

Tobias Hauser
Armin Kappler
Christian Wenz

ActionScript 1 und 2

Flash-Programmierung für Einsteiger



I	Einfacher Einstieg	17
	1 Flash für ActionScript	19
	2 ActionScript in Flash	33
	3 Flash MX 2004 Pro für ActionScript	57
	4 Grundlagen der Programmierung	63
II	Filme steuern	109
	5 Einfache Filmsteuerung	111
	6 Bildschirme und Präsentationen	127
	7 Drucken	139
III	Ein- und Ausgaben	147
	8 Textfelder	149
	9 Tasten	167
	10 Datum und Zeit	187
	11 Rechnen	199
IV	Animieren	215
	12 Farbe wechseln und Drag & Drop	217
	13 Zeichnen und Zufall	229
	14 Von Mäusen und Duplikaten	245
	15 Kollisionstest	259
V	Multimedia	273
	16 Sound	275
	17 Video	291
VI	Externe Filme und Vorladen	309
	18 Externe Filme	311
	19 Vorladen und Player erkennen	321
VI	Formulare, Komponenten und XML	333
	20 Formulare und UI-Komponenten	335
	21 Komponenten anpassen	361
	22 Formulareversenden	371
	23 Warenkorb	381

Auf einen Blick	3
Schritt für Schritt	12
Vorwort	14
Der Weg ist das Ziel	14
Landkarte	14
Wegweiser	15
Sackgasse?	15
Straßenausbesserer	15
Teil I: Einfacher Einstieg	17
1 Flash für ActionScript	19
1.1 Zeitleiste	20
1.2 Symbole	21
1.3 Koordinatensystem und Bühne	24
1.4 Verhalten	27
1.5 Veröffentlichen und Testen	28
1.6 Flash-Versionen	29
Windows und Mac	30
2 ActionScript in Flash	33
2.1 Wo ist ActionScript möglich?	34
Ereignisse für Schaltflächen	36
Ereignisse für MovieClips	36
Ereignisprozeduren	37
Wo befinden Sie sich?	38
2.2 Was lässt sich mit ActionScript steuern?	38
Zugriff auf MovieClips	39
Zugriff auf Schaltflächen	42
2.3 Aktionen-Bedienfeld	43
Aktuelle Aktion	44
Schnellzugriff	45
Skriptfeld	45
Zeile und Spalte	46
Werkzeugleiste mit allen Skriptelementen	46
Skriptobjekt hinzufügen	46
Suchen	46
Ersetzen	47
Zielpfad einfügen	47

	Syntax überprüfen	47
	Auto-Format	47
	Codehinweise zeigen	47
	Referenz	48
	Debug-Optionen	48
	Ansichtsoptionen.....	48
	Menü	48
2.4	Hilfe.....	49
2.5	Einstellungen	49
	Auto-Format-Optionen	49
	Voreinstellungen.....	50
2.6	Fehler finden.....	51
	Testen	52
2.7	trace() und die Ausgabe.....	53
	Debugger	53
2.8	ActionScript-Versionen	54
	Flash-Player-Versionen.....	55
3	Flash MX 2004 Pro für ActionScript	57
3.1	Bildschirme	58
	Folienpräsentation oder Formularanwendung	58
	Hierarchie und Verschachtelung.....	60
3.2	Komponenten.....	60
3.3	Web Services	60
3.4	Entwicklungsmöglichkeiten	61
4	Grundlagen der Programmierung	63
4.1	Variablen.....	64
	Variablenamen	66
	Datentyp	68
	Codehinweise für Datentypen	70
	Datentyp feststellen und wechseln.....	71
	Kurzformen für Variablen.....	73
	Gültigkeit von Variablen	73
4.2	Kommentare	74
4.3	Operatoren	74
	Arithmetische Operatoren	75
	Vergleichsoperatoren	76
	Logische Operatoren	77
4.4	Fallunterscheidungen	79
	if-Fallunterscheidung	79

	switch case-Fallunterscheidung.....	83
4.5	Schleifen	86
	for-Schleife	86
	while-Schleife	87
	do-while-Schleife	88
4.6	Funktionen	89
	Parameter.....	91
	Rückgabewerte.....	91
	Gültigkeitsbereich.....	93
	Rekursive Funktionen	94
	Fertige Funktionen in ActionScript.....	95
4.7	Objekte.....	95
	Klassen und davon abgeleitete Objekte.....	96
	Eigene Klassen und Objekte.....	98
4.8	Strings und ihre Besonderheiten	98
	Sonderzeichen.....	99
	Strings zusammenfügen	100
	Strings vergleichen.....	101
4.9	Arrays	101
	Datentypen und Arrays.....	102
	Mit Arrays arbeiten	103
	Schleifen und Arrays.....	104
	Assoziative Arrays	105
	Multidimensionale Arrays	106
	Teil II: Filme steuern	109
5	Einfache Filmsteuerung	111
5.1	Einfache Navigation	112
	Bildmarkierungen	116
5.2	Szenen	117
5.3	Auswahl treffen.....	119
5.4	MovieClips.....	121
5.5	Ereignisprozeduren	123
6	Bildschirme und Präsentationen	127
6.1	Bildschirme in ActionScript.....	128
	Name und Zugriff auf Folien	128
	Hierarchie und Verschachtelung.....	130
	Verhalten.....	131

Präsentieren in wenigen Schritten	133
6.2 Präsentationen ohne Pro	135
7 Drucken	139
7.1 Drucken auf Knopfdruck	140
Bestimmtes Bild drucken	142
7.2 PrintJob-Klasse	143
Optionen für <code>addPage()</code>	143
Teil III: Ein- und Ausgaben	147
8 Textfelder	149
8.1 Textfelder und ActionScript	150
Dynamische Textfelder	150
Eingabefelder	151
8.2 Mit Textfeld-Inhalten arbeiten	152
8.3 Textfelder formatieren und verändern	155
Inhalt des Textfelds	156
Textfelder = MovieClip?	157
Äußere Erscheinung	157
Text formatieren	157
Style Sheets	160
Text scrollen	161
8.4 Textfelder dynamisch erzeugen	165
9 Tasten	167
9.1 Tastaturereignis <code>keyPress</code>	168
9.2 <code>Key</code> -Objekt	173
Event-Listener	174
ASCII und Co.	174
Das <code>key</code> -Objekt in der Praxis	176
9.3 Spielerische Tasten	178
10 Datum und Zeit	187
10.1 Gegenwart, Vergangenheit und Zukunft	188
Deutsches Datum	189
Eigene Datumswerte	190
10.2 Zeit und Uhr	191
10.3 Countdown	194

11 Rechnen	199
11.1 Einfache Rechenhilfen	200
11.2 Bögen und Trigonometrie	201
Trigonometrie – am Einheitskreis erklärt	201
Trigonometrie in ActionScript	202
11.3 Zufall und Runden	205
Runden und Artverwandtes	206
Beliebige Zufallszahlen	206
 Teil IV: Animieren	 215
12 Farbe wechseln und Drag & Drop	217
12.1 Animation mit ActionScript	218
12.2 Wechselspiele	218
12.3 Drag & Drop	222
12.4 Schieberegler	225
13 Zeichnen und Zufall	229
13.1 Zeichnen	230
13.2 Tiefe	232
13.3 Fortlaufende Animation mit
setInterval() oder onEnterFrame?	233
setInterval()	234
enterFrame	234
13.4 Zufall und Animation	235
13.5 Alles in einem	237
14 Von Mäusen und Duplikaten	245
14.1 Mausbewegungen und mehr	246
Mouse-Klasse	247
Event-Listener	247
Dem Mauszeiger folgen	248
14.2 Duplizieren von MovieClips	250
14.3 Von der Bibliothek zur Bühne	255
15 Kollisionstest	259
15.1 Kollisionen abfangen	260
hitTest() im Test	261
Handgeschriebene Lösungen	262

hitArea – Kollision für Schaltflächen-MovieClips	265
15.2 Aus Kollisionen wird eine Anwendung.....	266

Teil V: Multimedia.....273

16 Sound	275
16.1 Sound in Flash.....	276
Sound-Steuerung mit Verhalten	277
16.2 Sound mit ActionScript	278
Globale Steuerung	278
Sound-Klasse.....	279
Externer Sound.....	283
Wenn der Sound fertig ist	284
16.3 Mehrere Sounds.....	284
Exkurs: Drehregler	288
17 Video	291
17.1 Videos in Flash	292
Video-Steuerung mit Verhalten	293
17.2 Videos mit ActionScript.....	294
Videosteuerung	294
Video aus der Bibliothek	296
Ein Schalter für zwei Anlässe	297
Ende in Sicht?.....	301
17.3 Video mit Sound	301
17.4 Videos dynamisch verändern	303

Teil VI: Externe Filme und Vorladen.....309

18 Externe Filme	311
18.1 Externe Filme laden.....	312
Werte über Stufen austauschen	313
Absolute Adressierung	313
Film entfernen.....	314
18.2 Externe Filme in MovieClips laden.....	314
Externen Film aus MovieClip entfernen.....	316
_lockroot im Hauptfilm	317
Ladevorgang überwachen	317
18.3 Externe Bilder	318

19	Vorladen und Player erkennen	321
19.1	Vorladen	322
	Mit Bildern	323
	Mit Bytes.....	325
	Restzeit berechnen	327
	Preloader mit externer Datei.....	330
19.2	Player-Erkennung.....	331
	 Teil VI: Formulare, Komponenten und XML	 333
20	Formulare und UI-Komponenten	335
20.1	UI-Komponenten	336
	Textfeld(er)	338
	Mehrzeiliges Textfeld	341
	Checkbox	343
	Radiobutton	344
	Auswahlliste	346
	Schaltfläche	350
20.2	Beispielsanwendung mit Flash MX 2004 Professional.....	351
20.3	Beispielsanwendung ohne Flash MX 2004 Professional.....	357
21	Komponenten anpassen	361
21.1	Eigene Radiobuttons	363
21.2	Eigene Checkboxes.....	365
21.3	Zugriffskontrolle.....	366
21.4	Komponenten anpassen	367
22	Formulareversenden	371
22.1	Daten versenden und austauschen	373
22.2	Kontaktformular.....	375
23	Warenkorb	381
23.1	Allgemeines	382
23.2	Artikelseiten.....	383
23.3	Der Warenkorb	387
23.4	Lieferdaten und Bestätigung	391
	 Index	 395

