben wurden – eine gute Möglichkeit, um leere Einträge zu vermeiden.

Im Falle eines korrekten Eintrags sollte ein positives Feedback erfolgen (im gegenteiligen Fall selbiges auch in negativer Ausführung!).

Als Nächstes wird eine Abfrage gestartet, die die ID-Nummer des aktuellen Eintrags wieder ausliest (dient zur genauen Identifizierung des Eintrags beim späteren Abholen der Karte) und in der Variable \$id speichert.

Um alle bis hierhin erklärten Funktionen realisieren zu können, fügen Sie bitte folgendes Skript vor dem <html>-Tag ein:

```

<?php
    if($submit2) {
        include ("connect.inc");
        MySQL_db_query("$dbName", "insert into ecards
            (passwort,absender,empfaenger,tomail,
            frommail,datum,inhalt)
        VALUES ('$passwort','$absender','$empfaenger'.'$tomail',
            '$frommail','$datum','$inhalt')");

        $num = MySQL_affected_rows();

        if ($num>0) {
            $ergebnis="Ihre Ecard wurde versendet !";
        }

        else
            { $ergebnis="ECard wurde nicht versendet !";
            }

        require ("connect.inc.php");
        $result = MySQL_query("SELECT * FROM ecards
            WHERE empfaenger='$empfaenger' AND absender='$absender'"
            or die("Datenbankfehler! DB: ECards");

        while($row = MySQL_fetch_array($result)) {
            $id=$row["id"];
        }
    }

?>

```

■ ■ Listing 2.8:
Eintragungsskript

Nun folgt die Generierung der E-Mail für den Empfänger, in der (neben einem kurzen Erläuterungstext) die URL (Adresse) zur richtigen ECard steht.

Abschließend wird dem Absender ebenfalls genau diese Adresse und auch das Passwort noch einmal ausgegeben, damit dieser auch die Möglichkeit hat, sich die ECard anzusehen.

Für diese Funktionen fügen Sie bitte folgenden Code zwischen dem <body>- und dem </body>-Tag ein:

```

<?php

$subject = "Du hast eine ECard von $absender bekommen !";
$recipient = "$tomail";
$sender = "From: BookWorker ECards<ecards@bookworker.de>";

$mailbody = "Hallo $empfaenger,\n\n";
$mailbody = $mailbody."$absender sendet dir eine BookWorker
    ECard !\n";
$mailbody = $mailbody."Du kannst die Karte unter\n\n";
$mailbody = $mailbody."http://www.bookworker.de/
    ecardview.php?id=$id \n\n";

```

■ ■ Listing 2.9:
Mail-Skript

```

$mailbody = $mailbody."mit dem Passwort : $passwort \n\n";
$mailbody = $mailbody."abholen und von dort aus auch gleich
eine Antwort-Ecard versenden.\n\n";
$mailbody = $mailbody."Wir hoffen dir gefällt unser Service und
du empfiehlst uns weiter.\n\n";
$mailbody = $mailbody."Mit freundlichen Grüßen,\n";
$mailbody = $mailbody."Das BookWorker-Team\n\n";
$mailbody = $mailbody."http://www.bookworker.de";

mail($recipient,$subject,$mailbody,$sender);

echo" Ihre ECard wurde versendet, sie steht für den
Empfänger nun unter
<a href=ecardview.php?id=$id target=_blank>
ecardview.php?id=$id</a>
mit dem Passwort <b>$passwort</b> zum Abholen bereit. <p>
Vielen Dank, dass Sie unseren Service genutzt haben und
weiterhin viel Spaß auf BookWorker !";

?>

```

Mit dem Abspeichern der Datei unter dem Namen ECARD_SEND.PHP ist die Erstellung der Skripte für den Absendeprozess beendet. Es folgt nun das „Abholssystem“ für den Empfänger.

Einfacher Passwortschutz

Die E-Mail ist mittlerweile beim Empfänger angekommen und er möchte sich die ECard ansehen. Die in der E-Mail angegebene Adresse lautet beispielsweise *http://www.bookworker.de/ecardview.php?id=10*. Wie unschwer zu erkennen ist, basiert die Personalisierung der ECard auf der Zahl hinter der Variablen `$id` (`?id=10`). Wären wir bei der Programmierung dieses ECard-Systems unvorsichtig gewesen, könnte nun jeder diese Adresse in die Adresszeile eintippen und die Nummer ändern. Somit könnte er sich jede einzelne ECard mit dem spezifischen Text ansehen, was sicherlich nicht im Sinne der Absender wäre. Aus diesem Grund ist ein einfacher Passwortschutz auf Basis einer IF-Schleife eingebaut worden.

Die IF-Schleife teilt die Seite in zwei Schritte auf. Wird sie zum ersten Mal geladen, kann noch kein Passwort eingegeben sein. Folglich ist die Variable, in der sich das Passwort befinden soll, leer (`if ($passwortheingabe == '')`). Es wird also beim ersten Laden ein Formular geladen, in das das Passwort eingegeben werden soll. Die Schleife fragt nun nach dem Bestätigen der Eingabe ab, ob das eingegebene Passwort mit dem übereinstimmt, welches aus der Datenbank ausgelesen wurde. (In unserem Skript ist dabei `$pass` das ausgelesene Passwort und `$passwort` das eingegebene Passwort.)

Bei korrekter Eingabe ist die Bedingung der IF-Schleife erfüllt (if (\$passworteingabe == '\$richtigespasswort')) und es wird nun der gewünschte Inhalt angezeigt!

```
<?php

if ($passwort == '') {

    echo "Angezeigter Inhalt für die Passwortabfrage";

}

if ($passworteingabe == '$richtigespasswort') {

    echo "Angezeigter Inhalt bei Eingabe des richtigen Passworts";

}

else {
    echo "Angezeigter Inhalt bei Eingabe des falschen
        Passworts";
}

?>
```

Für die jeweiligen Inhalte in den Anführungsstrichen nach dem echo-Befehl können beliebige Inhalte, wie z.B. ein Formular oder sonstiger HTML-Code, eingesetzt werden.

Anzeige der persönlichen ECard

Der Empfänger hat das richtige Passwort in das Formular eingegeben und erwartet nun seine personalisierte ECard. Auch hier kommen wir um eine Datenbankabfrage nicht herum, die aber schon aus dem Abschnitt „Die Vorschaumöglichkeit“ bekannt ist.

Dieses Skript fügen Sie bitte vor dem <HTML>-Tag in der Seite ein.

```
<?php

require ("connect.inc.php");
$result = MySQL_query("SELECT * FROM ecards WHERE
    id=$id")
    or die("Fehler beim Auslesen der Datenbank !");

while($row = MySQL_fetch_array($result)) {

    $id=$row["id"];
    $pass=$row["passwort"];
    $absender=$row["absender"];
}
```

```

$empfaenger=$row["empfaenger"];
$inhalt=$row["inhalt"];
$tomail=$row["tomail"];
$frommail=$row["frommail"];
$datum=$row["datum"];

?>

```

Hinweis: Die While-Schleife ist in diesem Skript mit Absicht nicht geschlossen, es macht nur Sinn in Zusammenhang mit dem noch folgenden Code!

Der nächste Schritt bringt uns noch einmal zurück zu der Passwortabfrage, die wir nun mit folgendem Code in die Seite einbauen:

```

<?php

if($password == '') {

```

Dieser Teil erscheint beim ersten Aufruf der Seite. Die Bedingung lautet, dass die Passwort-Variable (Eingabe) leer ist. Sollte sie nicht leer sein, wird dieser Teil übersprungen.

```

echo" <form action=ecardview.php method=post>
<table width=470 border=0 cellspacing=0 cellpadding=0>
  <tr>
    <td>
      <div align=left><font size=2>Guten Tag! Geben Sie
      bitte Ihr Passwort an um sich die ECard ansehen
      zu &ouml;nnen :
      <input type=\"pass\" name=\"password\" class=form>
      <input type=\"hidden\" name=\"id\" value=\"$id\">
      <input type=\"submit\" name=\"go\" value=\"Login\"
      class=button></font></div>
    </td>
  </tr>
</table>
</form>

```

Eingabeformular für das Passwort. Wichtig ist, dass die ID-Nummer in einem versteckten Feld mit übergeben wird, da ansonsten die Datenbankabfrage beim Reload nicht funktionieren würde.

```

"; }

```

Schließen Sie das Skript an dieser Stelle noch nicht mit dem ?>-Tag, da nun gleich der Code für das Anzeigen des Flash-Films (ECard) folgt. Diesen kennen Sie ebenfalls schon aus dem Abschnitt „Die Vorschaumöglichkeit“.

```

if($pass == $password) {

echo"      <table width=470 border=0 cellspacing=0
           cellpadding=0>
           <tr>
             <td rowspan=3 width=105 valign=top>
               <div align=right></div>
             </td>
             <td width=260 height=10></td>
             <td rowspan=3 width=105 valign=top></td>
           </tr>
           <tr>
             <td width=260>
               <div align=center>" ;
?>

```

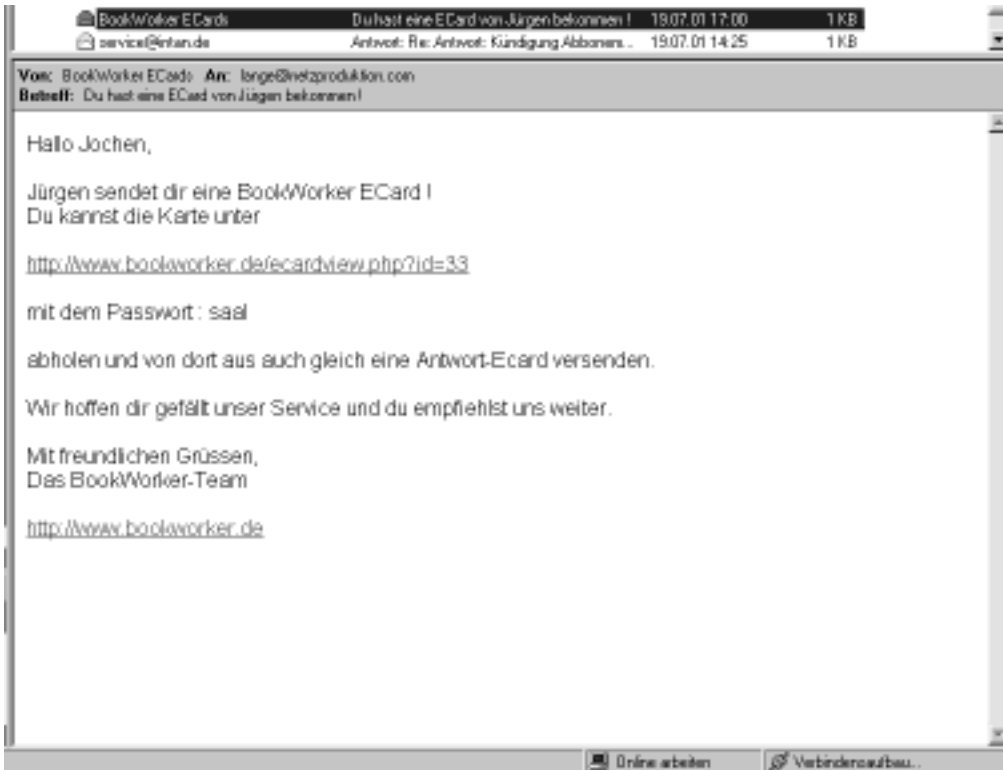
Das Skript muss vor Beginn des Flash-Einbettungscode geschlossen werden, da Flash die PHP-Variablen mit dieser Variante sonst nicht einlesen kann.

```

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase=http://download.macromedia.com/pub/shockwave/cabs/
flash/swflash.cab#version=5,0,0,0 width=260 height=200>

<param name=movie value="e_card.swf?id=<? echo "$id" ;?>&emp-
faenger=<? echo "$empfaenger" ;?>&absender=<? echo "$absender"
;?>&inhalt=<? echo "$inhalt" ;?>&tomail=<? echo "$tomail"
;?>&frommail=<? echo "$frommail" ;?>&datum=<? echo "$datum" ;?>
"><param name=quality value=high><embed src="e_card.swf?id=<?
echo "$id" ;?>&empfaenger=<? echo "$empfaenger" ;?>&absender=<?
echo "$absender" ;?>&inhalt=<? echo "$inhalt" ;?>&tomail=<? echo
"$tomail" ;?>&frommail=<? echo "$frommail" ;?>&datum=<? echo
"$datum" ;?> "e_card.swf?id=<? echo "$id" ;?>&empfaenger=<? echo
"$empfaenger" ;?>&absender=<? echo "$absender" ;?>&inhalt=<?
echo "$inhalt" ;?>&tomail=<? echo "$tomail" ;?>&frommail=<? echo
"$frommail" ;?>&datum=<? echo "$datum" ;?> "" quality=high
width=280 height=220 type="application/x-shockwave-Flash"
PlugInpage=http://www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash ></embed></object>

```



— **Abbildung 2.65:**
Die erzeugte Mail in
Outlook Express

Im vorletzten Schritt folgt noch eine kurze Information, von wem die Karte ist.

```

    <?php echo"
        </div>
    </td>
</tr>
    <tr>
        <td width=260 height=10><font size=2></td>
    </tr>
</table>
<br>
<table width=470 border=0 cellspacing=0 cellpadding=0>
    <tr>
        <td>
            <div align=center><font size=2>
                Diese Karte wurde von $absender
                für $empfaenger hinterlegt.
            </font></div>
        </td>
    </tr>
</table>
"; }}
?>

```

Speichern Sie die Datei unter dem Namen `ECARDVIEW.PHP` ab. Beachten Sie, dass Sie beim Testen der Seite immer die Variable `ID` mit übergeben, sodass die Adresse immer `ecardview.php?id=beliebigenummer` (z.B. 2) lautet!

Testen Sie nun das ECard-System auf Ihrem Server, indem Sie ein paar ECards an sich selbst oder Freunde verschicken. Wichtig ist, darauf zu achten, dass die E-Mails immer ihren Empfänger erreichen und die ECard den richtigen Text anzeigt.



Flash-Spiel mit Highscore (Flash und PHP)

Azad Adsay

*„Ich beneide Menschen, die mit drei oder vier Stunden Schlaf in der Nacht prächtig zurechtkommen. Sie haben so viel mehr Zeit zu arbeiten, zu lernen und zu spielen.“
Bill Gates*

In diesem Kapitel werden wir ein kleines Flash-Spiel mit einer Highscore-Verbindung entwickeln. Aufgabe im Spiel ist es, die kleinen Kreise abzuschießen, die durch das Bild fliegen. Pro abgeschossenem Kreis gibt es Punkte, die später in einer Datenbank eingetragen werden.

Als Hindernis erweist sich, dass es Punktabzüge gibt, wenn man mit der Maus in den grünen Bereich kommt.

Die Variable des Highscore wird mit PHP an eine MySQL-Tabelle geschickt und dort gespeichert.



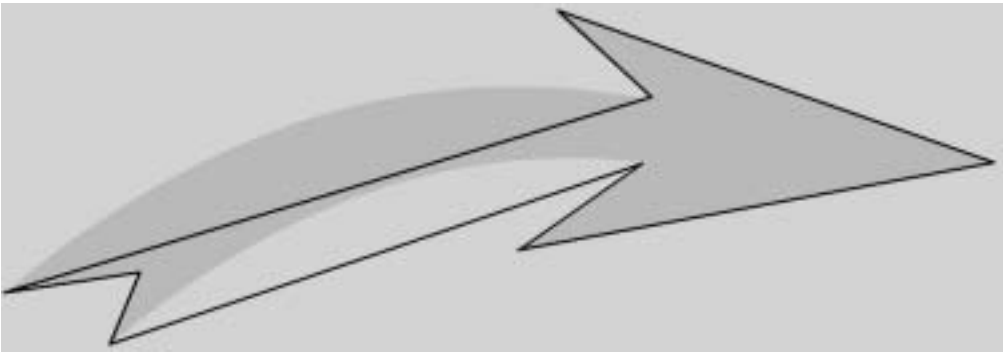
■ ■ ■ **Abbildung 3.1:** Das Flash-Spiel

3.1 Das Spiel programmieren

Die kleinen Kugeln fliegen in drei verschiedenen Größen durch das Bild. Für jeden Abschuss gibt es Punkte: für die kleinste Kugel 100 Punkte, für die mittlere 50 und für die große nur 25 Punkte. Die Kugeln fliegen anhand vorbestimmter Bahnen durch das Bild, jedoch variieren die Y-Werte. Unten rechts wird die verbleibende Zeit angegeben, welche insgesamt eine halbe Minute beträgt. Bei einer Bildzahl von 20 Bps bedeutet das, die Variable muss $30 \cdot 20$ bis auf 0 hinunterzählen.

Beginnen Sie, indem Sie den Hintergrund entwickeln. Zeichnen Sie einen grünen Bereich oben, mit einer Höhe von ca. 165 Pixel. Dazu zeichnen Sie ganz einfach ein Viereck, welches Sie dann über die Eigenschaften auf eine Höhe von 165 Pixel skalieren.

Zeichnen Sie nun den Pfeil, der die Flugrichtung anzeigt.



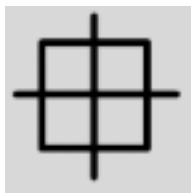
■ *Abbildung 3.2: Der Pfeil*

Im Hintergrund befindet sich der fertige graue Pfeil. Im schwarzen Rahmen sehen Sie, wie Sie den Pfeil anlegen. Nachdem Sie ihn so gezeichnet haben, verbiegen Sie die hinteren Linien. Füllen Sie den Pfeil aus und entfernen Sie die Umrandung wieder.

Zeichnen Sie nun drei Kugeln, die etwa so positioniert sind wie in der nachfolgenden Abbildung:



■ *Abbildung 3.3: Die drei Kugeln*



■ *Abbildung 3.4: Ein Fadenkreuz*

Zeichnen Sie jetzt ein Fadenkreuz wie in der Abbildung. Legen Sie es so an, dass es nicht zu groß, jedoch auch nicht zu klein ist. Die Größe des Fadenkreuzes bestimmt maßgeblich den Schwierigkeitsgrad. Formatieren Sie das Fadenkreuz zu einer Filmsequenz. In dieser Filmsequenz muss in einer unteren Ebene noch eine Fläche sein, welche genau hinter dem Fadenkreuz liegt. Dies gewährleistet einen kleinen Spielraum um das Fadenkreuz für die Spieler.

Weisen Sie dem Fadenkreuz die Instanz „jet“ sowie das nachfolgende ActionScript zu

```
onClipEvent(mouseMove){
    this._x = _root._xmouse;
    this._y = _root._ymouse;

    if (this.hitTest(_root.out)) {
        _parent.hit--;
    }

}
```

■ ■ ■ Listing 3.1:
Jet soll den Mauszeiger darstellen.

Die Jet-Filmsequenz, angesprochen mit `this`, übernimmt immer die aktuellen X- und Y-Positionen der Maus. Dafür gibt es die Funktion `_xmouse` und `_ymouse`.

Wenn der Mauszeiger dann noch den Sperrbereich (die obere grüne Fläche) berührt, werden wiederum Punkte abgezogen. Die Punkte werden in der Variablen `hit` gespeichert.

`onClipEvent()` ist das Gegenstück zu `on()` eines Knopfs. Je nach Art des Aufrufs (hier `load`) wird eine bestimmte Prozedur ausgeführt. Zum Beispiel wird das ActionScript bei `load` nur ausgeführt, wenn die Filmsequenz geladen wird – aber nicht öfter.

Erstellen Sie nun zwei Textfelder, die Sie in die Ecke rechts unten legen. Weisen Sie dem unteren die Variable `hit` zu. Hier wird ausgegeben, wie viele Punkte der Spieler erreicht hat. Das obere Textfeld formatieren Sie zu einer Filmsequenz und programmieren folgendes ActionScript hinzu:

```
onClipEvent (load) {
    time_left = 30*20;
}
onClipEvent (enterFrame) {
    if (time_left<=0) {
        if (_parent.hit>1000) {
            _root.scorer = _root.hit;
            _root.gotoandstop("win");
        } else {
            _root.gotoandstop("looser");
        }
    } else {
        time_left--;
    }
}
```

■ ■ ■ Listing 3.2:
Zeitkontrolle

Mit diesem kleinen Skript wird die Zeit überprüft. Wird der Clip geladen, wird die Zeit auf 30 Sekunden gesetzt (30x20, weil wir 20 Bps haben und 20 Bps = 1 Sekunde).

Während des Spielvorgangs zählt dieses Skript dann die Zeit herunter. Wenn die Zeit abgelaufen ist und die Punkte einen höheren Wert als tausend haben, springt Flash zur Szene „win“, andernfalls zur Szene „looser“ – diese Szenen werden später noch entwickelt.

Formatieren Sie jetzt die bereits angesprochenen Kugeln zu Filmsequenzen und weisen Sie ihnen die folgenden Instanzen zu: dem kleinen „ball1“, dem mittleren „ball2“ und dem großen „ball3“.

Die Filmsequenzen müssen in zwei Abschnitte unterteilt werden. Im ersten ist die Kugel zu sehen, im zweiten nicht mehr. Ein Schlüsselbild im zweiten Teil muss die Bezeichnung „off“ tragen!



■ **Abbildung 3.5: Definiertes Weg**

Die Kugel soll dieser Linie folgen. Dazu muss diese Ebene als „Führungsebene“ eingestellt und die Kugel in einer Ebene darunter zu finden sein. Jeder der Bälle bekommt ein ActionScript zugewiesen:

Listing 3.3: ■ ■ ■
Abschusskontrolle

```
onClipEvent (mouseDown) {
    if(this.hitTest(_root.jet)){
        this.gotoandplay("off");
        _parent.hit = _parent.hit+25;
    }
}
```

Die Funktion hitTest überprüft, ob etwas mit dem Objekt zusammenstößt, hier die Filmsequenz mit der Bezeichnung „jet“. Ist das der Fall, so wird die Szene „off“ abgespielt, hier ist die Kugel nicht mehr sichtbar.

Die Zeile _parent.hit = _parent.hit+25; variiert. Je nachdem, wie groß der Ball ist, muss er auch die entsprechende Punktzahl bringen (kleine Kugel 100 Punkte, mittlere 50 und die große Kugel 25).

`_parent` bedeutet, dass wir eine Ebene zurückgehen. Hier wechseln wir in die `_root`-Ebene, hätten also statt `_parent` auch `_root` einsetzen können.

Im ersten Schlüsselbild der Szene sollen nun die Bälle dupliziert werden.

```
duplicateMovieClip("jet", "jet", 1000);

often = 2;
balls = 3;
set = 2;

show = number(random(often)+1);
set_ball = number(random(balls)+1);

if (show == set) {
    i++;
    duplicateMovieClip("ball" add set_ball, "ball" add i, i);
    setProperty("ball", "_y", Number(random(200)+50));
}
```

■ ■ ■ *Listing 3.4:*
Das erste Schlüssel-
bild der Szene

Die Variablen `often` und `set` sollen die Häufigkeit des Erscheinens von Bällen regulieren. Bei

```
show = number(random(often));
```

wird eine Zahl von 0–2 (weil `often = 3`) per Zufallsgenerator ermittelt. Da wir jedoch richtige Werte wollen, addieren wir immer eins hinzu, das sieht dann so aus:

```
show = number(random(often)+1);
```

Wenn `show` gleich 2 ist, da `set = 2`, wird ein Ball dupliziert. Dieser wird ebenfalls per Zufallsgenerator ermittelt: Je nachdem, was `set_ball` ausgibt, wird „ball1“, „ball2“ oder „ball3“ eingefügt.

Die erste Zeile des Skripts (`duplicateMovieClip("jet", "jet", 1000);`) garantiert, dass das Fadenkreuz („jet“) ganz oben ist, also niemals hinter einem Ball liegt.

Mit der Funktion `duplicateMovieClip` können Filmsequenzen kopiert werden, deren Kopien mit zwei Attributen direkt spezifiziert werden. Dieser Vorgang findet in den Klammern statt. Zuerst wird die Quelle angegeben, also welcher Film dupliziert werden soll, anschließend der neue Name sowie das Level.

```
duplicateMovieClip("quelle", "neuer Name", Ebene);
```

■ ■ ■ *Listing 3.5:*
Die Funktion duplicate-
MovieClip();

3.2 Der Highscore auf Datenbankbasis

Kommen wir nun zu der Szene „win“, in der wir den Score und den Namen sowie die E-Mail des Spielers an eine Datenbank schicken wollen. Legen Sie zunächst eine Tabelle namens „highscore“ wie folgt auf Ihrem Server an:

Listing 3.6: ■■■
Tabelle „highscore“
anlegen

```
CREATE TABLE highscore (
  id int(11) NOT NULL auto_increment,
  name text NOT NULL,
  email text NOT NULL,
  highscore text NOT NULL,
  PRIMARY KEY (id)
) TYPE=MyISAM;
```

Wir entwickeln nun die „win“-Szene.

■■■ **Abbildung 3.6:** Der „win“-Abschnitt

Legen Sie eine neue Szene an, in der Sie das erste Schlüsselbild mit „win“ bezeichnen. Hier sind vier Textfelder zu sehen. Das oberste und das unterste Textfeld sind dynamisch und beinhalten die Variablen `scorer` und `status`. Die beiden Felder in der Mitte sind Input-Felder, das obere enthält die Variable `name`, das untere `email`.

Der Sendeknopf soll mit dem nachfolgenden ActionScript die Daten abschicken. Zur Sicherheit überprüfen wir hier noch einmal die Höhe des Score.

```

on (press) {
    if (scorer >= 1000) {
        loadVariablesNum ("highscore.php?status=input_data", 0,
            "get");

        status = "In den Highscore eingetragen, wir melden uns,
            wenn Sie gewonnen haben";

        scorer = 0;
    }
}

```

■■■ Listing 3.7:
Daten absenden

Wichtig ist noch ein ActionScript des ersten Schlüsselbildes des Abschnitts „win“ und des ersten Schlüsselbildes des Abschnitts „looser“:

```

for (kick in this) {
    this[kick].removeMovieClip();
}
stop();

```

■■■ Listing 3.8:
Alle Filmsequenzen
löschen

Mit diesem kleinen Skript werden alle Bälle gelöscht.

Bevor wir zum Highscore kommen, muss noch der Abschnitt „looser“ erstellt werden. Hier soll einfach nur eine Nachricht für Spieler angezeigt werden, die 1000 Punkte nicht erreichen.



■■■ Abbildung 3.7: Leider verloren!

3.3 Highscore-Ausgabe

Fahren wir nun fort mit der Datei HIGHSCORE.PHP. Sie soll nur dem internen Einsatz dienen, muss also grafisch nicht sehr aufwändig gestaltet werden:

Listing 3.9: ■■■
Die Highscore.php

```

<html>
<head>
<title>Highscore</title>
<meta http-equiv="Content-Type" content="text/html;
  charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" text="#000000">

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<?
include ("open.inc.php");
  if ($status != "input_data") {
    $content = MySQL_query ("SELECT * from highscore order by
      highscore asc LIMIT 0,5");

    while ($row = MySQL_fetch_array ($content)) {

?>

<tr>
  <td width="11%"><b><font face="arial" size="2">
    Highscore:</font></b></td>
  <td width="89%"> <font face="arial" size="2">
    <? echo $row[highscore]; ?>
  </font></td>
</tr>
<tr>
  <td width="11%"><b><font face="arial" size="2">
    Name:</font></b></td>
  <td width="89%"> <font face="arial" size="2">
    <? echo $row[name]; ?>
  </font></td>
</tr>
<tr>
  <td width="11%"><b><font face="arial" size="2">
    E-Mail</font></b></td>
  <td width="89%"> <font face="arial" size="2">
    <? echo $row[email]; ?>
  </font></td>
</tr>
<tr>
  <td width="11%">&nbsp;</td>

```

```

        <td width="89%">&nbsp;</td>
    </tr>
<? } ?>
</table>

<?
    } else {
        $sql_befehl = "INSERT INTO highscore (name, email, highscore)
        VALUES('$name', '$email', '$scorer')";
        $ergebnis = MySQL_query($sql_befehl,$conn);
    }

?>
</body>
</html>

```

Wird die Datei ohne `status = "input_data"` geladen, werden die fünf Einträge mit dem größten Score aufgelistet. Beim Aufruf mit `status = "input_data"` (wie in Flash) werden die Daten an die Tabelle „highscore“ gesendet und gespeichert.

Die Tabelle wird über `$content = MySQL_query ("SELECT * from highscore order by highscore asc LIMIT 0,5");` angesprochen. Das `asc` gibt die Art der Sortierung und das Limit die Anzahl der anzuzeigenden Einträge an. In diesem Beispiel werden nur die fünf Einträge mit dem größten Score angezeigt.

Die `OPEN.INC.PHP` wird für die Datenbankverbindung eingesetzt.

```

<?
$user   = "";
$pass   = "";
$server = "";
$dbase  = "buch";

$conn = @MySQL_connect($server, $user, $pass);

if($conn) {
    MySQL_select_db($dbase, $conn);
} else {
    die("Fehler");
}
?>

```

■ ■ ■ Listing 3.10:
*open.inc.php –
 Schnittstelle zur
 Datenbank*



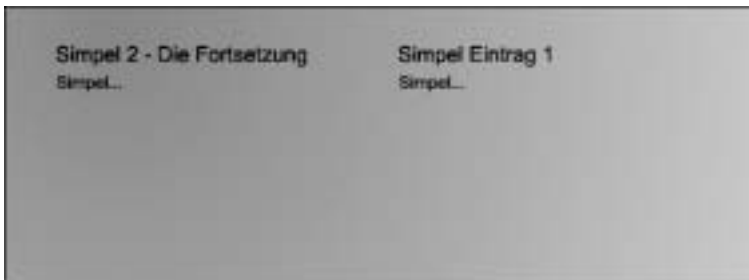
Kleines News-System (Flash und PHP)

Azad Adsay

*„Wir produzieren heute Informationen in
Massen, wie früher Automobile“.*

John Naisbitt

In diesem Kapitel werden wir ein News-System entwickeln, genauer gesagt zwei. Zunächst ein simples System, in dem nur ein Titel sowie ein kleiner Text zu sehen sind (siehe Abbildung 4.1) und außerdem ein komplexes System, das das Datum sowie den Zeitpunkt der Eintragung ausgibt. Es bietet Platz für den Namen des Autors sowie eine Verlinkung.

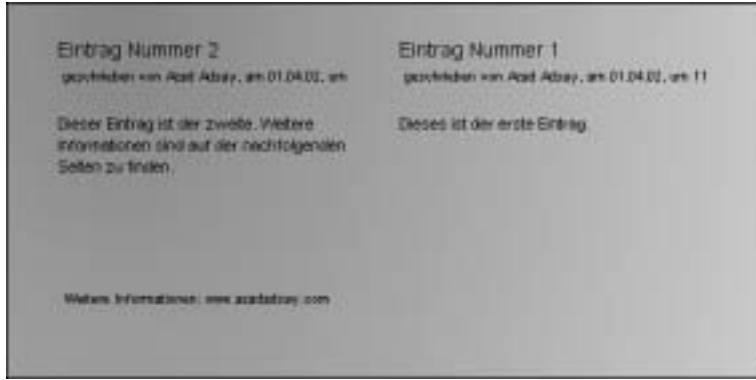


■ ■ ■ **Abbildung 4.1: Das simple System**

Die Dateneingabe erfolgt in einer HTML-Seite, die Ausgabe in Flash. Die Datensätze werden alle mit `rawurlencode()`; für Flash optimiert. Das komplexe sowie das simple System laden die letzten beiden Einträge aus der Datenbank. Dazu wird die `max(id)`, der Eintrag mit der größten ID, erfasst und dann ausgelesen. Anschließend wird der davor eingegebene Datensatz gelesen.

Das komplexe System baut später auf dem einfachen System auf.

Abbildung 4.2: ■■■
Das komplexe
News-System



4.1 Die Datenbank-Tabelle

Wir beginnen mit dem einfachen System und legen gleich eine neue Tabelle an:

Listing 4.1: ■■■
Tabelle „news“

```
CREATE TABLE news (
  id int(11) NOT NULL auto_increment,
  titel text NOT NULL,
  message text NOT NULL,
  PRIMARY KEY (id)
) TYPE=MyISAM;
```

In der Spalte „titel“ steht die Überschrift des jeweiligen Beitrags und in der Spalte „message“ der Text.