Zusatzkapitel auf CD

Kontextmenü Drucken

Streaming

XML

Grundlagen der Programmierung

- ▶ Variablen, Operatoren, Fallunterscheidungen, Schleifen **65**
- ▶ Arrays und Objekte **90**

Einfache Filmsteuerung

- ▶ Einfache Filmsteuerung **113**
- ▶ Szenenwechsel **118**
- ▶ Ein Bild auswählen **120**
- ▶ Umfärben mit Filmsteuerung **122**

Bildschirme und Präsentationen

- ▶ Verhalten einsetzen **132**
- ▶ Präsentieren mit Folien **133**

Drucken

- ▶ Drucken mit Schaltfläche **141**
- ▶ Schritt für Schritt: Drucken mit dem `PrintJob`-Objekt **144**

Textfelder

- ▶ Textfelder zum Leben erwecken **153**
- ▶ Text beständig scrollen **163**

Tasten

- ▶ Quiz mit Tastaturereignissen **169**
- ▶ Quiz mit `key`-Objekt **176**
- ▶ Kransteuerung **180**
- ▶ Kransteuerung II **182**
- ▶ Kransteuerung III **184**

Datum und Zeit

- ▶ Uhr mit ActionScript **191**
- ▶ Flexibler Countdown **194**

Rechnen

- ▶ Schaukelstuhl mit Pendelbewegung **204**
- ▶ »17 und 4« **208**

Farbe wechseln und Drag & Drop

- ▶ Chamäleon umfärben **220**
- ▶ Drag & Drop der Palette **223**
- ▶ Schieberegler **226**

Zeichnen und Zufall

- ▶ Zufällige Formen animieren **238**

Von Mäusen und Duplikaten

- ▶ Dem Mauszeiger folgen **248**
- ▶ Blumenmeer hinter dem Mauszeiger **251**

Kollisionstest

- ▶ Friseur per Kollisionstest **268**

Sound

- ▶ Einfache Soundsteuerung **281**
- ▶ Ein kleiner Mixer **285**

Video

- ▶ Einfache Videosteuerung **295**
- ▶ Eine Schaltfläche zum Abspielen und Pausieren **298**
- ▶ Sound im Video **302**
- ▶ Videobearbeitung per ActionScript **304**

Externe Filme

- ▶ Mit externen Filmen arbeiten **315**
- ▶ Externe Bilder laden **318**

Vorladen und Player erkennen

- ▶ Preloader mit Bildern **323**
- ▶ Preloader mit Bytes **326**
- ▶ Restzeit berechnen **328**

Formulare und UI-Komponenten

- ▶ Textfelder einsetzen **339**
- ▶ Textfelder einsetzen **341**
- ▶ Mehrzeilige Textfelder einsetzen **342**
- ▶ Mehrzeilige Textfelder einsetzen **343**
- ▶ Radiobuttons einsetzen **345**
- ▶ Einfache Auswahlliste **347**
- ▶ Mehrfach-Auswahlliste **349**
- ▶ Umfrage mit Flash MX Pro 2004 **352**
- ▶ Umfrage mit ohne pro **358**

Komponenten anpassen

- ▶ Radiobuttons erstellen **364**
- ▶ Checkbox erstellen **365**
- ▶ Formularüberprüfung integrieren **366**
- ▶ Stile erzeugen und zuweisen **367**
- ▶ Komponenten-Stil zuweisen **369**

Formulare versenden

- ▶ Ein Kontaktformular erstellen **375**

Warenkorb

- ▶ Navigation der Artikelseiten **384**
- ▶ Farb-Schaltflächen programmieren **385**
- ▶ Layout für Auswahllisten **386**
- ▶ Warenkorb-Funktionalität **387**
- ▶ Lieferdaten aufnehmen **391**
- ▶ Navigation vervollständigen **393**

Der Weg ist das Ziel ...

... oder wie Sie zu einem guten ActionScript-Programmierer werden.

Der Erfolg von Flash ist enorm: Der Flash-Player ist die Browser-Erweiterung mit der größten Verbreitung. Neue Versionen werden jenseits des Marketing-Hypes sehr schnell von den Nutzern übernommen. Auch aus Sicht des Designers weiß Flash zu gefallen: Die optischen Freiheiten sind wesentlich größer als mit HTML, CSS und JavaScript. Kompatibilitätsprobleme beschränken sich auf verschiedene Player-Versionen und die Oberfläche von Flash kommt den Anforderungen eines Gestalters entgegen.

Aber irgendwann gelangt jeder Flasher an einen Punkt, an dem er sich weiter entwickeln möchte. Die rein optischen Hilfsmittel reichen nicht mehr aus, er wirft erste Blicke auf das AKTIONEN-Bedienfeld und überlegt sich, die Programmiersprache von Flash, ActionScript, zu lernen. Wenn Sie an diesem Punkt angelangt sind, ist unser Buch für Sie das richtige!

Landkarte

Dieses Buch zeigt Ihnen, wie Sie mit ActionScript arbeiten und in Flash wichtige Aufgaben erledigen. Die Spanne reicht von einfacher Filmsteuerung über Multimedia bis hin zu Formularen und XML. Eine Liste der wichtigsten Beispiele finden Sie auf der Innenseite des Titelbilds in der Klappe.

AS
1.0

Die Grundlagen zu ActionScript sind kompakt im ersten Teil des Buches dargestellt. Dort erfahren Sie, was Sie aus ActionScript-Sicht über Flash wissen müssen und wie Sie ActionScript einsetzen. Das »dickste« Kapitel ist Kapitel 4, die »Grundlagen der Programmierung«. Hier lernen Sie Syntax und Programmierelemente von Flash kennen.

AS
2.0

In unserem Buch gehen wir auf ActionScript 1 und ActionScript 2 ein. ActionScript 2 ist neu in Flash MX 2004. Allerdings können Sie dort und in allen älteren Flash-Versionen auch mit ActionScript 1 programmieren.

MX

MX
04

MX
04 PRO

Wir berücksichtigen die Flash-Versionen MX 2004, MX 2004 Pro und den Vorgänger Flash MX. Besonderheiten in Flash 5 finden teilweise Erwähnung.



Auf der CD-ROM zum Buch liegen Ihnen alle Beispiele aus dem Buch ohne und mit ActionScript-Code vor. Die Beispiele gibt es für Flash MX 2004 (immer im Ordner FLASHMX2004 zum jeweiligen Kapitel) und für Flash MX (im Ordner FLASHMX zum Kapitel). Die Variante für Flash MX 2004 ist in ActionScript 2 geschrieben, erkennbar an der Dateieindung `_AS2`, die Version für Flash MX dagegen in ActionScript 1 mit der Dateieindung `_AS1`.

Wegweiser

Jeder Lernende hat eigene Anforderungen und möchte seinen Lernweg individuell festlegen. Wir empfehlen Ihnen für dieses Buch einen von drei Wegen:

- ▶ Den langsamen Aufbau: Sie arbeiten das Buch in Ruhe von vorne nach hinten durch. Dabei lernen Sie viel, benötigen aber ein wenig Zeit.
- ▶ Schneller Einstieg: Sie springen direkt ins kalte Wasser. Beginnen Sie mit Kapitel 2, »ActionScript in Flash«, und lernen Sie dann die »Grundlagen der Programmierung« in Kapitel 4 kennen. Schließlich wenden Sie sich den Beispielen zu, die Sie am meisten reizen.
- ▶ Lernen am Beispiel: Sie kennen sich schon etwas aus? Machen Sie sich direkt auf den Weg zu einem Beispiel Ihrer Wahl. Wenn Ihnen Grundlagen unklar sind, springen Sie einfach zurück in den ersten Teil des Buches, um Ihr Wissen ein wenig aufzufrischen.

Sackgasse?

Wenn Sie einmal nicht weiter wissen, einen Fehler bemerken oder irgendein anderes Problem haben, wenden Sie sich an uns. Die Support-Seite zum Buch ist unter <http://www.hauser-wenz.de/support/> zu erreichen. Dort können Sie sich auch per E-Mail an uns wenden.

Zusätzlich betreibt der Galileo-Verlag unter <http://www.galileo-press.de> eine umfangreiche Website. Dort können Sie sich im Bereich MYGALILEO für dieses Buch registrieren. Die entsprechende Nummer finden Sie auf der Vorderseite der Infoklappe.

Straßenausbesserer

Ein Autor legt Stück für Stück einen Pflasterstein an den anderen, bis daraus ein Buch entsteht. Dabei gerät leicht in Vergessenheit, wie viel Arbeit außen herum notwendig ist. Deswegen möchten wir uns hier bei allen bedanken, die dieses Buch besser gemacht und unsere Arbeit erst ermöglicht haben.

Wir danken besonders unserer Lektorin Judith Stevens-Lemoine, die uns trotz längerer Abwesenheit in gewohnt hoher Qualität betreut hat. Des Weiteren gilt unser Dank Stefanie Kirchner für das Ausbessern der offensichtlichen und weniger offensichtlichen Schlaglöcher in unserer Sprache. Die verbleibenden Schäden sind uns zuzurechnen.

Sebastian Bauer hat uns in der Schlussphase mit vollem Einsatz unterstützt; auch dafür vielen Dank. Außerdem danken wir Jens Knall, der einige der schönsten Grafiken für dieses Buch gezaubert hat. Stefan L. Prestele von Macromedia Deutschland sei gedankt für die Unterstützung und die gute Zusammenarbeit.

Der persönliche Dank von Armin Kappler gilt Melanie und Uwe. Nach kurvenreicher Fahrt halfen sie ihm jederzeit mit ihren Navigationskünsten. Der persönliche Dank von Tobias Hauser gilt Marianne für eine tolle Fahrgemeinschaft.