Legen Sie zuerst wieder die Datei OPEN.INC.PHP an, mit der Sie die Verbindung zu Ihrer Datenbank aufbauen:

Listing 4.2: ■■■
open.inc.php –
Verbindung zu einer
Datenbank aufbauen

```
<?
  $user = "Benutzername";
  $pass = "Passwort";
  $server = "localhost";
  $dbase = "buch";
  $table = "news";

$conn = @MySQL_connect($server, $user, $pass);
if($conn) {
  MySQL_select_db($dbase, $conn);
} else {
  die("Fehler");
}
?>
```

4.2 Daten auslesen

Danach folgt die Datei `READ_DB.PHP`. Diese Datei lädt alle Daten aus der Datenbank, formatiert sie und gibt sie aus.

```
<?
include ("open.inc.php");
$content = MySQL_query ("SELECT max(id) from $table");
$row = MySQL_fetch_array ($content);

$content = MySQL_query ("SELECT * from $table where id =
$row[0]");
$read_arow = MySQL_fetch_array ($content);
$arow[titel] = rawurlencode($read_arow[titel]);
$arow[message] = rawurlencode($read_arow[message]);

echo "&a_titel=$arow[titel]";
echo "&a_message=$arow[message]";

$content = MySQL_query ("SELECT * from $table where id =
$row[0]-1");
$read_brow = MySQL_fetch_array ($content);
$brow[titel] = rawurlencode($read_brow[titel]);
$brow[message] = rawurlencode($read_brow[message]);

echo "&b_titel=$brow[titel]";
echo "&b_message=$brow[message]";
?>

$content = MySQL_query ("SELECT max(id) from $table");
$row = MySQL_fetch_array ($content);
```

■ ■ ■ *Listing 4.3:*
read_db.php – Daten
aus der Datenbank
laden

Diese zwei Zeilen, die schon sehr weit am Anfang des Skripts stehen, ermitteln die maximale ID. In `$row[0]` wird die maximale ID schließlich ausgegeben. Sind also zum Beispiel 56 Einträge in der Datenbank, ist der Wert von `$row[0]` dementsprechend 56.

```
$content = MySQL_query ("SELECT * from $table where id =
$row[0]");
$read_arow = MySQL_fetch_array ($content);
$arow[titel] = rawurlencode($read_arow[titel]);
```

Hier wird der letzte Eintrag geladen, also dort, wo die maximale ID (`$row[0]`) ist. Zum Aufbau des Skripts ist noch Folgendes anzumerken: Wir verwenden fünf Arrays: zu Beginn das Array `$row`, das lediglich die maximale ID ausgibt, dann die Arrays `$read_arow` und schließlich `$arow`. Nachfolgend gibt es noch die gleichen Arrays, bloß heißt es `brow` statt `arow`. Die Arrays `$read_arow` bzw. `$read_brow` kommen direkt aus der Datenbank und sind noch nicht für Flash kodiert, die Arrays `$arow` bzw. `$brow` hingegen schon.

In der letzten Zeile dieses Codeschnipsels wurde `$read_arow[titel]` kodiert, der kodierte Wert ist dann in `$arow[titel]` gespeichert.

```
echo "&a_titel=$arow[titel]";
```

Der Flash-Film selbst braucht jetzt vier dynamische Textfelder: `a_titel` und `a_message` sowie die Gegenstücke `b_titel` und `b_message`.

Abbildung 4.3: ■■■
Das News-System
in Flash



Die Daten werden mit `loadVariables` eingeladen:

Listing 4.4: ■■■
Einladen der
Datensätze

```
loadVariablesNum("read_db.php", 0, "GET");
stop();
```

4.3 Daten eintragen

Jetzt folgt die Datei `ENTER.PHP`, welche die Daten in die Tabelle speichert:

Listing 4.5: ■■■
`enter.php`

```
<html>
<head>
<title>Eintragen</title>
<meta http-equiv="Content-Type" content="text/html;
  charset=iso-8859-1">
</head>
<? if ($function != "senden") { ??
<body bgcolor="#FFFFFF" text="#000000">
<form name="form1" method="post" action="enter.php?
  function=senden">
<table width="28%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="27%"><b><font face="arial" size="2">
      Titel</font></b></td>
    <td width="73%">
      <font face="arial" size="2">
        <input type="text" name="titel">
      </font>
    </td>
  </tr>
</table>
```

```

        </td>
    </tr>
    <tr>
        <td width="27%"><b><font face="arial" size="2">
            Nachricht</font></b></td>
        <td width="73%"> <font face="arial" size="2">
            <textarea name="message"></textarea>
        </font></td>
    </tr>
    <tr>
        <td width="27%"><b><font face="arial" size="2">
            </font></b></td>
        <td width="73%"> <font face="arial" size="2">
            <input type="submit" name="Abschicken"
                value="Abschicken">
        </font></td>
    </tr>
</table>
</form>
<?
} else {
include ("open.inc.php");
$sql_befehl = "INSERT INTO $table (titel, message)";
$sql_befehl .= "VALUES('$titel', '$message)";
$ergebnis = MySQL_query($sql_befehl,$conn);
echo "Daten gespeichert!";
}
?>
</body>
</html>

```

Ist der Wert der Variablen `$function` gleich „senden“, werden die Daten gespeichert. Ansonsten wird die Eingabemaske angezeigt.

4.4 Komplexes System mit erweiterter Tabelle

Das komplexe System baut auf dem einfachen System auf. Sie können also diese Datei weiter verwenden. Die `OPEN.INC.PHP` bleibt dieselbe. Die `READ_DB.PHP` weist die meisten Änderungen auf. Beginnen wir jedoch mit der neuen Tabelle:

Listing 4.6: ■■■
Tabelle
„news_komplex“

```
CREATE TABLE news_komplex (
  id int(11) NOT NULL auto_increment,
  autor text NOT NULL,
  titel text NOT NULL,
  message text NOT NULL,
  more_info text NOT NULL,
  more_info_url text NOT NULL,
  datum text NOT NULL,
  zeit text NOT NULL,
  PRIMARY KEY (id)
) TYPE=MyISAM;
```

Wir haben jetzt die folgenden neuen Spalten hinzubekommen: „Autor“, „more_info“, „more_info_url“ sowie „Zeit“ und „Datum“. Die Spalte „Autor“ soll den Namen des Autors speichern, welcher den News-Eintrag geschrieben hat. Die Spalte „more_info_url“ beinhaltet einen Link mit weiteren Informationen, „more_info“ erlaubt bei der späteren Ausgabe eine Bezeichnung für den Link. Die Spalten „Datum“ und „Zeit“ enthalten den Zeitpunkt der Erfassung des Nachrichteneintrags. Die Werte von „Datum“ sowie von „Zeit“ werden automatisch gespeichert.

Komplexes System: Daten auslesen

Die neue `READ_DB.PHP`:

Listing 4.7: ■■■
`read_db.php`

```
<?
include ("open.inc.php");
$content = MySQL_query ("SELECT max(id) from $table");
$row = MySQL_fetch_array ($content);

$content = MySQL_query ("SELECT * from $table where id =
  $row[0]");
$read_arow = MySQL_fetch_array ($content);
$arow[titel] = rawurlencode($read_arow[titel]);
$arow[message] = rawurlencode($read_arow[message]);
$arow[autor] = rawurlencode($read_arow[autor]);
$arow[more_info] = rawurlencode($read_arow[more_info]);
$arow[more_info_url] = rawurlencode($read_arow[more_info_url]);

echo "&a_titel=$arow[titel]";
```

```

echo "&a_message=$arow[message]";
echo "&a_info=geschrieben von $arow[autor], am
      $read_arow[datum], um $read_arow[zeit]";

if ($arow[more_info_url] != "") {
echo "&a_more_info=Weitere Informationen:
<a href='$arow[more_info_url]' target = '_blank'>
      $arow[more_info]</a>";
}

$content = MySQL_query ("SELECT * from $table
                        where id = $row[0]-1");
$read_brow = MySQL_fetch_array ($content);
$brow[titel] = rawurlencode($read_brow[titel]);
$brow[message] = rawurlencode($read_brow[message]);
$brow[autor] = rawurlencode($read_brow[autor]);
$brow[more_info] = rawurlencode($read_brow[more_info]);
$brow[more_info_url] = rawurlencode($read_brow[more_info_url]);

echo "&b_titel=$brow[titel]&b_message=$brow[mes-
sage]&b_info=geschrieben von $brow[autor], am $read_brow[datum],
um $read_brow[zeit]";

if ($brow[more_info_url] != "") {
echo "&b_more_info=Weitere Informationen:
<a href='$brow[more_info_url]' target = '_blank'>
      $brow[more_info]</a>";
}
?>

```

Wie auch schon in der einfachen Version lädt die Datei READ_DB.PHP die Datensätze aus der Tabelle. Die neuen Array-Teile müssen kodiert werden, die erste wirkliche Neuerung erfolgt hier:

```

echo "&a_info=geschrieben von $arow[autor], am
      $read_arow[datum], um $read_arow[zeit]";

```

Der Wert der Variablen a_info ist nicht nur ein einfacher Wert; er beinhaltet Text sowie den Namen des Autors, das Datum und die Zeit. Das Datum und die Zeit mussten nicht kodiert werden.

```

if ($arow[more_info_url] != "") {
echo "&a_more_info=Weitere Informationen:
<a href='$arow[more_info_url]' target = '_blank'>
      $arow[more_info]</a>";
}

```

Hier wird überprüft, ob \$arow[more_info_url] einen Wert hat. Ist dies nicht der Fall, dann wird auch keine Variable a_more_info ausgegeben. Die Variable a_more_info beinhaltet alle Informationen für die spätere Weiterverlinkung aus dem News-System heraus.

Noch eine kleine Erklärung am Rand: Bisher habe ich alle Variablen, die an Flash weitergegeben werden, einzeln mit echo ausgegeben. Dies kann natürlich auch anders gelöst werden.

```
echo "&a_titel=$arow[titel]";
echo "&a_message=$arow[message]";
echo "&a_info=geschrieben von $arow[autor], am
      $read_arow[datum], um $read_arow[zeit]";
```

Stattdessen können die Variablen auch direkt aneinander gekettet werden:

```
echo "&b_titel=$brow[titel]&b_message=$brow[message]&b_info=
      geschrieben von $brow[autor], am $read_brow[datum], um
      $read_brow[zeit]";
```

Aufgrund der besseren Übersichtlichkeit entschied ich mich dafür, die Variablen einzeln auszugeben.

Komplexes System: Daten eintragen

Die Datei ENTER.PHP speichert auch hier die Daten ab, wenn \$function gleich „senden“ ist:

Listing 4.8: ■■■
enter.php

```
<html>
<head>
<title>eintragen</title>
<meta http-equiv="Content-Type" content="text/html;
      charset=iso-8859-1">
</head>
<? if ($function != "senden") { ?>
<body bgcolor="#FFFFFF" text="#000000">
<form name="form1" method="post" action="enter.php?
      function=senden">
  <table width="41%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td width="40%"><b><font face="arial" size="2">
        Titel</font></b></td>
      <td width="60%"> <font face="arial" size="2">
        <input type="text" name="titel">
        </font> </td>
    </tr>
    <tr>
      <td width="40%"><font face="arial" size="2">
        Autor:</font></td>
      <td width="60%"><font face="arial" size="2">
        <input type="text" name="autor">
        </font></td>
    </tr>
```

```

<tr>
  <td width="40%"><b><font face="arial" size="2">
    Nachricht</font></b></td>
  <td width="60%"> <font face="arial" size="2">
    <textarea name="message"></textarea>
  </font></td>
</tr>
<tr>
  <td width="40%"><font face="arial" size="2">
    Link mit weiteren Informationen:</font></td>
  <td width="60%"><font face="arial" size="2">
    <input type="text" name="more_info_url">
  </font></td>
</tr>
<tr>
  <td width="40%"><font face="arial" size="2">Bezeichnung
    f&uuml;r weitere Informationen:</font></td>
  <td width="60%"><font face="arial" size="2">
    <input type="text" name="more_info">
  </font></td>
</tr>
<tr>
  <td width="40%"><b><font face="arial" size="2">
    </font></b></td>
  <td width="60%"> <font face="arial" size="2">
    <input type="submit" name="Abschicken"
    value="Abschicken">
  </font></td>
</tr>
</table>
</form>
<?
} else {
include ("open.inc.php");

$datum = (date("d.m.y"));
$zeit = (date("H:i"));

$sql_befehl = "INSERT INTO $table (autor, titel, message,
    more_info, more_info_url, datum, zeit)";

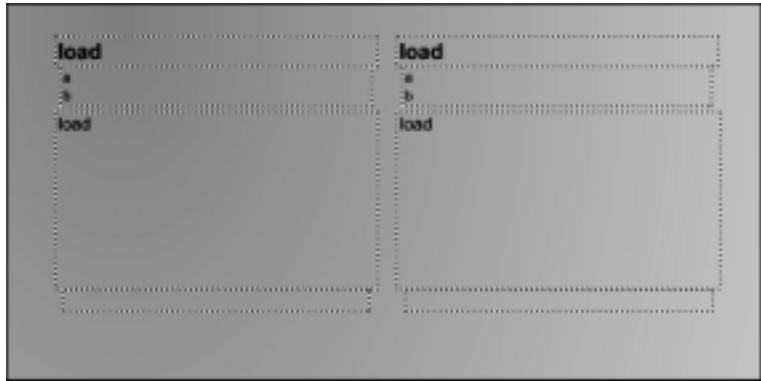
$sql_befehl .= "VALUES('$autor', '$titel', '$message',
    '$more_info', '$more_info_url', '$datum', '$zeit)";

$ergebnis = MySQL_query($sql_befehl,$conn);
echo "Daten gespeichert!";

}

?>
</body>
</html>

```



■ ■ ■ *Abbildung 4.4: Die Datenausgabe in Flash*

Wie Sie in *Abbildung 4.4* sehen können, sind vier neue Textfelder hinzugekommen.



Newsletter (PHP)

Azad Adsay

„Je höher die Technologie, desto höher das Kontaktbedürfnis“.

*John Naisbitt (*1930),
amerik. Prognostiker*

In diesem Kapitel entwickeln wir ein kleines Newsletter-System. E-Mail-Adressen werden in eine MySQL-Datenbank eingetragen, um danach spielend einfach E-Mails an alle eingetragenen Adressaten zu versenden. Trägt sich jemand ein, bekommt er eine Bestätigungsmail. Die Adressen werden an eine Datenbank geschickt, um mit einem Skript ausgelesen und nacheinander verschickt zu werden. Ziel dieses Kapitels ist es, das bereits vermittelte Vorwissen im Umgang mit Flash und PHP weiter zu festigen.

Beginnen Sie mit dem Anlegen einer neuen Tabelle:

```
CREATE TABLE newsletter (  
  id int(11) NOT NULL auto_increment,  
  email text NOT NULL,  
  name text NOT NULL,  
  UNIQUE KEY id (id)  
) TYPE=MyISAM;
```

■ ■ Listing 5.1:
Die Tabelle
„newsletter“

Nachfolgend setzen wir drei Dateien ein: NEWSLETTER.PHP, SENDEN.PHP und die Datenbankverbindungsdatei OPEN.INC.PHP.

5.1 Mailadressen eintragen

Die Datei NEWSLETTER.PHP ist in zwei wesentliche Funktionen unterteilt: enter und save. Mit Hilfe der enter-Funktion können sich die Interessenten eintragen, die Funktion save dient zum Speichern der Daten in der Datenbank:

Listing 5.2: ■■■
Der Inhalt der Datei
newsletter.php

```
<html>
<head>
<title>Eintraget</title>
<meta http-equiv="Content-Type" content="text/html;
      charset=iso-8859-1">

<? include ("open.inc.php"); ?>

</head>
<body bgcolor="#FFFFFF" text="#000000">

<?
if ($function == "enter") {
?>

<form name="form1" method="post" action="newsletter.php?
      function=save">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><b><font face="arial" size="2">Name:</font></b></td>
<td> <font face="arial" size="2">
<input type="text" name="name">
</font></td>
</tr>

<tr>
<td><b><font face="arial" size="2">E-Mail:</font></b></td>
<td> <font face="arial" size="2">
<input type="text" name="email">
</font></td>
</tr>

</table>
<font face="arial">
<input type="submit" name="Abschicken" value="Abschicken">
</font>

</form>

<?
}
```

```

if ($function == "save") {
    $sql_befehl = "INSERT INTO newsletter (email, name)";
    $sql_befehl .= "VALUES('$email', '$name')";
    $ergebnis = MySQL_query($sql_befehl,$conn);

    $send_name = "automatische e-mail";
    $send_mail = "no.replay@work4.de";
    $send_title = "Azads Newsletter";

    $message = "Hallo $name\nDu bist erfolgreich in unseren
                Newsletter eingetragen.";

    mail($email, $title, $message, "From: $send_name
    <$send_mail>\nReply-To:$send_mail");
}
?>
</body>
</html>

$send_name = "automatische e-mail";
$send_mail = "no.replay@work4.de";
$send_title = "Azads Newsletter";

$message = "Hallo $name\nDu bist erfolgreich in unseren
            Newsletter eingetragen.";

mail($email, $title, $message, "From: $send_name
<$send_mail>\nReply-To:$send_mail");

```

■ ■ ■ *Listing 5.3:*
Bestätigungsmails

Die Werte der Variablen `$send_name`, `$send_mail`, `$send_title` und `$message` sind frei variierbar. Die Funktion `mail()`; schickt die Daten dann ab.

```

if ($function == "save") {
    $sql_befehl = "INSERT INTO newsletter (email, name)";
    $sql_befehl .= "VALUES('$email', '$name')";
    $ergebnis = MySQL_query($sql_befehl,$conn);

```

■ ■ ■ *Listing 5.4:*
Daten speichern

Die Daten der Variablen `$email` und `$name` werden an die Tabelle „newsletter“ gesendet und gespeichert.