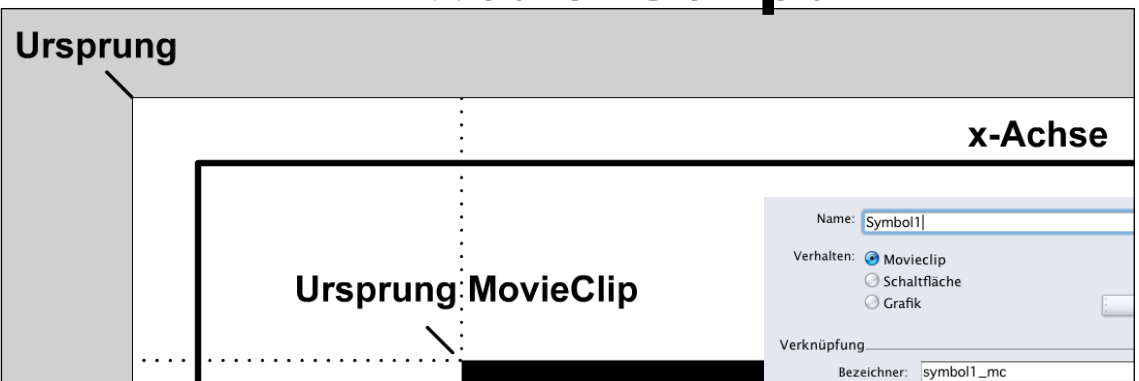
Viel Spaß und Erfolg auf Ihrem ganz persönlichen Weg zu Action-Script!

München, im Frühjahr 2004

Tobias Hauser, Armin Kappler und Christian Wenz

Teil I: Einfacher Einstieg

1 Flash für ActionScript



Flash-Tatsachen für ActionScript-Programmierer

- ▶ Wie ist ein Flash-Film aufgebaut?
- ▶ Welche Symbole gibt es und inwieweit sind sie für ActionScript relevant?
- ▶ Welche Bedeutung hat das Koordinatensystem?
- ▶ Wie funktionieren Verhalten?
- ▶ Wie wird aus der Flash-Datei ein SWF-Film?
- ▶ Was müssen Sie bei den verschiedenen Flash-Versionen beachten?

*Dem Himmel entriss er den Blitz und das Zepter den Herrschern.
– An Franklins Bildsäule*

ActionScript 1 und 2

ActionScript in Flash MX 2004 trägt die Versionsnummer 2. ActionScript in den Versionen davor ist also ActionScript 1. Was sich in Version 2 wirklich geändert hat, verrät das nächste Kapitel »ActionScript in Flash«.

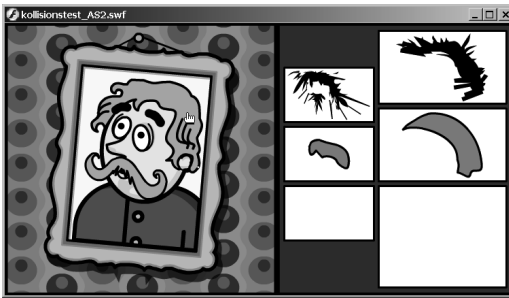
Mit Flash hat Macromedia eine Revolution in Gang gebracht: Vektorgrafik wurde im Internet salonfähig. Mit Version 5 kam dann eine zweite, etwas kleinere Revolution, die Flash selbst betraf: Die Aktionen in Flash wurden zu einer kompletten Skriptsprache, zu ActionScript. Macromedia lehnte ActionScript an ECMAScript an, das als Standard auch den Sprachkern von JavaScript (und Microsofts Klon JScript) bildet. Mit ActionScript gibt es auf einmal eine Menge an Möglichkeiten wie zufallsgesteuerte Animationen, programmierte Textänderungen und MovieClips, die sich per Programmcode bewegen. Interessante Welten tun sich dem Programmierer auf – und unser Buch möchte Sie mit hineinnehmen. Im Mittelpunkt stehen dabei die Flash-Versionen MX und MX 2004.

Um mit ActionScript produktiv und erfolgreich zu arbeiten, müssen Sie Flash beherrschen. Dieses Kapitel beleuchtet Flash aus dem Blickwinkel eines ActionScript-Programmierers. Obwohl wir die wichtigsten Themen ansprechen, können unsere wenigen Seiten ein Flash-Buch nicht ersetzen. Sie helfen aber, eine gemeinsame Sprache zu finden, so dass Sie immer wissen, wovon wir reden.

1.1 Zeitleiste

Flash ist ein klassisches Animationswerkzeug: Der produzierte Film läuft Bild für Bild ab, die Animation steuern Sie über die Zeitleiste. Was bedeutet das für ActionScript? Der Abspielkopf läuft in Flash ausschließlich über die Bilder der Zeitleiste; mit ActionScript können Sie ihn steuern (siehe Kapitel 5, »Einfache Filmsteuerung«) und darauf reagieren. Mit ActionScript lassen sich aber auch Flash-Filme mit viel Funktionalität schaffen, die nur aus einem einzigen Bild bestehen.

Wenn Sie in der Zeitleiste ActionScript einfügen, gibt es dafür nur eine geeignete Bildart: das Schlüsselbild. Nur ein Schlüsselbild kann ActionScript enthalten. Mehr dazu in Kapitel 2, »ActionScript in Flash«, im Abschnitt »Wo ist ActionScript möglich?«.



◀ **Abbildung 1.1**

In Kapitel 15, »Kollisions-test«, treffen Sie auf einen älteren Herrn, den Sie per Drag & Drop mit neuem Bart und Haarteil ausstatten können – alles in nur einem Schlüsselbild.

1.2 Symbole

Ein **Symbol** umfasst eine oder mehrere Vektorformen. In Flash gibt es drei Arten von Symbolen:

- ▶ Grafik-Symbole sind eine Sammelstelle für Formen. Die Zeitleiste des Grafik-Symbols läuft parallel zur Hauptzeitleiste. Über Grafik-Symbole werden Sie in diesem Buch nichts lesen, denn sie sind mit ActionScript nicht steuerbar und erlauben keinen ActionScript-Code.
- ▶ Schaltflächen-Symbole besitzen vier Zustände: UP (Normalzustand), DARÜBER (Mauspfeil über Schaltfläche), GEDRÜCKT und AKTIV. Diese Zustände sind rein grafisch, sie enthalten keinen ActionScript-Code. Allerdings können Sie Schaltflächen direkt mit ActionScript-Code versehen und bestimmte Ereignisse abfragen.
- ▶ MovieClip-Symbole besitzen eine eigene Zeitleiste, die ActionScript-Code erlaubt. Sie haben die Möglichkeit, auf MovieClips per ActionScript zuzugreifen und die MovieClips nach Belieben in Größe, Position und Aussehen zu verändern. MovieClips sind das flexibelste und wichtigste Element eines Flash-Films, wenn es um die Steuerung mit Flash geht.

Sie erstellen ein Symbol aus bestehenden Objekten mit dem Befehl **IN SYMBOL KONVERTIEREN**. Diesen Befehl finden Sie im Menü **MODIFIZIEREN** oder im Kontextmenü der Objekte (rechte Maustaste, beim Mac: **Ctrl** + Maustaste). Mit dem Befehl **EINFÜGEN • NEUES SYMBOL** fügen Sie ein neues Symbol noch ohne Inhalt hinzu.

MovieClips und Filmsequenzen

In früheren deutschen Flash-Versionen wurden MovieClips als Filmsequenzen bezeichnet; diese Übersetzung verschwand in Flash MX. Die Schreibweise mit großem »C« ist nicht immer einheitlich: Im **EIGENSCHAFTENINSPEKTOR** und beim Erstellen neuer Symbole enthalten MovieClips nur ein kleines »c«. Wir verwenden die Schreibweise mit großem »C«, da diese der zugehörigen Klasse in ActionScript entspricht. Eine Klasse ist in der Programmierung eine Struktur, die den Zugriff auf Funktionalität gewährt. Mehr dazu erfahren Sie in Kapitel 4, »Grundlagen der Programmierung«.

In Symbol konvertieren:

F8

Neues Symbol:

Strg bzw. **⌘** + **F8**

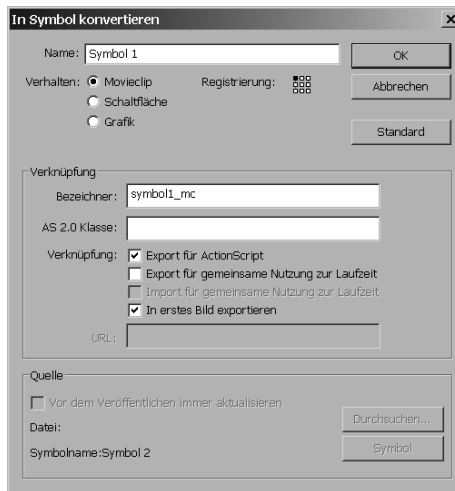
Abbildung 1.2 ▶
Das Dialogfeld für das
Konvertieren des Symbols



Die Dialogfelder für das Konvertieren beziehungsweise für ein neues Symbol gleichen sich wie ein Ei dem anderen – bis auf den Namen. Hier haben Sie mehrere Wahlmöglichkeiten:

- ▶ Bei VERHALTEN wählen Sie die Symbolart.
- ▶ Bei NAME geben Sie dem Symbol einen Namen. Dies ist der Name des Symbols in der Bibliothek. Er ist für ActionScript nicht relevant.
- ▶ REGISTRIERUNG teilt Ihnen mit, an welcher Position das Symbol seinen Nullpunkt des Koordinatensystems hat. Dazu müssen Sie wissen: Jedes Symbol hat ein eigenes Koordinatensystem, das dem Koordinatensystem des Hauptfilms untergeordnet ist. Dieser Fakt kommt vor allem bei MovieClips zum Tragen; mehr dazu im Abschnitt »Koordinatensystem und Bühne«.

Abbildung 1.3 ▶
Die erweiterte Variante
des Dialogfelds

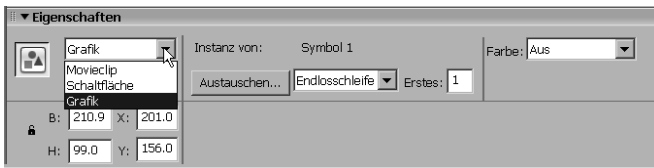


Vielleicht ist Ihnen die Schaltfläche ERWEITERT aufgefallen. Sie enthüllt weitere Optionen, die für ActionScript-Programmierer von Bedeutung sind. Wenn Sie die Schaltfläche EXPORT FÜR ACTIONSCRIPT anklicken, kann ein Zugriff auf das Symbol über ActionScript erfolgen. Vor allem für MovieClips ist das wichtig: Sobald Sie einen

BEZEICHNER angeben, lässt sich der MovieClip per ActionScript auf die Bühne bringen (siehe Kapitel 14, »Von Mäusen und Duplikaten«).