5.2 Newsletter senden

Auch diese Datei ist in zwei grundlegende Funktionen unterteilt. Die Funktion `send` verschickt die Daten mittels einer `while()`-Schleife, die Funktion `newsletter` ermöglicht die Eingabe der Daten.

Listing 5.5: ■■■
Die Datei `senden.php`

```

<html>
<head>
<title>senden</title>
<meta http-equiv="Content-Type" content="text/html;
      charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<?
    include ("open.inc.php");
    if ($function == "send") {
?>

<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>
<div align="center"><font face="arial" size="2"><b>
    Gesendete E-Mails:</b></font></div>
</td>
</tr>

<?
$content = MySQL_query ("SELECT * from newsletter");
while ($row = MySQL_fetch_array ($content)) {
$send[name] = "automatische e-mail";
$send[mail] = "no.replay@work4.de";

mail($row[email], $head, $text, "From: $send[name]
    <$send[email]>\nReply-To:$send[email]");

?>

<tr>
<td>
<div align="center"><font face="arial"><font size="2">
<? echo $row[email] ?>
</font></font></div>
</td>
</tr>

<?
    }
?>
</table>

```

```

<?
}
if ($function == "newsletter") {
?>

<form name="form2" method="post" action="senden.php?
    function=send">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><b><font face="arial" size="2">Betreff:</font></b></td>
<td>
<font face="arial" size="2">
<input type="text" name="head" size="50">
</font>
</td>
</tr>

<tr>
<td><b><font face="arial" size="2">Nachricht:</font></b></td>
<td><font face="arial" size="2">
<textarea name="text" cols="50" rows="5"></textarea>
</font></td>
</tr>

</table>

<input type="submit" name="Abschicken2" value="Abschicken">
</form>

<?
}
?>

</body>
</html>

<?
$content = MySQL_query ("SELECT * from newsletter");
while ($row = MySQL_fetch_array ($content)) {

```

■ ■ Listing 5.6:
While-Schleife

Durch die while()-Schleife wird an jede E-Mail-Adresse in der Tabelle eine E-Mail geschickt.

```

$send[name] = "automatische e-mail";
$send[mail] = "no.replay@work4.de";

mail($row[email], $head, $text, "From: $send[name]
<$send[email]>\nReply-To:$send[email]");

?>

```

■ ■ Listing 5.7:
Festlegen der E-Mail

In \$send[name] und \$send[mail] definieren wir einen Absender.

5.3 Datenbankanbindung

Abschließend schreiben Sie die Datei `OPEN.INC.PHP`, welche die Datenbankverbindung öffnet:

```
<?
$user  = "Benutzername";
$pass  = "Passwort";
$server = "localhost";
$dbase  = "buch";
$conn  = @MySQL_connect($server, $user, $pass);
if($conn) {
    MySQL_select_db($dbase, $conn);
} else {
    die("Fehler");
}
?>
```



Feedback-Protokoll (Flash und PHP)

Azad Adsay

*„Besser kritisiert als ignoriert. Ich kann
wunderbar leben mit schlechter Kritik.“*

Frank Farian



■■■ **Abbildung 6.1:** Das Layout des Gästebuchs orientiert sich an Claudia Blacks Website.

In diesem Kapitel werden wir eine Erweiterung der Website von Claudia Black entwickeln. In diesem Projekt soll der Besucher die Möglichkeit erhalten, das neue Logo von Claudia Black zu kommentieren. Die Besucher sollen ihren Namen, ihre E-Mail-Adresse, die Adresse ihrer Webseite (sofern vorhanden) sowie einen Kommentar einfügen. Dazu werden die IP-Daten protokolliert.

Das Interface wird in Flash entwickelt, die Datenübermittlung mit der MySQL-Datenbank erfolgt über PHP.

Um eventuell auftretende Probleme bezüglich des Cache zu vermeiden, empfehle ich Ihnen, zunächst das Kapitel 5 zu lesen.

6.1 Tabelle anlegen

Beginnen wir damit, eine Tabelle in einer Datenbank anzulegen.

Listing 6.1: ■■■
*MySQL-Befehl für die
Datenbank*

```
CREATE TABLE kritik (
  id int(11) NOT NULL auto_increment,
  date text NOT NULL,
  name text NOT NULL,
  email text NOT NULL,
  web text NOT NULL,
  text text NOT NULL,
  ip text NOT NULL,
  UNIQUE KEY id (id)
)
```

6.2 PHP-Programmierung

In diesem Abschnitt werden wir die PHP-Module entwickeln, welche die Kommunikation mit der Datenbank und Flash organisieren. Wir verwenden die folgenden fünf Dateien:

- `admin.php`
Ermöglicht die Administrierung der Datenbank über das Flashinterface, d.h., der Kunde kann die Einträge sehr einfach über Flash ändern.
- `check_id.php`
Lädt immer den aktuellen Datenbankeintrag. Wird also weitergeblättert im Interface, überträgt diese Datei die aktuellen Daten.
- `enter.php`
Speichert neue Einträge ab.
- `open.inc.php`
Enthält die Zugangsdaten zur Datenbank und stellt die Verbindung zur Datenbank her.
- `read_db.php`
Lädt den letzten Eintrag der Datenbank beim Aufrufen der Seite.

6.3 PHP-Dateien

Nachfolgend ist der Quellcode in PHP für das Gästebuch aufgeführt. Ich verzichte auf detaillierte Beschreibungen, bei Fragen oder Problemen mit PHP bzw. den einzelnen PHP-Befehlen finden Sie Erläuterungen in der PHP-Einführung.

open.inc.php

Wir beginnen mit der Datei `OPEN.INC.PHP`. Hier werden die Passwörter gespeichert und die Verbindung zur Datenbank hergestellt. Alle PHP-Dateien müssen diese Datei benutzen, um mit der Datenbank kommunizieren zu können.

```
<?
$server = "localhost";
$user = "Benutzerbezeichnung";
$password = "Ihr Passwort";
$dbname = "Name der Datenbank";
$db = MySQL_connect($server,$user,$password);
MySQL_select_db($dbname,$db);
?>
```

■ ■ Listing 6.2:
*Erstellen einer
Verbindung zur
Datenbank*

Die Variablen `$server`, `$user` sowie `$password` müssen für den Verbindungsaufbau angegeben werden. Die Variable `$dbname` gibt schließlich noch den Namen der Datenbank an. Die Funktion `MySQL_connect()`; stellt die Verbindung zur Datenbank her; die Funktion `MySQL_select_db()`; wählt die Datenbank aus.

admin.php

In der Flash-Seite kann der Kunde die Datenbank bearbeiten. Damit die Einträge gespeichert werden, benutzen wir die Datei `ADMIN.PHP`, die sich wie folgt aufbaut:

```
<?
include("open.inc.php");
$update = "update kritik set date = '$date',";
$update .= "name = '$name',";
$update .= "email = '$email',";
$update .= "web = '$web',";
$update .= "text = '$text'";
$update .= "where id = $No";
$result = MySQL_query($update);
?>
```

■ ■ Listing 6.3:
*Update eines
MySQL-Datensatzes*

Diese Datei muss zuerst, wie alle anderen PHP-Dateien, die Datei `OPEN.INC.PHP` laden. Dies geschieht mit `include()`. Mit dieser Funktion wird die Zeile mit der `id = $no` upgedatet. Das heißt, wenn `$no = 3`, erfolgt ein Update der Zeile drei.

check_id.php

Diese Datei lädt immer die aktuell anzuzeigenden Dateien. Blättert ein Besucher in den Einträgen, lädt diese Datei die aktuellen Daten aus der Datenbank.

Listing 6.4: ■■■
Auslesen von Einträgen

```
<?
include("open.inc.php");

$get_date = "SELECT date FROM kritik WHERE id = '$no'";
$date_result = MySQL_query($get_date);
$return = MySQL_result($date_result, 0, "date");
$send_date = "&date=";
$send_date .= rawurlencode($return);
echo "$send_date";

$get_name = "SELECT name FROM kritik WHERE id = '$no'";
$name_result = MySQL_query($get_name);
$return = MySQL_result($name_result, 0, "name");
$send_name = "&name=";
$send_name .= rawurlencode($return);
echo "$send_name";

$get_email = "SELECT email FROM kritik WHERE id = '$no'";
$email_result = MySQL_query($get_email);
$return = MySQL_result($email_result, 0, "email");
$send_email = "&email=";
$send_email .= rawurlencode($return);
echo "$send_email";

$get_web = "SELECT web FROM kritik WHERE id = '$no'";
$web_result = MySQL_query($get_web);
$return = MySQL_result($web_result, 0, "web");
$send_web = "&web=";
$send_web .= rawurlencode($return);
echo "$send_web";

$get_text = "SELECT text FROM kritik WHERE id = '$no'";
$text_result = MySQL_query($get_text);
$return = MySQL_result($text_result, 0, "text");
$send_text = "&text=";
$send_text .= rawurlencode($return);
echo "$send_text";
```

```

$get_ip = "SELECT ip FROM kritik WHERE id = '$no'";
$ip_result = MySQL_query($get_ip);
$return = MySQL_result($ip_result, 0, "ip");
$send_ip = "&ip=";
$send_ip .= rawurlencode($return);
echo "$send_ip";
?>

```

Als Beispiel möchte ich hier noch einmal folgendes Fragment des Codes erläutern:

```

$get_name = "SELECT name FROM kritik WHERE id = '$no'";
$name_result = MySQL_query($get_name);
$return = MySQL_result($name_result, 0, "name");
$send_name = "&name=";
$send_name .= rawurlencode($return);
echo "$send_name";

```

■ ■ ■ *Listing 6.5:*
Auszug aus
check_id.php

Die Spalte „name“ mit der ID im Wert der Variablen \$no wird aus der Datenbank geladen und danach für Flash tauglich ausgegeben.

enter.php

Fahren wir nun mit der Datei ENTER.PHP fort, welche die Eintragungen in die Datenbank vollzieht:

```

<?
include("open.inc.php");
$date = (date("d.m.Y, H:i"));
$enter = "INSERT INTO kritik (id,date,name,email,web,text,ip)
VALUES
('$id','$date','$name','$email','$web','$text','$REMOTE_ADDR)";
$result = MySQL_query($enter);
?>

```

■ ■ ■ *Listing 6.6:*
Daten eintragen

In der Variablen \$date führen wir die Funktion date() aus, welche das aktuelle Datum ausgibt, und zwar wie folgt: date ("Tag,.Monat,.Jahr, Stunden:Minuten"). Die Variable \$REMOTE_ADDR enthält die IP-Adresse des Besuchers.

read_db.php

Abschließend folgt die Datei READ_DB.PHP, welche den ersten Eintrag in das Flashinterface lädt. Es soll immer der letzte Eintrag der Datenbank zu Beginn angezeigt werden:

Listing 6.7: ■■■
Code der read_db.php

```
<?php
include("open.inc.php");

$get_count = "SELECT count(id) FROM kritik";
$count_result = MySQL_query($get_count);
$count = MySQL_result($count_result, 0, "count(id)");

$get_date = "SELECT date FROM kritik WHERE id = '$count'";
$date_result = MySQL_query($get_date);
$return= MySQL_result($date_result, 0, "date");
$send_date = "&date=";
$send_date .= rawurlencode($return);
echo "$send_date";

$get_id = "SELECT id FROM kritik WHERE id = '$count'";
$id_result = MySQL_query($get_id);
$return= MySQL_result($id_result, 0, "id");
$send_id = "&No=";
$send_id .= rawurlencode($return);
echo "$send_id";

$get_name = "SELECT name FROM kritik WHERE id = '$count'";
$name_result = MySQL_query($get_name);
$return = MySQL_result($name_result, 0, "name");
$send_name = "&name=";
$send_name .= rawurlencode($return);
echo "$send_name";

$get_email = "SELECT email FROM kritik WHERE id = '$count'";
$email_result = MySQL_query($get_email);
$return = MySQL_result($email_result, 0, "email");
$send_email = "&email=";
$send_email .= rawurlencode($return);
echo "$send_email";

$get_web = "SELECT web FROM kritik WHERE id = '$count'";
$web_result = MySQL_query($get_web);
$return = MySQL_result($web_result, 0, "web");
$send_web = "&web=";
$send_web .= rawurlencode($return);
echo "$send_web";

$get_text = "SELECT text FROM kritik WHERE id = '$count'";
$text_result = MySQL_query($get_text);
$return = MySQL_result($text_result, 0, "text");
$send_text = "&text=";
$send_text .= rawurlencode($return);
echo "$send_text";
```

```

$get_ip = "SELECT ip FROM kritik WHERE id = '$count'";
$ip_result = MySQL_query($get_ip);
$return = MySQL_result($ip_result, 0, "ip");
$send_ip = "&ip=";
$send_ip .= rawurlencode($return);
echo "$send_ip";

$get_No_max = "SELECT id FROM kritik WHERE id = '$count'";
$No_max_result = MySQL_query($get_No_max);
$return = MySQL_result($No_max_result, 0, "id");
$send_No_max = "&Nomax=";
$send_No_max .= rawurlencode($return);
echo "$send_No_max";
?>

$get_web = "SELECT web FROM kritik WHERE id = '$count'";
$web_result = MySQL_query($get_web);
$return = MySQL_result($web_result, 0, "web");
$send_web = "&web=";
$send_web .= rawurlencode($return);
echo "$send_web";

```

■ ■ ■ Listing 6.8:
Datensatz mit der
read_db.php auslesen

In diesem Auszug wird der Wert der Variablen \$send_web aus der Tabelle geladen (dort, wo der Wert der id dem Wert der Variablen \$count entspricht), kodiert und gesendet.

index.php

Die Datei INDEX.PHP ist eine ganz einfache HTML-Seite mit eingebundenem Flash. Jedoch müssen Sie bei der Einbindung Folgendes per Hand nachbearbeiten: Statt einer Verknüpfung zu KRITIK.SWF im HTML-Code muss auf kritik.swf?status=<? echo \$status ?>&password=<? echo \$password ?> verwiesen werden. So werden die Variablen status und password mit übergeben. Diese Variablen sind für den Administriermodus wichtig, mehr dazu folgt später. Dies sieht dann wie folgt aus:

```

<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/
Flash/swFlash.cab#version=5,0,0,0" width="500" height="350">
<param name=movie value="kritik.swf?status=<? echo $status
?>&password=<? echo $password ?>">
<param name=quality value=high>
<embed src="kritik.swf?status=<? echo $status ?>&password=<?
echo $password ?>" quality=high Plug-Inspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash" type="application/x-shockwave-Flash" width="500" height="350">
</embed>
</object>

```

■ ■ ■ Listing 6.9:
Das Flash wird später
mit Modifikationen
eingebunden.

6.4 Flash

Das Flash ist in fünf grundsätzliche Abschnitte unterteilt: Preload, Intro, Eingabe, Ausgabe und Admin. Es wird in zwei Szenen aufgeteilt. In der ersten Szene wird der gesamte Film geladen, mit einem sehr simplen Preloader. Im Intro wird die Seite aufgebaut und die Motivation dieser Befragung erläutert. Im Eingabeabschnitt können die Besucher ihre Meinungen eintippen, im Ausgabemodus können sie Meinungen lesen. Der Admin-Abschnitt ermöglicht die Bearbeitung der Datenbankeinträge im Flashinterface.

Die Größe beträgt 500x350 Pixel, der Hintergrund ist wie bei Claudia Blacks Homepage grau (#CECFCE).

Szene 1: Preloading

Beginnen wir mit einem kleinem Preloader, welcher den gesamten Film lädt. Der Besucher sieht nur Claudia Blacks Logo und den Schriftzug „Preloading“. Dem ersten Frame weisen Sie das folgende ActionScript zu:

Listing 6.10: ■■■
Ein simples Preloading

```
if (_framesloaded ==_totalframes) {
    gotoAndPlay("Scene 2", 1);
}
```

Wenn die geladenen Frames identisch sind mit allen im Film vorhandenen Frames, springt Flash zu Szene 2. `_framesloaded` ermittelt die Anzahl der geladenen Frames, `_totalframes` die Anzahl aller Schlüsselbilder in dem aktuellen Film.

In Schlüsselbild 2 wird zurück zu Schlüsselbild 1 gesprungen, so dass eine Schleife entsteht:

Listing 6.11: ■■■
Zurück zu Bild 1 – so entsteht eine Schleife

```
gotoandplay (1);
```



■ ■ ■ *Abbildung 6.2: Der Preloader besteht aus dem Logo von Claudia Black.*

Szene 2: Intro

Wir verzichten hier auf ein langes und Kilobyte fressendes Intro und lassen lediglich eine Erklärung zur Motivation der Umfrage erscheinen:



■ ■ ■ *Abbildung 6.3: Es sind eindeutige Parallelen zur Webseite von Claudia Black zu erkennen.*

Der nachfolgende Text soll jetzt erscheinen:

„Willkommen zu dieser "Umfrage". Seit 2 Wochen habe ich ein neues Logo. Nun möchte ich wissen, was meine Besucher, also Sie, über das Schmuckstück denken.

Bitte geben Sie mir hier Ihre Meinungen. Danke.“

Zusätzlich werden zwei Knöpfe, „Meinung geben“ und „Andere Meinungen lesen“ eingeblendet und Claudia Blacks Logo erscheint.

Abbildung 6.4: ■■■
Willkommen zu dieser
Umfrage ...





Verknüpfen Sie später die Knöpfe mit den entsprechenden Szenen. Weisen Sie dem ersten Schlüsselbild noch folgendes ActionScript zu:

Listing 6.12: ■■■
Laden des ersten
Datensatzes

```
loadVariablesNum ("read_db.php", 0, "GET");
```

Daten auslesen

In diesem ersten Hauptabschnitt können die Besucher andere Meinungen lesen. Setzen Sie in jede Ebene ein neues Schlüsselbild. Markieren Sie dazu alle Schlüsselbilder, indem Sie das letzte Bild der obersten Ebene und das der untersten Ebene mit gedrückter -Taste anklicken. Drücken Sie nachfolgend .

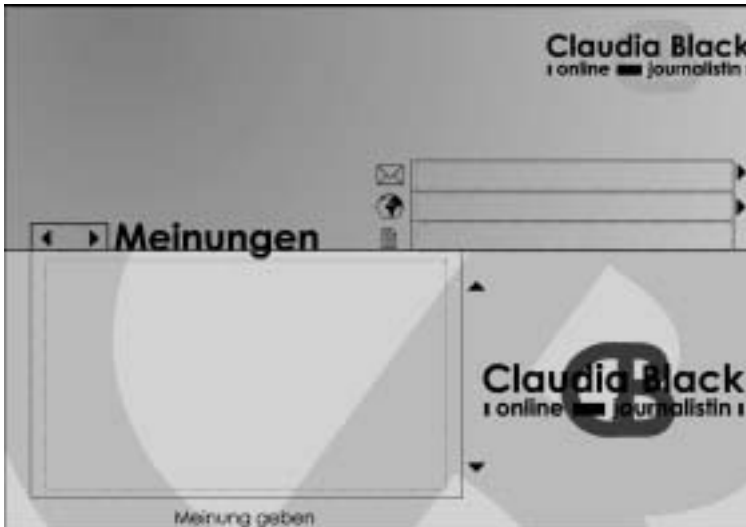
Das Logo bleibt weiterhin unten rechts zu sehen, während die fünf neuen Textfelder hinzugekommen sind.

Im ersten Textfeld, Mitte links, wird die Nummer des aktuellen Eintrags wiedergegeben, links und rechts daneben befinden sich Pfeile zum Blättern. In den Textfeldern rechts sollen von oben nach unten der Name, die Webseite und das Datum der Eintragung angegeben werden. Ein Klick auf den Pfeil neben dem Namen ermöglicht es, eine E-Mail an

den Verfasser des Beitrags zu schreiben, über einen Klick auf den unteren Pfeil besucht man die Webseite.

Im letzten Textfeld wird der Kommentar wiedergegeben, die Pfeile daneben ermöglichen das Scrollen. Diese können Sie aus dem Claudia-Black-Kapitel übernehmen.

Löschen Sie jetzt noch den Knopf „Andere Meinungen lesen“.



■ ■ ■ *Abbildung 6.5:*
Meinungen lesen

Die fünf dynamischen Textfelder müssen alle eine Variable beinhalten. Welche dies jeweils sein soll, erfahren Sie nun:

- das Textfeld zum Blättern in den Einträgen (Mitte links) erhält die Variable `no`
- die Textfelder zur Anzeige des aktuellen Eintrags (Mitte rechts) erhalten für das oberste Textfeld `name`, für das zweite Feld `web` und für das dritte `date`.
- Dem Kommentarbereich weisen Sie die Variable `text` zu.

Wie Sie vielleicht bemerkt haben, kommen diese Variablenbezeichnungen bereits in den PHP-Skripten vor.

Benennen Sie dieses Schlüsselbild jetzt „Vor“ und fügen Sie ein `stop()`; ein. Eine Position weiter schreiben Sie folgendes ActionScript in ein leeres Schlüsselbild:

```
if (Number(_root.no) == Number(_root.nomax)) {
    _root.no = "1";
    LoadVariablesNum ("check_id.php", 0, "GET");
} else {
    _root.no = Number(_root.no)+1;
    LoadVariablesNum ("check_id.php", 0, "GET");
}
```

■ ■ ■ *Listing 6.13:*
Mit Flash in den Daten-
bankeinträgen blättern

Dieses Skript erlaubt es, in den Einträgen nach vorne zu blättern. Im Schlüsselbild danach muss wieder auf „Lesen“ verwiesen werden:

Listing 6.14: `gotoAndStop ("Lesen");`

Daten anzeigen

Danach muss auch wieder eine Funktion zum Rückblättern eingebaut werden.

Benennen Sie dieses Schlüsselbild „Zurueck“ und schreiben Sie das folgende Skript in das leere Schlüsselbild:

Listing 6.15: `if (Number(_root.No) != 1) {
_root.No = _root.No-1;
loadVariablesNum ("check_id.php", 0, "GET");
}`

Zurückblättern

Jedes nachfolgende Schlüsselbild bekommt wieder einen Verweis auf „Lesen“:

Listing 6.16: `gotoAndStop ("Lesen");`

Daten anzeigen

Weisen Sie den Pfeilen neben der Anzeige zum aktuellen Eintrag nun die Verweise zu den entsprechenden Filmpositionen zu, also dem linken Pfeil „Zurueck“ und dem rechten „Vor“.

Einträge schreiben

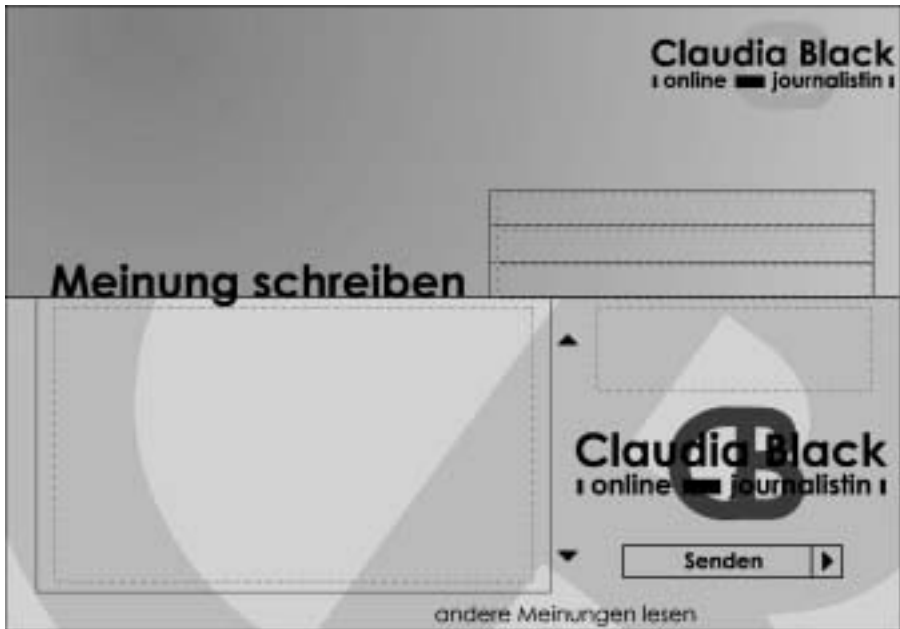


Abbildung 6.6: Flash-Interface zum Eintragen der Meinungen

Beginnen Sie, indem Sie wieder in jeder Ebene ein neues Schlüsselbild anlegen. Nennen Sie danach eines „Schreiben“. Die Textfelder müssen wieder benannt werden. Unten links wählen Sie die Bezeichnung „text“, die drei in der Mitte rechts von oben nach unten „name“, „web“ und „email“. Beachten Sie, dass die Textfelder als Input-Textfelder bestimmt werden und nicht mehr dynamisch sind!

Weisen Sie diesem Schlüsselbild nun folgendes ActionScript zu:

```
name = "Dein Name";
email = "Deine E-Mail";
web = "http://";
text = "Hier dein Kommentar";
stop();
```

■ ■ ■ *Listing 6.17:*
Standardvorgaben für
das Inputinterface

Damit wird den Input-Textfeldern ein Anfangswert gegeben. Nun schreiben Sie noch für den Senden-Knopf folgendes ActionScript:

```
on (release) {
  if (name eq "" or name eq "Ihr Name") {
    send_status= "Name fehlt !";
  } else if (
    text eq "" or text eq "Ihr Kommentar") {
    send_status = "Kommentar fehlt !";
  } else {
    send_status = "";
    play ();
  }
}
```

■ ■ ■ *Listing 6.18:*
Es wird überprüft, ob
Daten eingegeben wur-
den oder nicht.

Dieses Skript überprüft, ob beim Namen und beim Kommentar wirklich Daten eingegeben wurden. Wenn dem nicht so ist, wird der Variablen `send_status` eine Fehlermeldung übermittelt. Legen Sie ein dynamisches Textfeld mit der Bezeichnung `send_status` in einem freien Raum an.

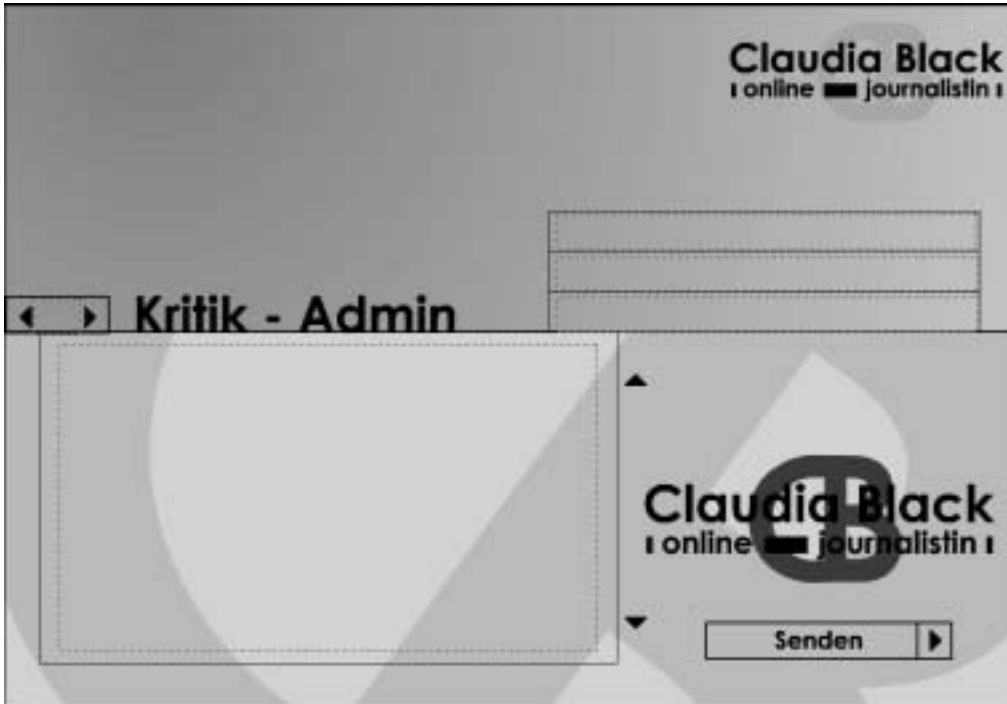
Legen Sie ein weiteres Schlüsselbild mit folgendem ActionScript an:

```
loadVariablesNum ("enter.php", 0, "GET");
gotoandplay(1);
```

■ ■ ■ *Listing 6.19:*
Daten speichern

Die Daten werden nun an `ENTER.PHP` gesendet, danach springt Flash zum Beginn des Films zurück.

Administriermodus



■ Abbildung 6.7: Das Admininterface

In diesem Abschnitt wird eine Schnittstelle für einen Administrator der Datenbank angelegt. Zuerst legen Sie wieder in jede Ebene ein Schlüsselbild. Die Bezeichnungen der Input-Textfelder können übernommen werden. Nennen Sie dieses Schlüsselbild nun „Admin“.

Die Blätterfunktion, Mitte links, ist im Wesentlichen die alte geblieben, nur dass jetzt direkt ohne Zwischenframes geblättert wird. Weisen Sie also dem nach links zeigenden Pfeil folgendes ActionScript zu:

Listing 6.20: ■ ■ ■
Blättern

```

on (press) {
    if (Number(_root.No) != 1) {
        _root.No = _root.No-1;
        loadVariablesNum ("check_id.php", 0, "GET");
    }
}

```

Und dem nach rechts zeigenden Pfeil:

```
on (press) {
    if (Number(_root.no) == Number(_root.nomax)) {
        _root.no = "1";
        loadVariablesNum ("check_id.php", 0, "GET");
    } else {
        _root.no = Number(_root.no)+1;
        loadVariablesNum ("check_id.php", 0, "GET");
    }
}
```

■ ■ Listing 6.21:
Blättern

Abschließend muss noch dem Senden-Knopf ein neues ActionScript hinzugefügt werden:

```
on (release) {
    if (passwort == "azad") {
        loadVariablesNum ("admin.php", 0, "GET");
    } else {
        status = "fehler";
    }
    gotoAndPlay ("admin");
}
```

■ ■ Listing 6.22:
Passwortabfrage und
Eintragen der Daten
mittels admin.php

Die Variablen werden nur an ADMIN.PHP geschickt, wenn der Wert der Variablen passwort „azad“ ist. Hier kann es natürlich auch eine andere Bezeichnung geben, aber diese Sperre soll gewährleisten, dass Unbefugte die Datenbank nicht manipulieren können.

Damit der Administrator jetzt auch noch die Schnittstelle erreichen kann, wechseln Sie zum ersten Schlüsselbild dieser Szene. Erweitern Sie die bisher vorhandene Zeile mit dem nachfolgenden ActionScript:

```
if (status == "admin") {
    gotoAndPlay ("admin");
}
```

Ist der Wert der Variablen status also „Admin“, so springt Flash zum Administriermodus. Ist das passwort dann auch noch „azad“, können Daten gespeichert werden.

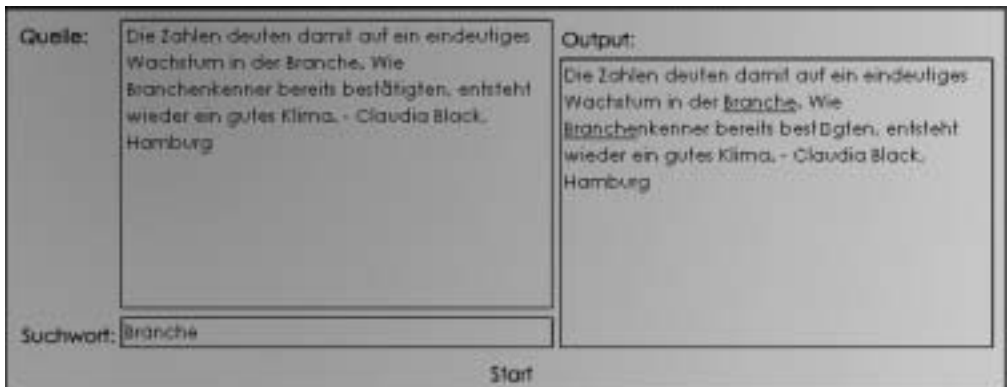


Texte ausleuchten (Flash und PHP)

Azad Adsay

*„Wissend ist, wer weiß, wo er findet, was
er noch nicht weiß.“*

Georg Simmel



■ ■ ■ **Abbildung 7.1:** Im Output ist das Suchwort (hier Branche) markiert.

PHP ermöglicht es, einzelne Wörter oder Buchstaben aus einem längeren Text herauszufiltern. Die Eingabe und die Ausgabe erfolgen in Flash.

Listing 7.1: ■■■
Suche nach einer Zeichenfolge

```
<?
$text = str_replace($search, "<u>$search</u>", $quelle);
print "&text=" . urlencode($text);
?>
```

Mit dem PHP-Befehl `str_replace()` wird nach dem Wert einer Variablen, hier `search`, gesucht. Wird eine Zeichenfolge in der Variablen `quelle` gefunden, wird diese unterstrichen ausgegeben.

Für die Eingabe in Flash brauchen Sie zwei Input-Textfelder und ein dynamisches Textfeld. Ein Input-Textfeld benennen Sie „quelle“, das andere „search“. In „quelle“ wird der zu suchende Text eingegeben. Das dynamische Textfeld soll „text“ heißen, es gibt den Text mit den unterstrichenen Werten aus.