Verwaltet werden die Symbole in der Bibliothek. Ein Symbol auf der Bühne, dem zentralen Arbeitsraum von Flash, ist die **Instanz** des Symbols in der Bibliothek. Ein Symbol in der Bibliothek kann mehrere Instanzen haben. Die Instanzen lassen sich in Position, Größe und Farbe verändern. Flash spart mit Symbolen Dateigröße, da ein Symbol nur einmal gespeichert werden muss. Für die Instanzen benötigt Flash dann kaum mehr Platz, da nur noch die Koordinaten und Änderungen gegenüber dem Original-Symbol zu speichern sind.

Sie können übrigens für Instanzen die Symbol-Art ändern. Diese Änderung, die Sie im EIGENSCHAFTENINSPEKTOR vornehmen, gilt allerdings nur für die jeweilige Instanz. Deswegen ist eine solche Änderung meist nicht sinnvoll.



Bibliothek:

F11

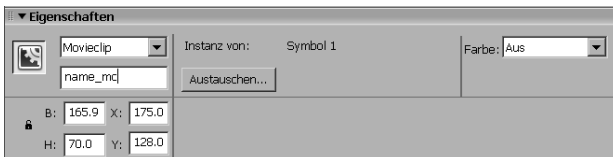
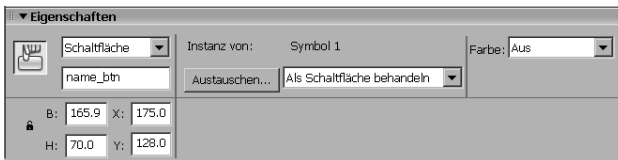
Namensgebung

In der Abbildung sehen Sie, dass der Instanzname der Schaltfläche auf `_btn` endet, der Name des MovieClips auf `_mc`. Wir greifen ein wenig voraus: Diese Endungen sorgen im AKTIONEN-Bedienfeld für Codehinweise, die Ihnen Tipps geben, welche Befehle für die Schaltfläche oder den MovieClip zur Verfügung stehen.

◀ Abbildung 1.4

Im EIGENSCHAFTENINSPEKTOR ändern Sie die Art der Instanz, nicht die Art des Symbols.

Sie haben schon den ersten Anknüpfungspunkt für ActionScript kennen gelernt: EXPORT FÜR ACTIONSCRIPT. Der zweite und wichtigere Punkt ist der Instanzname. Ihn vergeben Sie für jede Instanz auf der Bühne im EIGENSCHAFTENINSPEKTOR. Grafik-Symbole besitzen natürlich keinen Instanznamen, da sie über ActionScript nicht steuerbar sind.



◀ Abbildung 1.5

Die Instanznamen vergeben Sie im EIGENSCHAFTENINSPEKTOR (oben für eine Schaltfläche, unten für einen MovieClip).

1.3 Koordinatensystem und Bühne

In Flash ziehen Sie die Objekte beliebig über die Bühne. Um Koordinaten müssen Sie sich dabei nicht unbedingt kümmern. Das Positionieren übernehmen Positionshilfen oder das Bedienfeld AUSRICHTEN. Wenn Sie mit ActionScript per Programmierung die Position eines MovieClips verändern, geht es aber nicht mehr ohne Koordinaten.

Die Grundlage des Koordinatensystems ist die Dokumentgröße. Sie ändern sie im EIGENSCHAFTENINSPEKTOR nach einem Klick auf GRÖSSE.

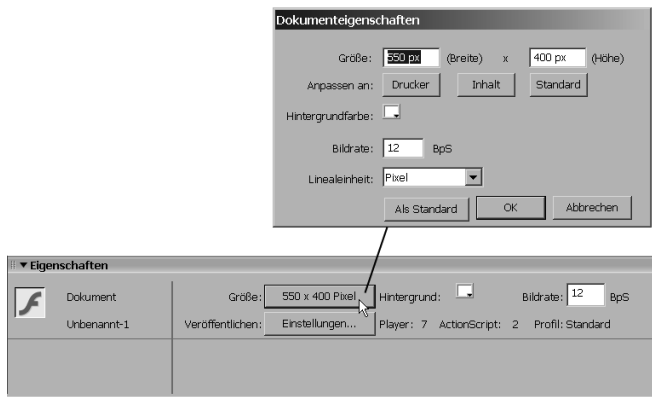
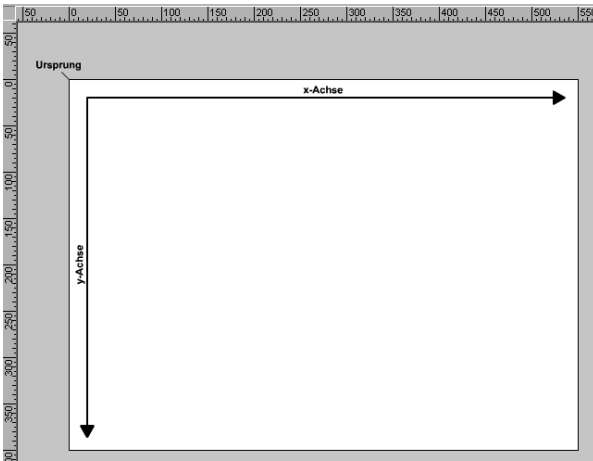


Abbildung 1.6 ▶
Die Größe der Bühne ist die erste wichtige Wahl.

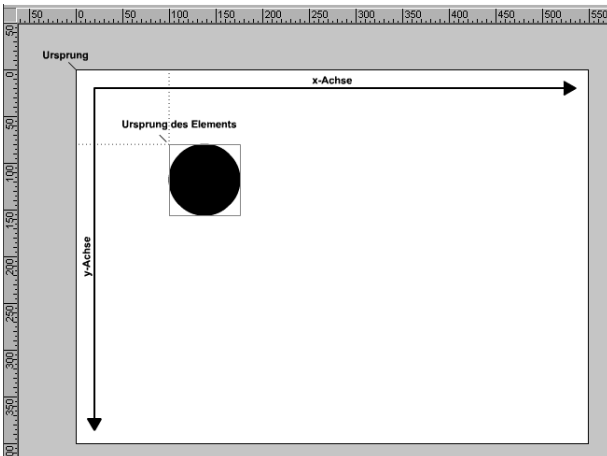
Sehen Sie sich das Koordinatensystem etwas genauer an:

- ▶ Das Koordinatensystem der Bühne hat seinen Ursprung in der linken oberen Ecke der Bühne. Sie blenden es mit dem Befehl ANSICHT • LINEALE ein.
- ▶ Die x-Achse verläuft horizontal vom Ursprung nach rechts (positive Werte). Auf der linken Seite außerhalb der Bühne befinden sich negative x-Koordinaten.
- ▶ Die y-Achse verläuft vertikal vom Ursprung nach unten (positive Werte). Die negativen Werte befinden sich oberhalb der Bühne.



◀ **Abbildung 1.7**
Das Koordinatensystem
der Bühne

Die Koordinaten, die Sie für Elemente wie Formen und Gruppen (nicht Symbole) im EIGENSCHAFTENINSPEKTOR sehen (Feld x und Feld y), beziehen sich auf das Koordinatensystem der Bühne, wenn die Elemente direkt auf der Bühne liegen. Sie werden von der linken oberen Ecke des Elements gemessen. Vorsicht, der Rand ragt zur Hälfte hinein und die Begrenzung ist bei ungruppierten Formen nicht sichtbar.



◀ **Abbildung 1.8**
Der Ursprungspunkt einer
Gruppe oder Form, der im
EIGENSCHAFTENINSPEKTOR die
Koordinate bildet, liegt immer
in der linken oberen Ecke der
Begrenzung.

Symbole haben ein eigenes Koordinatensystem, das seinen Ursprung im Registrierungspunkt hat. Entsprechend werden die Koordinaten eines Symbols auf der Hauptbühne auch vom Registrierungs-

Umwandeln

Das Koordinatensystem eines Symbols bezeichnet man auch als lokales Koordinatensystem. ActionScript bietet Methoden, um von diesem lokalen auf das globale Koordinatensystem und umgekehrt umzuwandeln. Ein Einsatzgebiet finden Sie in Kapitel 15, »Kollisionstest«.

punkt aus gemessen. Und zwar unabhängig vom Inhalt des Symbols. In Abbildung 1.9 hat der MovieClip in beiden Fällen die gleichen Koordinaten, da die Position seines Registrierungspunktes gleich ist. Allerdings sind die Elemente jeweils völlig anders angeordnet.

Wenn Sie ein Symbol erstellen, wählen Sie die Position des Registrierungspunktes. Selbstverständlich lassen sich die Formen, die das Symbol bilden, jederzeit verschieben. Sie können sich den Registrierungspunkt wie einen Dreh- und Angelpunkt vorstellen: Da das Koordinatensystem von Symbolen aufgebaut ist wie das Koordinatensystem der Hauptbühne, bildet der Registrierungspunkt den Ursprung. Von ihm aus verläuft die x-Achse horizontal und nimmt nach rechts positive Werte an; die y-Achse geht vertikal nach unten mit ebenfalls positiven Werten.

Abbildung 1.9 ▶

Die Koordinaten eines Symbols werden von seinem Registrierungspunkt aus gemessen – deswegen hat der MovieClip in beiden Abbildungen die gleichen Koordinaten.

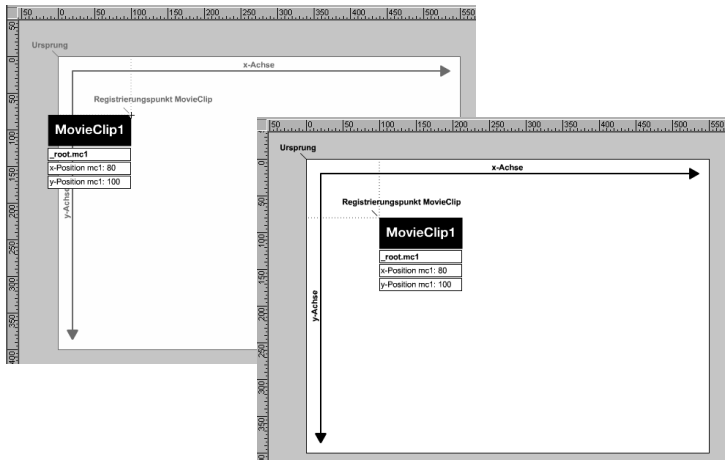
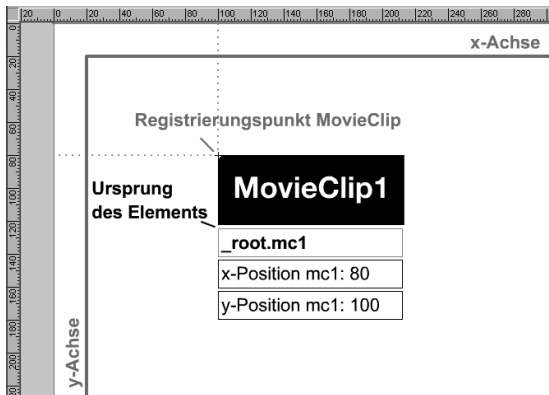


Abbildung 1.10 ▶

Das Koordinatensystem eines Symbols – das Element (Textfeld) hat die Koordinaten 0 (x) und 45 (y), da sein Ursprung auf der x-Achse genau auf dem Registrierungspunkt des MovieClips liegt und auf der y-Achse um 45 Pixel nach unten verschoben ist.



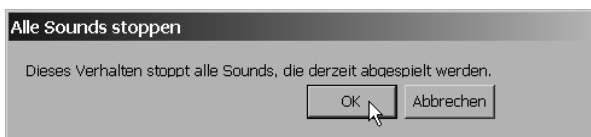
1.4 Verhalten

Das AKTIONEN-Bedienfeld ist der Ort, an dem ActionScript programmiert wird (nähere Informationen in Kapitel 2, »ActionScript in Flash«). In Flash MX 2004 fällt bei diesem Bedienfeld der Normalmodus weg. In diesem Modus konnte sich der Flash-Designer ActionScript-Befehle relativ bequem zusammenklicken. Stattdessen gibt es in Version 2004 das VERHALTEN-Bedienfeld. Verhalten sind vordefinierte ActionScript-Codes, die bestimmte Dinge erledigen. Im VERHALTEN-Bedienfeld können Sie per Klick auf das PLUS-Symbol neue Verhalten hinzufügen.



◀ **Abbildung 1.11**
Mit dem Plus-Symbol fügen Sie neue Verhalten hinzu.

Bei den meisten Verhalten erscheint eine Abfrage oder Sie müssen bestimmte Werte für das Verhalten, so genannte Parameter, eingeben. Die bestehenden Verhalten lassen sich mit ihren jeweiligen Parametern im Bedienfeld verändern.



◀ **Abbildung 1.12**
Für das Verhalten ALLE SOUNDS STOPPEN müssen Sie nur mit OK bestätigen.

Das Verhalten macht Folgendes: Es fügt vorgefertigten ActionScript-Code ein. Bei ALLE SOUNDS STOPPEN ist das beispielsweise der Code:

```
//stopAllSounds Behavior
stopAllSounds();
```

Vermutlich sagen Ihnen diese Zeilen noch recht wenig. Nur so viel sei Ihnen verraten: Die obere Zeile ist ein Kommentar, der den Namen des Verhaltens impliziert. Wenn Sie den Namen ändern, ist das Verhalten aus dem VERHALTEN-Bedienfeld verschwunden, da

Wissen ist Macht

Verhalten helfen nur wenig, wenn Sie von ActionScript keine Ahnung haben. Sie müssen zum Beispiel wissen, wo Sie ActionScript einfügen können, denn je nachdem was gerade aktiv ist – ein MovieClip, eine Schaltfläche, ein Schlüsselbild –, gibt es unterschiedliche Verhalten. Für Verhalten gelten dieselben Regeln wie für normalen ActionScript-Code; sie sagen Ihnen, wo und wofür Sie ActionScript-Code hinzufügen können (die Regeln finden Sie in Kapitel 2, »ActionScript in Flash«).

Flash die Kommentarzeile zum Identifizieren des Verhaltens verwendet.

Verhalten können Ihnen durchaus Arbeit abnehmen, beispielsweise wenn Sie nicht wissen, dass der Befehl zum Anhalten aller Sounds `stopAllSounds()` heißt. Wir zeigen Ihnen ausführlich den Einsatz von Verhalten in Kapitel 6, »Bildschirme und Präsentationen«.

1.5 Veröffentlichen und Testen

Flash MX bzw. MX 2004 ist eine Entwicklungsumgebung. Das produzierte Ergebnis ist ein SWF-Film. SWF ist eigentlich eine eigene Sprache und steht mittlerweile für Flash File Format (<http://www.macromedia.com/software/flash/open/licensing/fileformat/>). Die Bezeichnung hatte sich einige Male geändert, ursprünglich hieß SWF Shockwave Flash in Anlehnung an das Shockwave-Format von Macromedia Director. Macromedia benannte es dann in Small Web Format um. Mittlerweile tituliert Macromedia es auf der eigenen Website als Flash File Format.

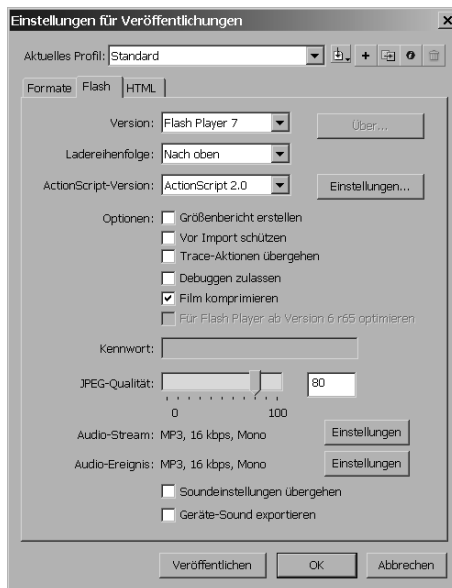


Abbildung 1.13 ►

Die EINSTELLUNGEN FÜR VERÖFFENTLICHUNGEN erlauben die Steuerung, was als SWF-Datei veröffentlicht wird.

Wenn die Entwicklungsumgebung Flash einen SWF-Film produziert, dann nennt Macromedia diesen Vorgang Kompilieren. SWF wird anschließend vom Flash-Player interpretiert. Flash verwendet also eine Mischung aus Kompilieren und Interpretieren.

Das Kompilieren von der Entwicklungsumgebung in SWF heißt in Flash VERÖFFENTLICHEN. Sie finden den entsprechenden Befehl im Menü DATEI. Aus Sicht des ActionScript-Entwicklers ist hier besonders wichtig, für welchen Flash-Player Sie veröffentlichen und ob Sie (bei Flash MX 2004) ActionScript 1 oder ActionScript 2 schreiben. Änderungen bei diesen Punkten können Sie in den EINSTELLUNGEN FÜR VERÖFFENTLICHUNGEN im Menü DATEI vornehmen. Das Register FLASH enthält alle Optionen für den SWF-Film.

Mit den EINSTELLUNGEN FÜR VERÖFFENTLICHUNGEN arbeiten Sie auch, wenn Sie den Flash-Film testen. Sie können den Flash-Film zwar im Menü STEUERUNG abspielen; ActionScript kommt allerdings nur zum Tragen, wenn Sie den Befehl STEUERUNG • FILM TESTEN verwenden. Dann wird nämlich von Flash automatisch ein SWF-Film erstellt und im Flash-Player ausgeführt. Für die Fehlersuche besteht dabei immer noch Anbindung an die Entwicklungsumgebung (Näheres dazu im nächsten Kapitel).

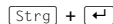

1.6 Flash-Versionen

Die Abbildungen in diesem Buch sind zum Großteil unter Flash MX 2004 entstanden. Viele Erklärungen und die Beispiele geben wir allerdings auch in anderen Versionen an, so dass Sie mit dem Buch gut arbeiten können, selbst wenn Sie Flash MX verwenden. Ebenso können Flash 5-Nutzer – bis auf wenige Einschränkungen – durchaus Anregungen aus dem Buch übernehmen, da ActionScript 1 eigentlich alle Flash-Versionen unter MX 2004 meint. Mit Flash MX 2004 hat Macromedia eine zweite Flash-Version eingeführt: Die Professional-Version. Sie wendet sich als Zielgruppe direkt an den Entwickler und bietet einige Neuerungen, die Thema des dritten Kapitels sind. Nur wenige Beispiele in diesem Buch sind auf die Professional-Version beschränkt; dies wird jeweils mit einem Icon angezeigt. Meist finden Sie danach eine Alternative, wie Sie dasselbe Ergebnis auch mit der Standard-Version von Flash MX 2004 und mit Flash MX erzielen können.

Alternativen

Nicht nur Flash produziert SWF-Dateien. Eine bekannte Alternative ist SWiSH (<http://www.swishzone.com/>). Auch ein Open Source-Projekt gibt es zu SWF: <http://www.openswf.org/about.html>. Da der Flash-Player und sämtliche Neuerungen in SWF von Macromedia entwickelt werden, hinkt die Konkurrenz immer einen Schritt hinterher – ein Grund, warum Adobe seine Bemühungen mit LiveMotion eingestellt hat. Auch bieten Alternativen wie SWiSH kein ActionScript, sondern eine eigene Skriptsprache, nämlich SWiSHscript.

Film testen:

 **Strg** + 

Deutsch und Englisch

Einige Leser teilten uns mit, dass sie eine englische Version von Flash besitzen; deswegen baten sie uns, die Menübefehle deutsch und englisch anzugeben. Schweren Herzens haben wir uns dagegen entschieden, da uns das sprachliche Mischen einzelner Befehle zu unübersichtlich und störend für den Lesefluss erscheint. Allerdings ist der Aufbau der Menüs exakt gleich, die Umstellung dürfte Ihnen nicht schwer fallen. →

→ Bei der Programmierung stellt die Sprache dankenswerterweise keine Barriere dar, da alle Klassen, Methoden, Eigenschaften und Befehle in ActionScript sowieso englisch und in allen Flash-Versionen gleich sind.

Windows und Mac

Die Abbildungen dieses Buches entstanden nicht nur mit Flash MX 2004, sondern auch unter Windows. Eingefleischte Mac-Fans mögen nun vermuten, die Autoren seien Mac-Gegner. Dem ist aber nicht so, ganz im Gegenteil; Einer der Autoren verwendet ausschließlich Mac, die anderen beiden sind Mac-erfahren. Grund für unsere Entscheidung war nur, dass die für die Produktion des Buches verwendete Vorlage auf Windows optimiert ist.