Für den Suchknopf benötigen Sie das folgende kleine Skript zum Einladen der Variablen:

Listing 7.2: ■■■
search4.php laden

```
on (release) {
    text = "Es wird geladen!";
    loadVariables("search4.php", this, "POST");
}
```

Während das PHP-Skript noch den Text durchsucht, wird die Nachricht angezeigt, dass noch geladen wird.



Bilder dynamisch mit PHP in Flash laden

Azad Adsay

„Bildung kommt von Bildschirm und nicht von Buch, sonst hieße es ja Buchung“.

Dieter Hildebrandt

In diesem Kapitel werden wir JPG-Bilder in Flash einladen. Wie gesagt, mit Flash MX ist das Laden von Bildern keine Schwierigkeit. Jedoch ist dies auch schon mit älteren Flash-Versionen möglich. Hier ist es aber bedeutend schwieriger. Mit einem PHP-Skript kann aber eine JPG-Datei in eine SWF-Datei konvertiert werden, welche dann in Flash mit `loadMovie` eingeladen wird. Diese Prozedur benötigt natürlich mehr Ressourcen auf dem Server, aber es ist eine, wenn nicht die beste Alternative zu Flash MX.

Ich werde hier das gesamte Skript aufführen, jedoch nur einzelne Zeilen analysieren und erklären.

Für dieses Kapitel benutzen wir den JPG>>SWF Converter von Norman Liebich (norm@studioadhoc.de).

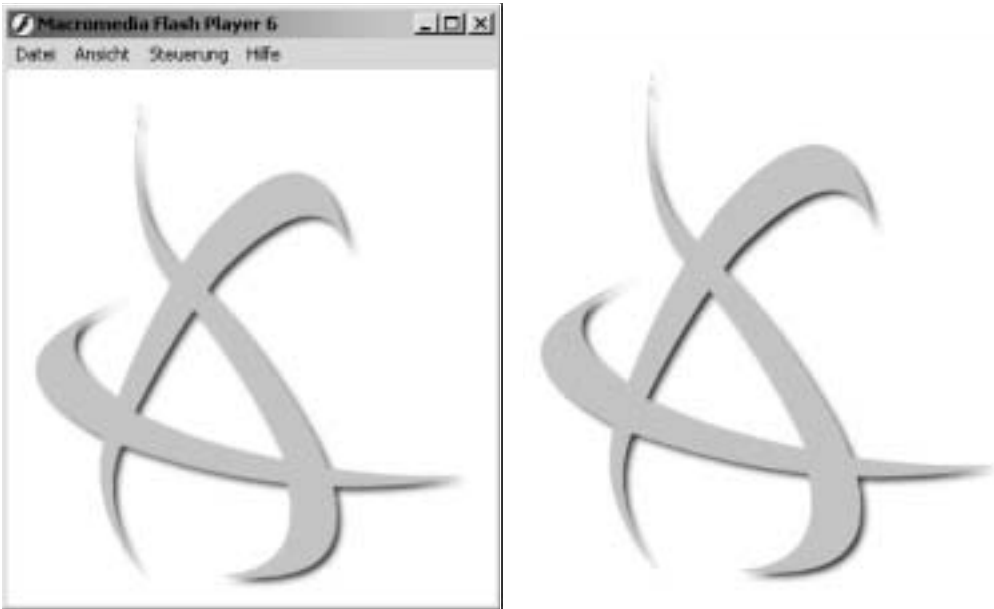
Das Skript konvertiert eine JPG-Datei in eine SWF-Datei. Die Namen werden per Variable übergeben:

```
http://localhost/swf/convert.php?pic=
azadadsay_com.jpg&swf=azadadsay_com.swf
```

Die Variable `pic` beinhaltet das Bild, das konvertiert werden soll, und die Variable `SWF` den Namen der SWF-Datei. Die Variable `SWF` kann den Ordner, in dem das SWF erstellt werden soll, beinhalten. Wird nur ein Dateiname eingegeben, wird die SWF-Datei in dem Ordner erstellt, in dem sich die PHP-Datei befindet.

Ich habe ein JPG umgewandelt, das 9,41 Kbyte groß war. Das SWF war lediglich um 0,09 Kbyte größer, also auf 9,50 Kbyte gewachsen. Und die Qualität war zudem außerordentlich gut.

■ ■ Listing 8.1:
Aufrufen des
`convert.php`-Skripts



— Abbildung 8.1: Links: die generierte SWF-Datei, rechts: das Original-JPG-Bild

Ohne den Rahmen des Flash 6 Player wäre es kaum möglich zu unterscheiden, welche Datei das SWF und welche Datei das JPG ist.

Beginnen wir mit der Datei SHAPERECORD.PHP, welche später in die Datei CONVERT.PHP eingebunden wird. Die Datei CONVERT.PHP wandelt schließlich die Datei um.

Listing 8.2: — shapeRecord.php

```
<?php
function convertInt($int) {
    $hexString=dechex($int);
    $null="";
    $length=strlen($hexString);

    for ($z=0;$z<8-$length;$z++) {
        $null.="0";
    }

    $hexString=$null.$hexString;
    $bigInt="";

    for ($z=3;$z>=0;$z--) {
        $temp=hexdec(substr($hexString,$z*2,2));
        $bigInt.=chr($temp);
    }

    return $bigInt;
}
```

```

function maxBit($var) {
    $curBit=1;

    while ($var >= pow(2,$curBit)) {
        $curBit++;
    }

    return $curBit;
}

function giveZero($var,$nBites) {
    $output="";
    $anzahl=$nBites-maxBit($var);
    for ($z=0;$z<$anzahl;$z++) {
        $output.="0";
    }
    return $output;
}

function doRect($minX,$minY,$width,$height) {

    $nBits=1;

    if (maxBit($minX)>$nBits) {
        $nBits=maxBit($minX);
    }

    if (maxBit($minY)>$nBits) {
        $nBits=maxBit($minY);
    }

    if (maxBit($width)>$nBits) {
        $nBits=maxBit($width);
    }

    if (maxBit($height)>$nBits) {
        $nBits=maxBit($height);
    }

    $nBits++;
    $bnBits=giveZero($nBits,5).base_convert($nBits,10,2);
    $bminX=giveZero($minX,$nBits).base_convert($minX,10,2);
    $bwidth=giveZero($width,$nBits).base_convert($width,10,2);
    $bminY=giveZero($minY,$nBits).base_convert($minY,10,2);
    $bheight=giveZero($height,$nBits).base_convert($height,10,2);
    $bigOne=$bnBits.$bminX.$bwidth.$bminY.$bheight;
    $bytes=round(strlen($bigOne)/8);
}

```

```

        if ($bytes<strlen($bigOne)/8) {
            $bytes++;
        }
        $rect=array();

        for ($z=0;$z<$bytes;$z++) {
            $rect[$z]=base_convert(substr($bigOne,$z*8,8),2,10);
        }

        return $rect;
    }

function edgeRecord($var,$direction) {

    if ($var < 0) {
        $bin="0".base_convert($var,10,2);
        $bin=binInvert($bin);
    }
    else {
        $bin="0".base_convert($var,10,2);
    }
    $nBits=strlen($bin)-2;
    $nBits=giveZero($nBits,4).base_convert($nBits,10,2);
    $output="11".$nBits."0".$direction.$bin;
    return $output;
}

function shapeRecord($width,$height) {
    $binString="0001001";
    $width*=20;
    $height*=20;
    /*    $nBits=1;

        if (maxBit($width) > $nBits) {
            $nBits=maxBit($width);
        }

        if (maxBit($height) > $nBits) {
            $nBits=maxBit($height);
        }

        $nBits++;
    */
    $binString.=edgeRecord($width,"0").edgeRecord($height,"1").edgeRecord($width*-1,"0").edgeRecord($height*-1,"1");
    $count=8-(strlen($binString) % 8);
    if ($count < 8) {
        for ($z=0; $z<$count;$z++) {

```

```

        $binString.="0";
    }
}
$bytes=strlen($binString)/8;
if ($bytes<strlen($binString)/8) {
    $bytes++;
}
$shape=array();
for ($z=0;$z<$bytes;$z++) {
    $shape[$z]=base_convert(substr($binString,$z*8,8),2,10);
}
return $shape;
}

function binInvert($var) {
    for ($z=strlen($var)-1;$z>=0;$z--) {
        if (substr($var,$z,1)=="1") {
            break;
        }
    }

    $binInv="";
    for ($zz=0;$zz<$z;$zz++) {
        if (substr($var,$zz,1)=="1") {
            $binInv.="0";
        }
        else {
            $binInv.="1";
        }
    }

    if ($z<strlen($var-1)) {
        $binInv.=substr($var,$z);
    }
    return $binInv;
}

?>

```

Beginnen wir nun mit der Analyse von Teilen des Quellcodes:

```

function convertInt($int) {
    $hexString=dechex($int);
    $null="";
    $length=strlen($hexString);
    for ($z=0;$z<8-$length;$z++) {
        $null.="0";
    }
    $hexString=$null.$hexString;
    $bigInt="";
    for ($z=3;$z>=0;$z--) {

```

■ ■ ■ Listing 8.3:
Funktion *convertInt*

```

        $temp=hexdec(substr($hexString,$z*2,2));
        $bigInt.=chr($temp);
    }
    return $bigInt;
}

```

Wir legen die Funktion `convertInt()`; an. Als Erstes wird die Variable `$hexstring` gesetzt, dazu wird die Funktion `dechex()`; für die Variable `$int` eingesetzt. Diese Funktion konvertiert eine Dezimalzahl in eine Hexadezimalzahl.

```
$length=strlen($hexString);
```

Die Funktion `strlen()`; überprüft das Vorkommen einer Zeichenkette. Wird nichts gefunden, gibt die Funktion `0` aus, andernfalls wird die Anzahl der Buchstaben in der Zeichenkette ausgegeben. Wäre der Wert der Variablen „`hexString`“, also z.B. „Azad Adsay“ (bzw. `Azad%20Adsay`), wäre die Ausgabe `10`.

```

    for ($z=0;$z<(8-$length);$z++) {
        $null.="0";
    }
    $hexString=$null.$hexString;

```

Diese For-Schleife überprüft, ob der Wert der Variablen `$length` größer ist als acht Zeichen. Wenn nicht, werden so viele Nullen vor den Wert von `$hexString` gesetzt, dass die Länge `8` ergibt. Wäre also der Wert der Variablen „Azad“, würde durch diese Funktion der Wert in „0000Azad“ umgewandelt.

Die Datei zur abschließenden Umwandlung des gesamt Skriptes ist die Datei `CONVERT.PHP`:

Listing 8.4: ■■■
convert.php

```

<?php
include "shapeRecord.php";

if (!isset($pic)) {
    die("Usage: convert.php?pic=<picFile>.jpg[&swf=<swf-
File.swf>]<br>
    If the swf-option isn't set, the Skript will return the generated
    swf-code, <br>otherwise the output goes into the file named under
    swf=...<br><br>
    (c)2001, norm@studioadhoc.de");
}

$picFile=@fopen($pic,"r");
if (!$picFile) {
    die("stat=".$pic." wurde nicht gefunden!");
}

$picFileSize=filesize($pic);
$imgSize=getimagesize($pic);

```

```

$head="FWS".chr(5);
$picSize=convertInt($picFileSize+6);
$framedata=doRect(0,0,$imgSize[0]*20,$imgSize[1]*20);
$bytes1=sizeof($framedata);
$rest="";

for ($z=0;$z<$bytes1;$z++) {
    $rest.=chr($framedata[$z]);
}

$shape=shapeRecord($imgSize[0],$imgSize[1]);
$bytes=sizeof($shape);
$shapeRecord="";

for ($z=0;$z<$bytes;$z++) {
    $shapeRecord.=chr($shape[$z]);
}

$rest.=chr(0).chr(12).chr(1).chr(0).chr(67).chr(2).chr(255).
    chr(255).chr(255).chr(127).chr(5).$picSize.chr(1).
    chr(0).chr(255).chr(217).chr(255).chr(216);
$eof="";

$eofArray1= array ("BF","00");

$eofArray2=array("01","41","01","00","d9","40","00","05",
    "00","00","00","00","10");

$eofArray3=array("00","86","06","06","01","00","02","00",
    "00","40","00","00","00");

$max=count($eofArray1);
for ($z=0;$z<$max;$z++) {
    $tmp=hexdec($eofArray1[$z]);
    $eof.=chr($tmp);
}

$length=sizeof($eofArray2)+sizeof($framedata)+sizeof($shape)+3;
$eof.=chr($length).chr(0).chr(0).chr(0).chr(2).chr(0);
$bytes1=sizeof($framedata);
for ($z=0;$z<$bytes1;$z++) {
    $eof.=chr($framedata[$z]);
}

$max=count($eofArray2);
for ($z=0;$z<$max;$z++) {
    $tmp=hexdec($eofArray2[$z]);
    $eof.=chr($tmp);
}

```

```

$eof.=$shapeRecord;
$max=count($eofArray3);
for ($z=0;$z<$max;$z++) {
    $tmp=hexdec($eofArray3[$z]);
    $eof.=chr($tmp);
}

$picData=fread($picFile,$picFileSize);
fclose($picFile);
$size=$picFileSize+54+9+$bytes+sizeof($framedata)*2;
$filesize=convertInt($size);

if (isset($swf)) {
    $swfFile=fopen($swf,"w");
    fputs($swfFile,$head);
    fputs($swfFile,$filesize);
    fputs($swfFile,$rest);
    fputs($swfFile,$picData);
    fputs($swfFile,$eof);
    fclose($swfFile);
} else {
    header("Content-Type: application/x-shockwave-Flash");
    echo $head.$filesize.$rest.$picData.$eof;
}

?>

if (!isset($pic)) {
    die("Useage: convert.php?pic=<picFile>.jpg[&swf=<swf-
File.swf>]<br>
If the swf-option isn't set, the Skript will return the generated
swf-code, <br>otherwise the output goes into the file named under
swf=...<br><br>
(c)2001, norm@studioadhoc.de");
}

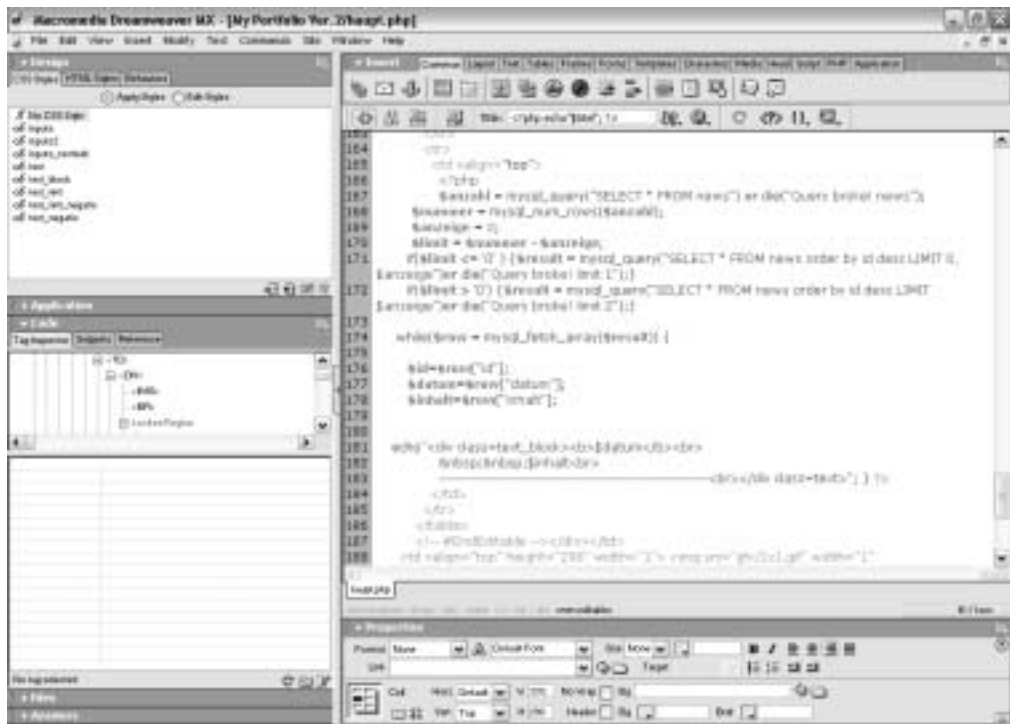
```

Dieses kleine Skript überprüft, ob die Variable \$pic einen Wert hat. Wenn dem nicht so ist, gibt das Skript eine kleine Anleitung des Programmierers aus.

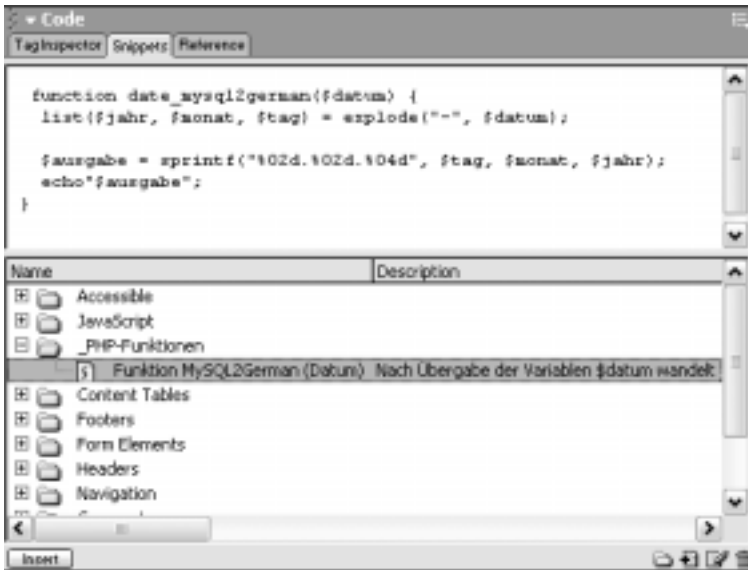


Ausblick Dreamweaver MX

Jürgen Lange



■ ■ ■ Abbildung 9.1: Der neue Dreamweaver MX



■ ■ ■ *Abbildung 9.3: Die neuen Code-Snippets in Dreamweaver*

Die neuen Funktionen umfassen außerdem einen verbesserten FTP-Klienten und die Möglichkeit (und das ist doch mal was wirklich Tolles), Code-Snippets in verschiedenen Ordnern zu speichern und per Drag&Drop in die Projekte einzufügen. Darüber hinaus gibt es bereits eine Auswahl an HTML, JavaScript und CSS-Snippets, die die alltägliche Arbeit von vornherein erleichtern werden. Wirklich nützlich wird das Feature allerdings erst bei einer umfangreichen Sammlung an Code-Snippets.

Zum Thema Designfeatures kann ich auf den ersten Blick nur den vollständig überarbeiteten CSS-Inspektor, der angeblich uneingeschränkt CSS-2-kompatibel sein soll, als wirkliche Neuerung entdecken. Die Designansichten lassen sich als „good-as-ever“ bezeichnen. Auch bezüglich der Layoutansicht sind auf den ersten Blick keine größeren Veränderungen erkennbar.



■ ■ ■ *Abbildung 9.4: Insert-Panel*

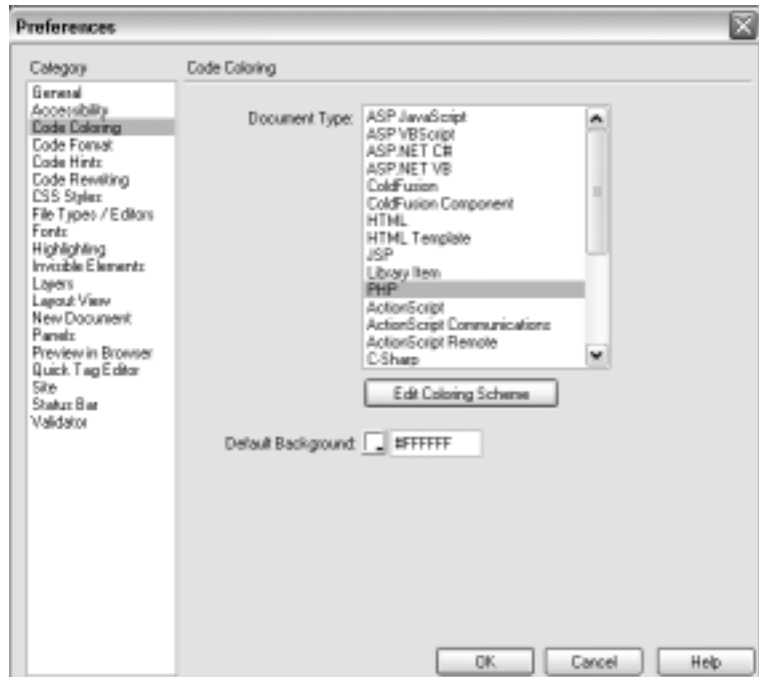
Nett ist das erweiterte Insert-Panel, welches nun, ähnlich wie die Delphi-Entwicklungsumgebung, eine Auswahl von Optionen und Aktionen bereithält. So kann man beispielsweise ein für mich bisher völlig unbekanntes, aber durchaus brauchbares „Fieldset“ (siehe INSERT > FORMS) in seine Projekte integrieren.

Abschließend möchte ich noch die neue Anordnung der Arbeitsumgebung loben, die meiner Meinung nach die Produktivität durch schnelleres Auffinden der gewünschten Funktionen enorm steigern kann. Für alle Gewohnheitstiere lässt sich das neue Outfit natürlich in den Eigenschaften wieder abstellen.

Zusammenfassend kann ich den neuen Dreamweaver, als Bestandteil der Macromedia-MX-Produktreihe, in höchsten Tönen loben. So viele durchdachte Neuerungen und Features findet man selten in einer neuen Version. Die Version MX wird meiner Meinung nach die Pole Position des Dreamweaver unter den WYSIWYG-Editoren regelrecht betonieren, bis jetzt gibt es keinen besseren.

9.2 Workshop: Dreamweaver richtig konfigurieren

Abbildung 9.5: ■■■
Eigenschaften in
Dreamweaver MX



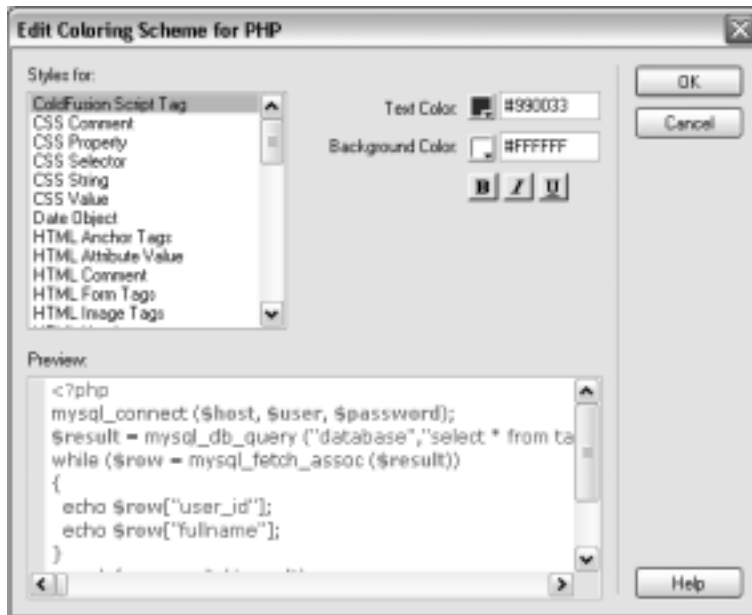
HTML-Seiten zu erstellen ist keine wirkliche Hürde für (angehende) professionelle Webdesigner/Webentwickler, zumal Dreamweaver es einfach wie nie zuvor macht. Deshalb möchte ich Ihnen auch nicht erklären, wie man schicke Formulare anfertigt, sondern wie Sie Ihren Dreamweaver richtig konfigurieren. Welche Veränderungen bieten sich bei den Standardeinstellungen an und was muss ich richtig einstellen, damit meine Seite perfekt konfiguriert ist? Das sollen die Leitthemen des nächsten Abschnitts sein.

Standard ade

Die Standardeinstellungen sind meistens die günstigsten für die breite Masse. Es gibt allerdings immer wieder kleine nützliche Änderungen, die Ihnen als individuellen User von Dreamweaver die Arbeit sehr erleichtern können.

Zuallererst wären da beispielweise die verschiedenen Arbeitsumgebungen. Arbeiten Sie am häufigsten als Coder, empfehle ich Ihnen die „Homepage-like“-Coder-Ansicht. Sie bietet eine für Codehacker übersichtliche Basis für schnelles Arbeiten. Sie haben alle codespezifischen Möglichkeiten direkt im Blick oder über maximal einen Klick erreichbar.

Für Gewohnheitstiere gibt es, wie schon gesagt, auch die Möglichkeit, die alte Dreamweaver-4-Ansicht zu behalten. Sie hat sicherlich auch ihre Vorteile, ist aber meiner Meinung nach etwas nervig in Bezug auf das Verschieben des Eigenschaften-Fensters oder der frei schwebenden Verhaltens- und CSS-Panels. Aber wer es so mag ... warum nicht?

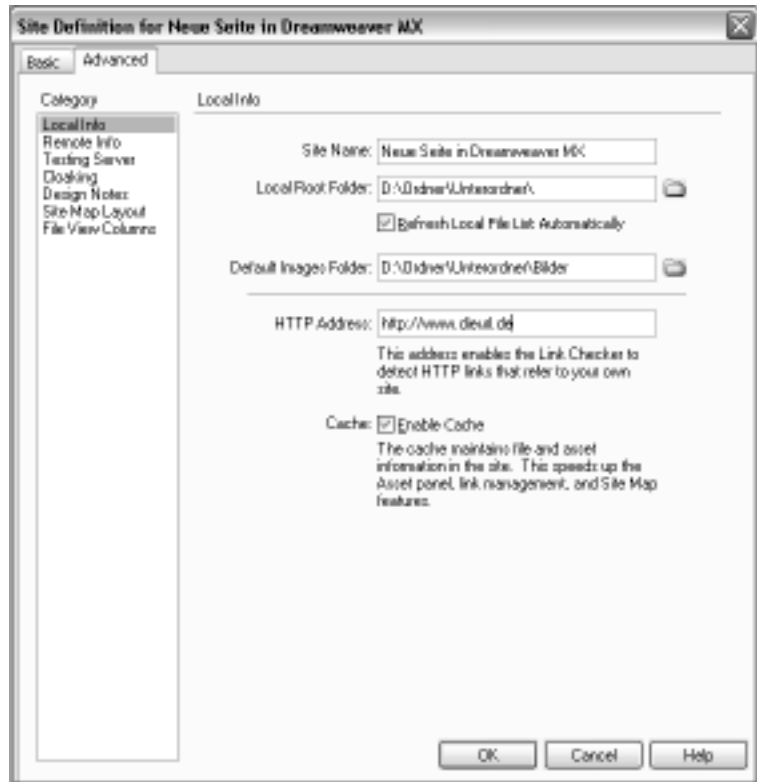


■ ■ ■ **Abbildung 9.6: Einstellung der Syntax-Hervorhebung**

Die nächste Standardeinstellung, die sich für eine Änderung anbietet, ist meiner Meinung nach die DEFAULT DOCUMENT-Einstellung im *New-Document*-Abschnitt des Eigenschaften-Fensters. Hier kann man konfigurieren, welche Dateierweiterung den Projekten standardmäßig angehängt werden soll. Beispielsweise eignet sich für PHP- oder ASP-Programmierer die Standardeinstellung HTML natürlich nur begrenzt. Daher können diese die Einstellung logischerweise auf `.php` bzw. `.asp` setzen.

Es lohnt sich also auf alle Fälle, jede einzelne Option in den Einstellungen genau zu betrachten und den eigenen Wünschen anzupassen.

Site-Konfiguration

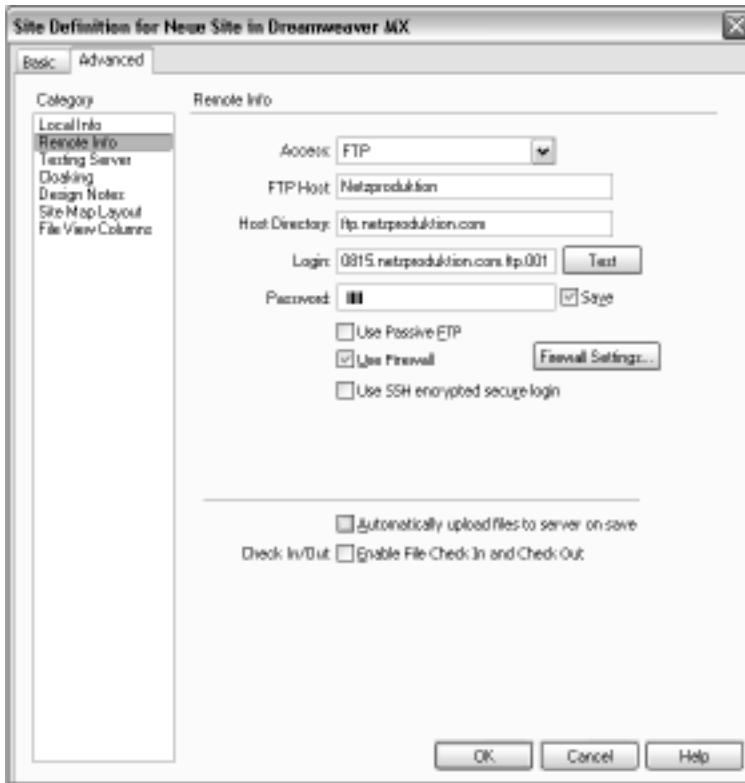


■ ■ ■ *Abbildung 9.7: Site-Definition*

Viele Dreamweaver-Anwender nutzen die Site-Funktion nicht, weil sie ihnen entweder vom Sinn her nicht bekannt ist oder zu umständlich wird, wenn man mehrere Projekte gleichzeitig entwickelt. In Dreamweaver MX gewinnt die Site-Funktion noch mehr an Bedeutung, weshalb es sich durchaus lohnt, sich damit zu beschäftigen.

Sinnvoll ist die Funktion, um die Dateien eines Projekts zu verwalten, FTP-Uploads zu koordinieren und neuerdings auch bei der Erstellung dynamischer Websites.

Wählen Sie im Menü **SITE > NEW SITE** bzw. **SITE > NEUE SITE...** (alle Befehle basieren auf der Dreamweaver-MX-Beta-Version, englisch). Sie gelangen in das Site-Definition-Fenster, wo Sie Namen, Ordner, ImageFolder und die URL des Projekts eingeben können. Unter **REMOTE INFO** können Sie u.a. Zugangsart und -daten für Ihr System eingeben (z.B. FTP-Zugang).



■ ■ ■ *Abbildung 9.8: FTP konfigurieren*

Auch Firewall-Einstellungen lassen sich hier beispielsweise für FTP-Uploads vornehmen.

Die Einstellungen für den Test-Server finden Sie unter „Testing Server“, wo Sie neben der Art des Servers (beispielsweise APS.Net, ColdFusion, JSP und PHP/MySQL) auch die Zugangsart (lokal oder direkt auf dem Online-Server) auswählen können. Dies ist wichtig für das Testen der Projektdateien während der Entwicklungsphase (oberste Regel: testen, testen, testen!).

Alle weiteren Angaben sind optional und für einfache Site-Verwaltungen in den Standardeinstellungen optimal konfiguriert.

Abschließend bleibt zu sagen, dass ein Tool wie Dreamweaver sehr viele versteckte Funktionen enthält, die es sich wirklich zu erkunden lohnt.



Die CD zum Buch

Die CD-ROM zum Buch enthält alle Beispieldateien, geordnet nach Kapitelnummern und -bezeichnungen.

Desweiteren haben Sie Zugriff auf Software, die für die Arbeit mit dem Buch nützlich ist:

- die aktuelle PHP-Distribution
- den frei verfügbaren Apache Webserver
- Testversionen von Flash MX und Dreamweaver MX

Zusätzlich finden Sie auf der CD eine Reihe von Interviews, die wir mit Designern und Gestaltern aus der Branche geführt haben. Namentlich sind dies:

- Gertolf (Gert Rudolf Köhler)
- Daniel Glass (glasskom)
- Kai Baumgartner
- Michael Bundscherer
- Dominik Schech (www.schech.net)

Wir möchten uns an dieser Stelle noch einmal herzlich bei unseren Interviewpartnern bedanken.