Ein zweiter Grund, warum wir uns auf ein Betriebssystem beschränken, ist sehr einfach: Es gibt kaum Unterschiede zu anderen Systemen. Die wenigen, die vorhanden sind, wollen wir im Folgenden kurz nennen.

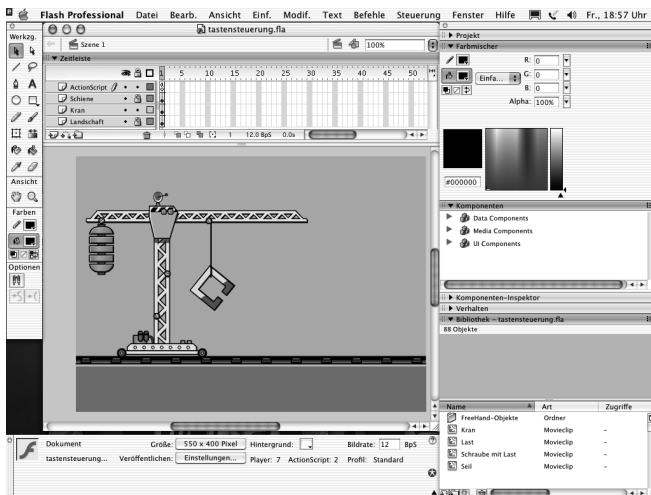


Abbildung 1.14 ►
Flash MX 2004 unter
MacOS X

Wenn Sie einen Blick auf Flash am Mac werfen (siehe Abbildung 1.14), sehen Sie genau dieselben Bedienfelder, dieselbe Bühne, den EIGENSCHAFTENINSPEKTOR, die Zeitleiste, kurz: viele alte Bekannte. Auch im Handling von ActionScript finden Sie keine Unterschiede.

Eine Besonderheit in der Oberfläche gibt es allerdings: Unter MacOS X hat sich das Programmmenü als Standard eingebürgert. Diese Neuerung der Aqua-Oberfläche heißt bei Flash natürlich FLASH bzw. FLASH PROFESSIONAL (siehe Abbildung 1.15). Sie finden dort als wichtigste Einstellungen die VOREINSTELLUNGEN und TASTENKOMBINATIONEN, die vorher unter BEARBEITEN verborgen waren. Wenn wir also im Verlauf dieses Kapitels und des Buches von VOREINSTELLUN-

GEN im Menü BEARBEITEN reden, wissen Aqua-Anhänger, dass sie im Programmmenü fündig werden.



◀ **Abbildung 1.15**
Das Programmmenü FLASH

Den letzten, etwas größeren Unterschied zwischen Mac und Windows bilden die Tastenkürzel. Für ActionScript-Fans fällt der Unterschied nicht so stark ins Gewicht, da es für sie sowieso nicht sehr viele Kürzel gibt, wie der Anhang zu Tastenkürzeln zeigt. Dennoch seien hier die verschiedenen Tasten erwähnt:

- ▶ Die **Strg**-Taste heißt am Macintosh Befehlstaste **⌘** (Apfel-Taste **⌘**). Wenn Sie also **Strg** sehen, müssen Sie den netten Apfel/Propeller/Doppelkreuz drücken.
- ▶ Die **Alt**-Taste entspricht am Macintosh der Wahltaaste **⇧**, die bei optischen Vergleichen ein bisschen an eine Badewanne oder Weiche erinnert.
- ▶ Das Kontextmenü erreichen Sie am Macintosh mit einer Ein-Tasten-Maus, indem Sie die **Ctrl**-Taste drücken und an die gewünschte Stelle mit der Maus klicken.
- ▶ Die **Entf**-Taste ist am Mac nur bei erweiterterem Tastaturlayout vorhanden. Sonst arbeiten Sie stattdessen mit Backspace **←**.

Teil IV: Animieren

12 Farbe wechseln und Drag & Drop



*Bewegung und Veränderung als
Triebfedern der Animation*

- ▶ Wie sorgt ActionScript für spannende Animationen?
- ▶ Wie arbeiten Sie mit Drag & Drop?
- ▶ Wie wechselt die Farbe eines MovieClips?

Der Wechsel sehr gefährlich ist.
– Rollenhagen

Flash ist ein Animationswerkzeug par excellence. Schon die Zeit-
leiste deutet darauf hin: Es gibt Tweens und in Flash MX 2004 Zeit-
leisten-Effekte für gekonnte Übergänge. Warum soll also unbedingt
ActionScript für die Animation zum Einsatz kommen? Diese Frage
beantwortet nicht nur das vorliegende Kapitel, sondern alle vier Ka-
pitel in diesem Teil des Buches. Nur so viel sei vorab schon gesagt:
Animation gewinnt Spannung durch Interaktion mit dem Nutzer und
durch überraschende Elemente. Und beides lässt sich optimal mit
ActionScript realisieren.

12.1 Animation mit ActionScript

Wenn Sie das Buch von vorne nach hinten durchgeblättert oder be-
reits durchgearbeitet haben, ist Ihnen schon so manche Animation
begegnet. Paradebeispiel ist sicher der bewegliche Kran aus dem
Kapitel 9, »Tasten«. Dort und in den meisten anderen Fällen auch
sind MovieClips der Dreh- und Angelpunkt für das Animieren mit
ActionScript.

Per Instanzname greifen Sie auf den MovieClip zu. Für Action-
Script ist ein MovieClip ein Objekt. Die übergeordnete Klasse finden
Sie im AKTIONEN-Bedienfeld unter INTEGRIERTE KLASSEN • FILM • MO-
VIECLIP. Einige der Eigenschaften und Methoden haben Sie bereits
kennen gelernt. In diesem vierten Teil des Buches treffen Sie auf wei-
tere Optionen, deren Bedeutung und Einsatz wir Ihnen erklären.

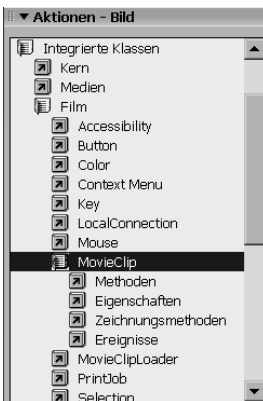


Abbildung 12.1 ▲
Die MovieClip-Klasse im
AKTIONEN-Bedienfeld.

12.2 Wechselspiele

Die Eigenschaften des MovieClip-Objekts sind besonders gut für
optische Modifikationen geeignet: Mit ihnen können Sie die Größe
verändern oder die Position. Beides kennen Sie bereits aus dem
Kran-Beispiel in Kapitel 9, »Tasten«. Interessant ist auch die Eigen-
schaft `_visible`, die einen Wahrheitswert erhält: `true` blendet den
MovieClip ein, `false` blendet ihn aus.

Der Wechsel von Farben gestaltet sich ein wenig schwieriger.
Hierfür hält ActionScript die `Color`-Klasse bereit, die Sie wie die

MovieClip-Klasse im AKTIONEN-Bedienfeld unter INTEGRIERTE KLASSEN • FILM finden.

Und so funktioniert es:

1. Für eine neue Farbe müssen Sie ein Color-Objekt instanziiieren:

```
var farbe_color:Color = new Color(movieclip_mc);
```

Um Codehinweise zu erhalten, fügen Sie der Variablen, die das Color-Objekt aufnimmt, die Namensendung `_color` hinzu. Entscheidend ist der MovieClip als Parameter des Color-Objekts. Dieser MovieClip kann mit dem Color-Objekt gefärbt werden.

2. Zum Färben verwenden Sie die Methode `setRGB(Farbe)`. Sie setzt eine Farbe als hexadezimalen Wert ein:

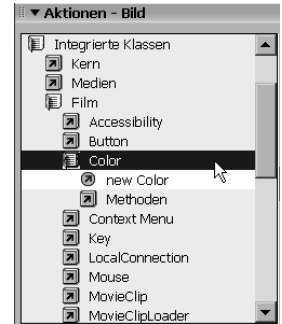
```
farbe_color.setRGB(0xFFFFFF);
```

Der MovieClip übernimmt den Farbwert sofort. Wollen Sie einen bestehenden Farbwert auslesen, verwenden Sie `getRGB()`.

Wie das Zuweisen von Farben in der Praxis funktioniert, zeigen wir Ihnen an einem Beispiel: Ausgangspunkt ist die Datei CHAMAELEON.FLA auf der CD-ROM im Ordner FLASHMX2004 bzw. FLASHMX. Sie besteht aus einem Chamäleon und einer Palette mit Schaltflächen. Natürlich kommt es in der Praxis eher selten vor, dass Sie die Farbe eines Chamäleons wechseln müssen – schließlich kann es das ja selbst. Aber ersetzen Sie das Tier (in Gedanken) einfach durch ein beliebiges Produkt, das der Nutzer konfigurieren soll, schon hat das Beispiel Praxiswert.



Nun sollen Sie die Schaltflächen zum Öffnen und Schließen der Augen und zum Umfärben des Chamäleons in Betrieb nehmen.



▲ **Abbildung 12.2**

Die Color-Klasse im AKTIONEN-Bedienfeld

Farben wechseln

Bei Verläufen und komplexeren Objekten scheitert der simple Farbwechsel. Hier gibt es Alternativen: Entweder blenden Sie die verschiedenen Objekte mit `_visible` ein und aus oder Sie versehen ein MovieClip mit mehreren Bildern. Diese gestalten Sie in unterschiedlichen Farben und arbeiten dann mit `gotoAndStop()` (siehe hierzu die Beispiele aus dem Kapitel 5, »Einfache Filmsteuerung«).

◀ **Abbildung 12.3**

Das Chamäleon im ursprünglichen Zustand

Fehlerquelle

Die häufigste Fehlerquelle bei MovieClips besteht darin, den Instanznamen oder die Adresse in der Hierarchie falsch anzugeben. Lassen Sie sich ruhig vom ZIELPFAD-Editor helfen.