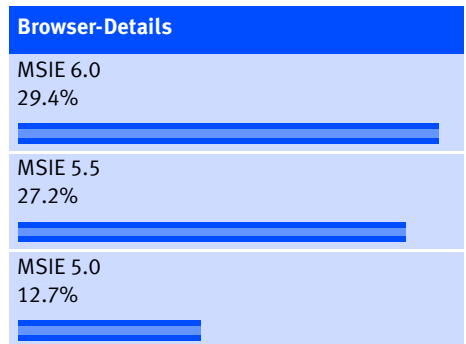
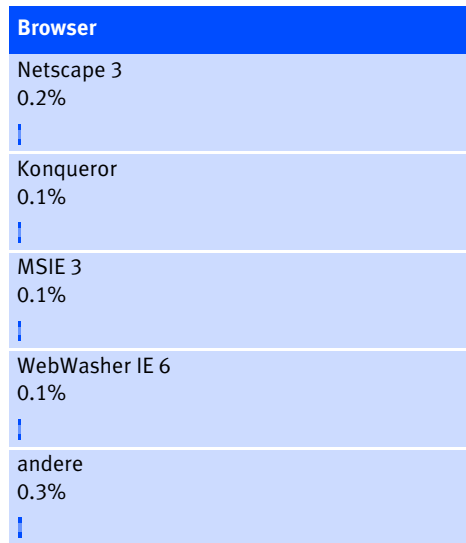
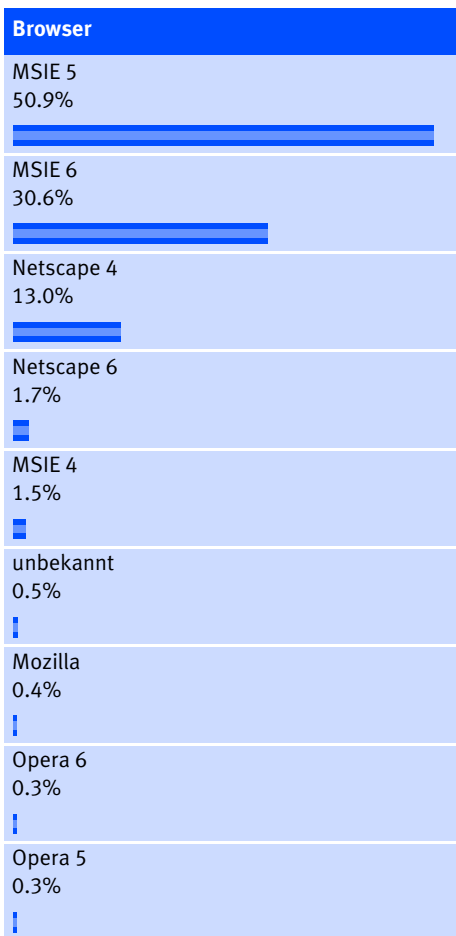
Sollten Sie Fragen zu den Daten auf der CD-ROM haben, wenden Sie sich bitte an buch@bookworker.de



Anhang

Statistiken*

Azad Adsay



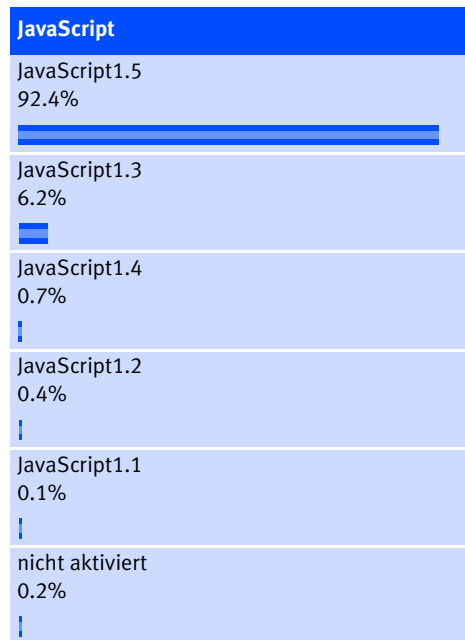
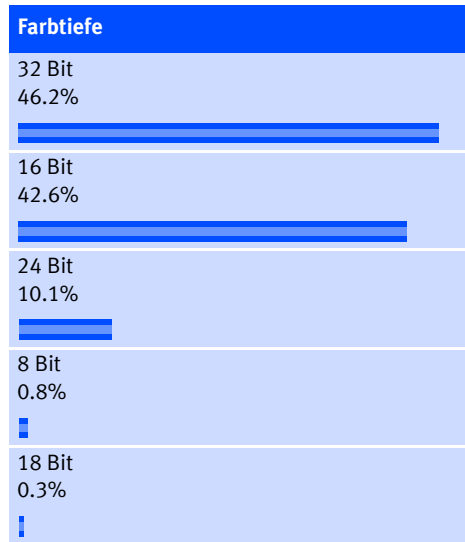
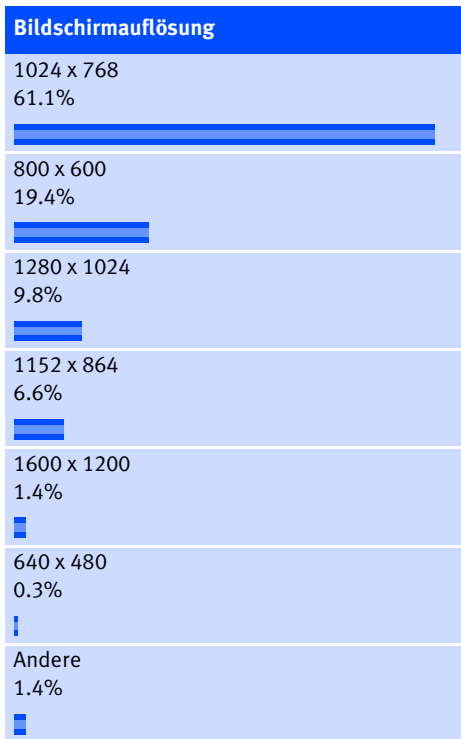
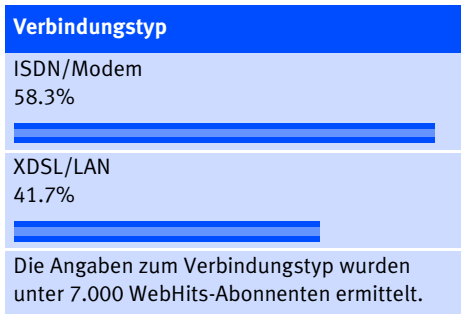
*) Die Statistiken werden mit freundlicher Genehmigung von WebHits (www.webhits.de) abgebildet.

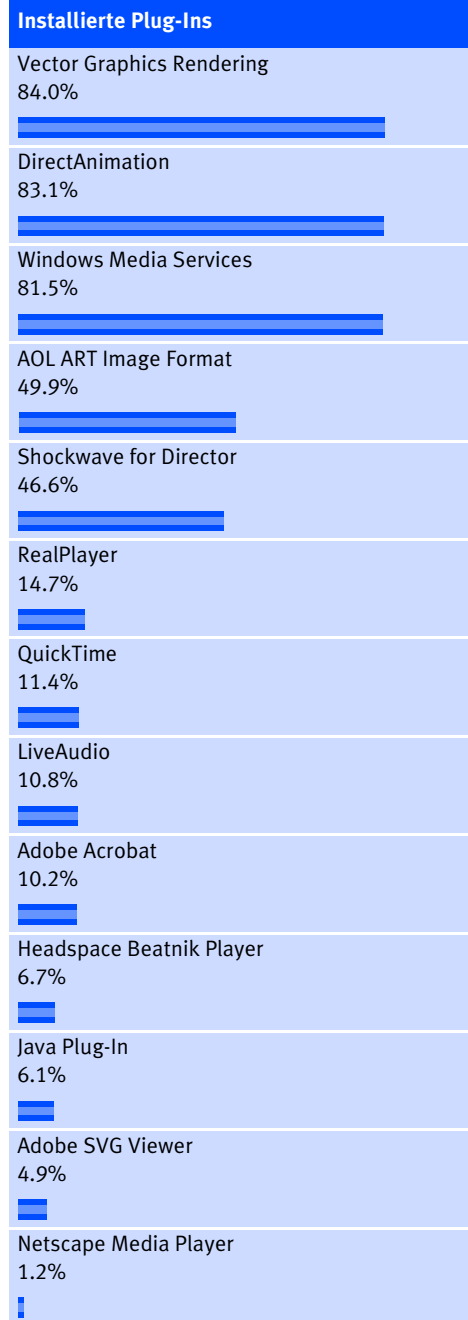
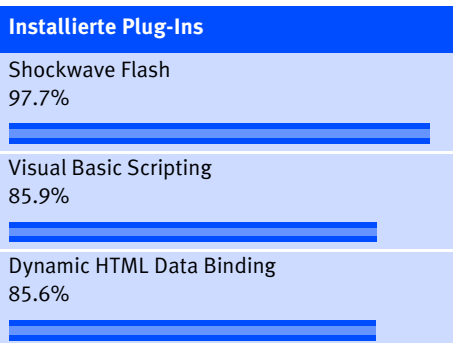
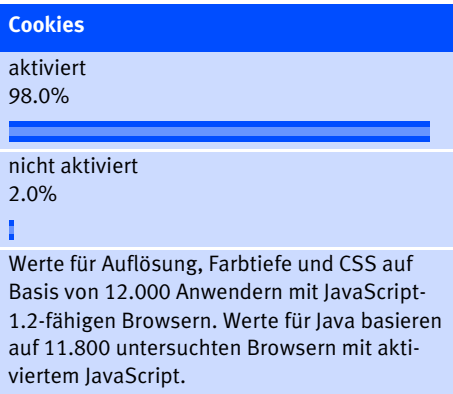
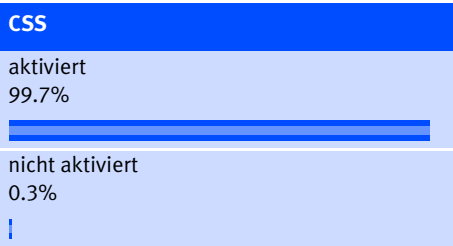
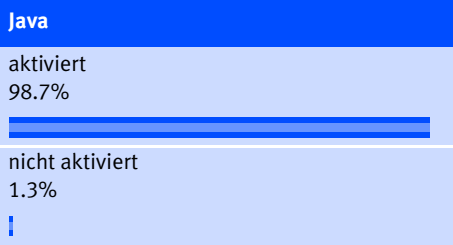
Browser-Details	
MSIE 5.01	10.3%
Netscape 4.7	2.6%
Netscape 4.75	2.0%
Netscape 4.73	1.9%
Netscape 4.5	1.5%
MSIE 4.01	1.4%
MSIE 6.0b	1.2%
Netscape 6.2	1.1%
Netscape 4.51	1.1%
Netscape 4.78	0.9%
Netscape 4.6	0.6%
Netscape 6.1	0.5%
unbekannt	0.5%
Netscape 4.05	0.4%

Browser-Details	
Mozilla	0.4%
Netscape 4.77	0.3%
Netscape 4.76	0.3%
MSIE 5.12	0.3%
Netscape 4.79	0.2%
Netscape 4.08	0.2%
MSIE 5.13	0.2%
Netscape 4.72	0.2%
Opera 6.0	0.2%
Netscape 4.61	0.2%
Opera 5.12	0.2%
Netscape 3.01	0.2%
Netscape 4.07	0.2%
MSIE 5.14	0.2%

Browser-Details	
Opera 6.01	0.1%
Netscape 4.06	0.1%
Konqueror	0.1%
Netscape 6.0	0.1%
Netscape 4.04	0.1%
WebWasher IE 6.0	0.1%
Netscape 4.03	0.1%
MSIE 4.0	0.1%
Netscape 4.01	0.1%
andere	0.5%

System-Details	
Win 98	38.51%
Win 2000	21.31%
Win ME	10.64%
Win XP	9.55%
Win NT	8.30%
Win 95	6.58%
Mac OS	3.31%
X11	0.80%
Win 3.x	0.10%
OS/2	0.09%
AmigaOS	0.06%
andere/unbekannt	0.75%
Diese Daten wurden unter 13.500 WebHits-Abonnenten ermittelt.	





Weblinks

Azad Adsay und Jürgen Lange

Website zum Buch

<http://buch.bookworker.de>

Unsere Seiten

<http://www.azadadsay.com>
<http://www.smokey.de>
<http://www.netzproduktion.com>
<http://www.bookworker.de>

High-End-Seiten

<http://www.hillmancurtis.com>
<http://www.alcapone.com>
<http://www.wmteam.de>
<http://www.jaypadua.com>
<http://www.derbauer.de>
<http://www.deepfriedlive.com>
<http://www.madrhino.com>
<http://www.2advanced.com>
<http://www.highersourcemusic.com>
<http://www.northlight.com>
<http://www.xiteweb.com>
<http://www.linefracture.com>
<http://www.reysantiago.com>
<http://www.turtleshell.com>
<http://www.australianinfront.com.au>
<http://www.mediabasement.com>
<http://www.retroviridae.com>

Ressourcen

► Flash

<http://www.2-serious.de>
<http://www.flash-school.de>
<http://www.flashkit.com>
<http://www.flashstar.de>

► PHP

<http://www.php.net>
<http://www.dynamic-webpages.de>
<http://www.php-homepage.de>
<http://www.php-center.de>

► MySQL

<http://www.mysql.com>

Index

A

addItem 29
 Apache 44
 httpd.conf 46

B

Bilder dynamisch in Flash laden 25

C

Cache 115
 Checkbox 30
 Combobox 29

D

Dateisystem 100
 dechex 258
 duplicateMovieClip 150

E

ECard 188
 PHP, Seite 165
 Eigenschaften der Ebenen 147

F

Filmsequenzen
 _parent 213
 _root 213
 duplicateMovieClip() 213
 onClipEvent () 211
 Flash 5 19
 Flash MX 17
 Flash MX Interface 20
 Flash-Filme in Flash laden 25
 FTP 109
 Führungsebene 148

G

getBytesLoaded() 168
 getLabel() 27
 getSelectedItem() 29
 getState() 31
 Guter Code 51
 Eindeutige Variablennamen 52
 Einrückungen 51
 Kommentare 51

J

JavaScript 194

L

Layer Properties 147
 length 149
 Listbox 29
 LoadMovie () 25
 loadMoviesNum() 25

M

Maus
 _xmouse 211
 _ymouse 211
 MySQL 82
 Datenbank anlegen 87
 Datenbank füllen 89
 Datenbankverbindung 85
 Datensätze auslesen 90
 Datensätze erzeugen 92
 Installation 48
 MySQL und PHP 85
 mysql_db_query() 99

P

PHP

Arrays 58

- Arrays leeren 61
- assoziative 59
- indizierte 58
- mehrdimensionale 60

Boolesche Werte 64

Datumsprobleme 78

Einfacher Passwortschutz mit PHP 202

Einführung 44

Fallunterscheidungen 66

- If-Else 66
- switch() 68

FAQ 109

FTP 109

Funktionen 73

- Beispielfunktionen 77
- chdir() 108
- chgrp() 107
- chmod() 107
- chown() 107
- closedir() 108
- copy() 107
- date() 78, 191
- echo() 57
- eigene Funktionen schreiben 73
- fclose() 102
- feof() 104, 106
- fgets() 104, 106
- fgetscsv() 106
- fgetss() 106
- filegroup() 107
- fileowner() 107
- fileperms() 107
- fopen() 102
- fputs() 106
- fread() 106
- fseek() 106
- ftell() 106
- fwrite() 104, 106
- include() 86
- is_dir() 108
- mkdir() 108
- opendir() 108
- Parameter 75
- readdir() 108
- rename() 107
- require() 86
- return() 74

rewinddir() 108

rmdir() 108

switch() 68

touch() 101

unlink() 101

Installation 44

Kommentare 51

Konfiguration 46

Konstanten 62

Mailversand mit PHP 201

Operatoren 62

Arithmetische Operatoren 63

Logische Operatoren 64

Stringoperatoren 64

Vergleichsoperatoren 65

Zuweisungsoperatoren 62

PHP und HTML 53

php.ini 47

Schleifen 70

Do-While-Schleife 72

For-Schleife 70

While-Schleife 71

str_replace 252

Variablen 56

Variablenübergabe an Flash per /-Tag 205

PHPmyAdmin 45

PHPTriad 44, 45

Konfiguration 46

R

Radiobox 31

random 118

rawurlencode() 113

S

Scrollbalken 33

setProperty 150

SQL 82

AND 84

BETWEEN 84

DELETE 84

DISTINCT 84

INSERT 98

LIKE 85

LIMIT 84

NOT 84

OR 84

ORDER BY 84

SELECT 82

UPDATE 84

WHERE 83

T

- Textdateien dynamisch in Flash laden 26
- Textfelder
 - Dynamische Textfelder 22
 - Input-Textfelder 22
 - Statistische Textfelder 22
- Tintenfasswerkzeug 24, 154

W

- Werkzeuge
 - Ellipsenwerkzeug 24
 - Füllungstransformation 24
 - Lassowerkzeug 21
 - Linienwerkzeug 21
 - Mauszeiger 20
 - Pinsel
 - Stiftwerkzeug 24
 - Pipette 24
 - Skalierwerkzeug 24
 - Textwerkzeug
 - Textfeld 22
- While 149



Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt.

Dieses eBook stellen wir lediglich als **Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschliesslich der Reproduktion, der Weitergabe, des Weitervertriebs, der Platzierung im Internet, in Intranets, in Extranets anderen Websites, der Veränderung, des Weiterverkaufs und der Veröffentlichung bedarf der schriftlichen Genehmigung des Verlags.

Bei Fragen zu diesem Thema wenden Sie sich bitte an:

<mailto:info@pearson.de>

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf der Website ist eine freiwillige Leistung des Verlags. Der Rechtsweg ist ausgeschlossen.

Hinweis

Dieses und andere eBooks können Sie rund um die Uhr
und legal auf unserer Website



(<http://www.informit.de>)

herunterladen