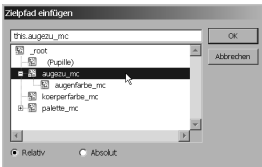
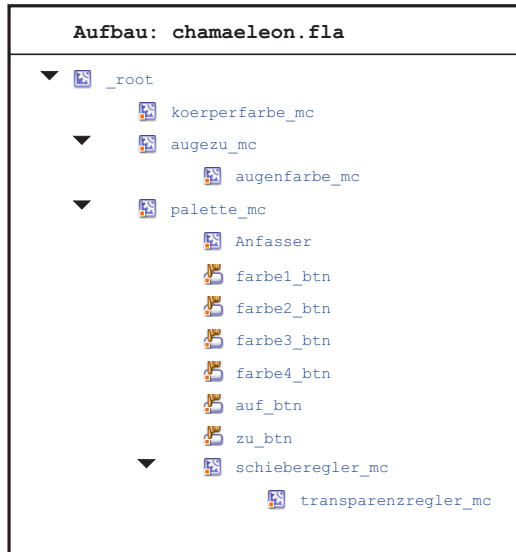


Abbildung 12.4 ▶

Der Film mit dem Chamäleon in der Übersicht

Wichtig ist dabei, die MovieClip-Hierarchie immer gut im Blick zu behalten. Die Datei ist so angelegt, dass alle MovieClips sauber untereinander angelegt sind.



Schritt für Schritt: Chamäleon umfärben

Der Code für die folgenden Schritte landet im ersten Schlüsselbild des Hauptfilms auf der Ebene ACTIONSCRIPT, da Sie für den Zugriff auf die Schaltflächen Ereignisprozeduren einsetzen.

1. Auge öffnen

Wenn der Nutzer auf die Schaltfläche AUF klickt, öffnen Sie das Auge:

```
palette_mc.auf_btn.onRelease = function() {
    augezu_mc._visible = false;
};
```

Hier handelt es sich um einen einfachen, aber sehr wirkungsvollen Trick: Das geschlossene Auge ist ein MovieClip, der in einer Ebene über der Grafik des offenen Auges liegt. Soll das Auge aufgehen, blenden Sie einfach den oberen MovieClip aus (_visible auf false setzen).

Für die Schaltfläche Zu blenden Sie den MovieClip mit dem geschlossenen Auge wieder ein:

```
palette_mc.zu_btn.onRelease = function() {
    augezu_mc._visible = true;
};
```



2. Auge schließen

◀ **Abbildung 12.5**

Das Chamäleon zwinkert Ihnen zu.

Für den Farbwechsel des Chamäleons müssen Sie erst zwei `Color`-Objekte instanziiieren – eines für den Körper des Chamäleons (`koerperfarbe_mc`) und eines für den MovieClip der Augenfarbe (`augenfarbe_mc`):

```
var koerperfarbe_color:Color = new
Color(koerperfarbe_mc);
var augenfarbe_color:Color = new Color(augezu_mc
.augenfarbe_mc);
```

Zum Umfärben weisen Sie den `Color`-Objekten mit der Methode `setRGB(Farbe)` eine neue Farbe zu, wenn der Nutzer auf die Farb-Schaltfläche klickt:

```
palette_mc.farbe1_btn.onRelease = function() {
    koerperfarbe_color.setRGB(0xE20916);
    augenfarbe_color.setRGB(0xE20916);
};
```

Anschließend müssen Sie nach demselben Muster die Farben bei den drei übrigen Farb-Schaltflächen wechseln:

```
palette_mc.farbe2_btn.onRelease = function() {
    koerperfarbe_color.setRGB(0x694420);
    augenfarbe_color.setRGB(0x694420);
};
palette_mc.farbe3_btn.onRelease = function() {
    koerperfarbe_color.setRGB(0xB4A714);
```

3. Color-Objekte instanziiieren

4. Umfärben

5. Die übrigen Farben

```

    augenfarbe_color.setRGB(0xB4A714);
};
palette_mc.farbe4_btn.onRelease = function() {
    koerperfarbe_color.setRGB(0x5D7C27);
    augenfarbe_color.setRGB(0x5D7C27);
};

```

Ende

Abbildung 12.6 ▶
Das Chamäleon
in Braun und Rot



So viel zur Farbwelt des Chamäleons. Wir hoffen, unsere bisherigen Ausführungen animieren Sie, Ihre eigenen Produkte oder die Produkte Ihrer Kunden in verschiedenen Konfigurationen anzubieten. Die Datei mit dem Code finden Sie auf der CD-ROM unter dem Namen CHAMAELEON_WECHSELSPIELE_AS2.FLA im Ordner FLASHMX2004 und unter CHAMAELEON_WECHSELSPIELE_AS1.FLA im Ordner FLASHMX.

12.3 Drag & Drop

Drag & Drop steht für Ziehen und Loslassen und hat in der Computer-Fachsprache mit den optischen Betriebssystemen Einzug gehalten. In Flash können Sie Drag & Drop sehr einfach und schnell mit Hilfe eines MovieClips implementieren. Das `MovieClip`-Objekt besitzt zwei Methoden, die Ihnen weiterhelfen:

- ▶ `startDrag(MovieClip)` beginnt den Drag & Drop-Vorgang für den `MovieClip`, der als erster Parameter angegeben ist. Die Methode gibt es noch in zwei anderen Varianten mit mehr Parametern:
 - ▶ `startDrag(MovieClip, Zentrum)` – hier gibt der Parameter `Zentrum` an, wo die Maus bei Drag & Drop einrastet: beim Wert `true` am Mittelpunkt des `MovieClips`, bei `false` an der Position, wo der `MovieClip` angeklickt wurde.

- ▶ `startDrag(MovieClip, Zentrum, x1, y1, x2, y2)` – hier definieren vier Koordinaten ein Rechteck. Die Bewegung des MovieClips ist auf dieses Rechteck beschränkt.
- ▶ `stopDrag()` beendet Drag & Drop für alle aktuell gezogenen MovieClips und lässt sie an der Stelle, wo sie sich befinden, einrasten.

Um vernünftiges Drag & Drop zu realisieren, benötigen Sie ein Ereignis für das Starten von Drag & Drop (`startDrag()`) und ein Ereignis für das Einrasten des MovieClips (`stopDrag()`).

Wie das geht, zeigen wir Ihnen anhand des Chamäleon-Beispiels aus dem letzten Abschnitt. Wir erweitern es so, dass der Nutzer die Palette mit den Schaltflächen beliebig durch die Gegend ziehen darf.

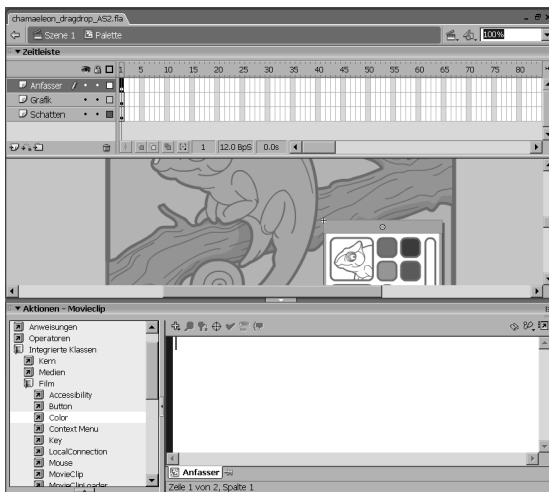
Schritt für Schritt: Drag & Drop der Palette

Wenn Sie das Beispiel aus dem letzten Abschnitt nicht in der Praxis nachvollzogen haben, können Sie die Datei CHAMAELEON_WECHSELSPIELE_AS2.FLA als Ausgangspunkt verwenden.

Klicken Sie auf die Palette. Innerhalb dieses MovieClips finden Sie eine graue Leiste. Sie ist ein eigener MovieClip und heißt `Anfasser`. Im AKTIONEN-Bedienfeld können Sie nun für diesen MovieClip den Code eingeben.



1. In den MovieClip `Anfasser` wechseln



◀ **Abbildung 12.7**
Im AKTIONEN-Bedienfeld geben Sie den Code für den `Anfasser`-MovieClip ein.

2. Ereignis hinzufügen

Um Drag & Drop zu starten, verwenden Sie das Ereignis `press`:

```
on (press) {
```

`press` ist hier dem Anfasser, also dem grauen Balken am oberen Rand der Palette, zugewiesen. Wenn der Nutzer auf diesen Balken klickt, soll Drag & Drop starten.

3. Drag & Drop starten

Verwenden Sie `startDrag()`, um mit Drag & Drop zu beginnen:

```
    startDrag(_root.palette_mc, false, 3, 3, 357,
213);
}
```

Als Parameter kommt zuerst der betroffene `MovieClip`, nämlich die Palette, zum Einsatz. Sie wird hier absolut adressiert; relativ würde die Adresse `_parent.palette_mc` heißen, da sie dem Anfasser-`MovieClip` übergeordnet ist.

Der zweite Parameter gibt an, dass beim Ziehen die Palette an der Position des Mauszeigers einrastet. Die vier Koordinaten in den folgenden Parametern verhindern, dass die Palette über den Rahmen des Flash-Films gezogen werden kann.

4. Drag & Drop anhalten

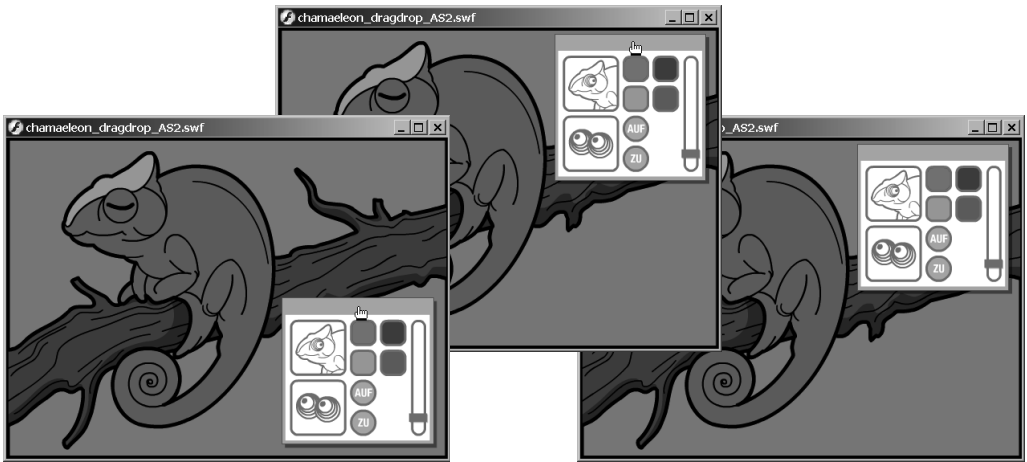
Zum Anhalten von Drag & Drop verwenden Sie die Methode `stopDrag()`, wenn der Nutzer die Maustaste über dem Anfasser-`MovieClip` loslässt:

```
on (release, releaseOutside) {
    stopDrag();
}
```

Als zweites Ereignis neben `release` kommt `releaseOutside` zum Einsatz. Das ist praktisch und hilfreich, falls der Nutzer sehr schnelle Mausbewegungen macht und die Maustaste aus Versehen außerhalb des `MovieClips` loslässt. Wäre in diesem Fall nur `release` aktiviert, würde Drag & Drop nicht stoppen.

Ende

Die Datei mit Drag & Drop trägt den Namen `CHAMAELEON_DRAGDROP_AS2.FLA` bzw. `CHAMAELEON_DRAGDROP_AS1.FLA`. Solche beweglichen Navigationselemente sind nicht nur eine nette, verspielte Idee, sondern sie können die Grundlage für interessante Produktkonfigurationen und spannende Anwendungen darstellen.



▲ **Abbildung 12.8**

Der Nutzer klickt auf den Anfasser, zieht und lässt los, um die Palette an der neuen Position einzurasten.

12.4 Schieberegler

Ein Element des Chamäleon-Beispiels blieb bisher unbeachtet: der Schieberegler an der rechten Seite der Palette. Ein Schieberegler gehört zu den häufiger benötigten Elementen, dennoch besteht das Vorurteil, es sei schwierig einen eigenen zu realisieren. Doch mit Ihrem bisher angesammelten Wissen können Sie sich durchaus an seine Gestaltung wagen. Der Schieberegler in der Palette soll die Deckkraft des Chamäleons anpassen, das heißt die Farbintensität des Chamäleons wird damit festgelegt. Die Deckkraft selbst ist eine Eigenschaft namens `_alpha` des `MovieClip`-Objekts und wird in Prozent von 0 (keine Deckkraft) bis 100% (volle Deckkraft) angegeben.

Bevor Sie aber beginnen, werfen Sie einen kurzen Blick auf den Aufbau des Schiebereglers: Der Schieberegler ist ein `MovieClip`. In unserem Beispiel trägt er den Instanznamen `schieberegler_mc`. Ihm untergeordnet finden Sie einen eigenen `MovieClip` für den eigentlichen Regler (`transparenzregler_mc`). Dieser Regler muss bewegt werden und entscheidet dann mit seiner Position über die Transparenz des Chamäleons. Die Position des Transparenzreglers ist immer relativ zum Registrierungspunkt des übergeordneten Schiebereglers. Der Einfachheit halber haben wir in der Ausgangssituation den Registrierungspunkt des Schiebereglers genau am Ursprung des Transparenzreglers gewählt. Dadurch hat der Transparenzregler die Ausgangsposition `_x = 0` und `_y = 0`.

Deckkraft und Transparenz

Deckkraft ist das Gegenteil von Transparenz; hat ein Element 100% Deckkraft, besitzt es 0% Transparenz. Bei 0% Deckkraft ist ein Element unsichtbar und damit 100% transparent.



▲ **Abbildung 12.9**

Der Registrierungspunkt des Schiebereglers befindet sich im Ursprungspunkt des untergeordneten Transparenzreglers.



Schritt für Schritt: Schieberegler

Sollten Sie die letzten Schritte nicht praktisch nachvollzogen haben, finden Sie die Ausgangsdatei unter dem Namen CHAMAELEON_DRAGDROP_AS2.FLA auf der CD-ROM.

1. Zur Aktion für den Transparenzregler wechseln

Aktivieren Sie den MovieClip für den Transparenzregler und wechseln Sie in das AKTIONEN-Bedienfeld, um dort den Code einzutragen:

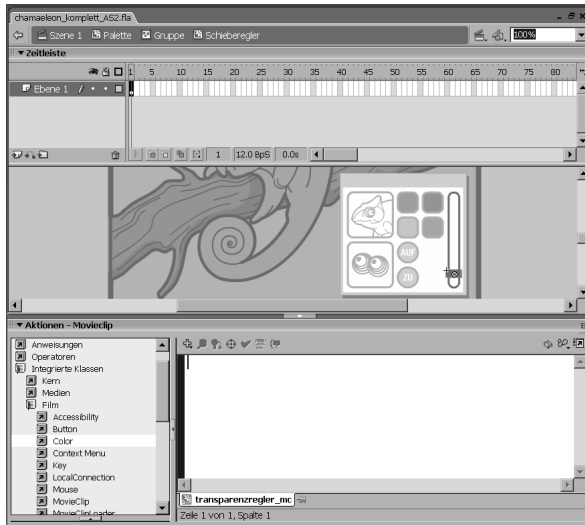


Abbildung 12.10 ►
Im AKTIONEN-Bedienfeld erkennen Sie unten, dass Sie sich beim Transparenzregler befinden.

Wir haben der Übersichtlichkeit halber auf den Einsatz von Ereignisprozeduren verzichtet. Wenn Sie trotzdem mit einer Ereignisprozedur arbeiten möchten, finden Sie den Transparenzregler unter der Adresse: `_root.palette_mc.schieberegler_mc.transparenzregler_mc`

2. Vorbereitungen treffen

Beim Laden des MovieClips mit dem Transparenzregler legen Sie einige Werte fest, die später bei Drag & Drop zum Einsatz kommen:

```
onClipEvent (load) {
    var regOben_num:Number = this._y - 100;
    var regUnten_num:Number = this._y;
    var regLinks_num:Number = this._x;
    var regRechts_num:Number = this._x;
}
```


`regOben_num` speichert die obere Begrenzung des Reglers und übernimmt als Wert die Ausgangskoordinate des Transparenzreglers (hier 0) minus 100. Das heißt, die obere Begrenzung liegt 100 Pixel über der aktuellen Position des Transparenzreglers. Die untere Begrenzung (`regUnten_num`) gibt die aktuelle Position an. Nach links und rechts darf der Transparenzregler nicht verschoben werden, deswegen sind beide Begrenzungen auf die aktuelle x-Koordinate gesetzt.

Fügen Sie als Nächstes das Ereignis `press` hinzu:

```
on (press) {
```

Damit fangen Sie das Tastaturreignis ab, wenn der Nutzer den Transparenzregler anklickt und Sie können Drag & Drop starten.

Öffnen Sie das Auge des Chamäleons mit folgendem Code:

```
_root.augezu_mc._visible = false;
```

Jetzt wird das Chamäleon immer »wach«, wenn der Nutzer die Deckkraft seiner Farben verändert. Außerdem müssen Sie so nicht die Deckkraft des Auges mit verändern.

Starten Sie Drag & Drop und verwenden Sie als Begrenzung die Werte, die Sie für den Transparenzregler vorbereitet haben:

```
startDrag(this, false, regLinks_num, regOben_num,
regRechts_num, regUnten_num);
}
```

Mit diesen Werten bewegt sich der Transparenzregler nur zwischen den y-Koordinaten 0 und -100, aber auf der x-Achse überhaupt nicht.

Drag & Drop beenden Sie, wenn der Nutzer die Maustaste loslässt:

```
on (release, releaseOutside) {
    stopDrag();
}
```

Zum Schluss ändern Sie die Transparenz, und zwar in Abhängigkeit von der aktuellen Position des Transparenzreglers. Hierzu verwenden Sie das Ereignis `enterFrame`:

3. Ereignis hinzufügen

4. Auge öffnen

5. Drag & Drop starten

6. Drag & Drop beenden

7. Transparenz regeln

```
onClipEvent (enterFrame) {
    _root.koerperfarbe_mc._alpha = (this._y -
regOben_num);
}
```

Zwei Dinge bedürfen hier der Erklärung: Zum einen wird `enterFrame` verwendet, da dieses Ereignis laufend eintritt. In unserem Fall wird bei diesem Ereignis laufend die Deckkraft geändert, und zwar auf die aktuelle Position des Transparenzreglers.

Zum anderen wollen wir erklären, wie sich die Deckkraft aus der Position des Schiebereglers berechnet. Wird der Regler nach oben verschoben, ist die Position negativ (z.B. -10); davon ziehen Sie die obere Begrenzung ab (minus 100). Aus der Rechnung $-10 - (-100)$ ergibt sich eine Deckkraft von 90%. Je weiter der Regler nach oben verschoben wird, desto niedriger ist die Deckkraft.

Ende



▲ **Abbildung 12.11**

Nach einem Klick auf den Transparenzregler geht das Auge des Chamäleons auf und der Hintergrund scheint durch.



Nun müssen Sie sich vom Chamäleon verabschieden. Sie finden es in seiner ganzen Pracht mit dem kompletten Code auf der CD-ROM unter `CHAMAELEON_KOMPLETT_AS2.FLA`. Als Besonderheit haben wir in der Variante `CHAMAELEON_EREIGNISPROZEDUREN_AS2.FLA` für Drag & Drop und für den Schieberegler auch Lösungen mit Ereignisprozeduren eingebaut. Im Ordner `FLASHMX` zum Kapitel finden Sie beide Versionen auch für ActionScript 1. Damit bieten wir Ihnen die Möglichkeit, selbst zu vergleichen, welche Variante Ihnen besser gefällt.

! 78
 != 77
 #b (Bildmarkierung) 140
 #include 48
 #p (Bildmarkierung) 142
 % 76
 %= 76
 && 78
 * 76
 */ 74
 *= 76
 + 76
 ++ 76
 += 76
 - 76
 -- 76
 -= 76
 / 76
 /* 74
 // 74
 /= 76
 < 77
 <= 77
 == 77
 > 77
 >= 77
 [] (Zugriffsoperator) 102
 _alpha (Eigenschaft) 225, 307
 _currentframe (Eigenschaft) 301
 _framesloaded (Eigenschaft) 322
 _frametotal (Eigenschaft) 322
 _global.styles 368
 _global (Eigenschaft) 74
 _height (Eigenschaft) 96, 157, 184
 _level (Schlüsselwort) 313
 _lockroot (Eigenschaft) 314, 317, 330
 _parent (Bildschirme) 130
 _parent (MovieClip) 41
 _root (MovieClip) 40
 _rotation (Eigenschaft) 191, 204
 _soundbuftime (Eigenschaft) 278
 _totalframes (Eigenschaft) 301
 _visible (Eigenschaft) 157, 218, 296
 _width (Eigenschaft) 157
 _xmouse (Eigenschaft) 247
 _xscale (Eigenschaft) 254
 _x (Eigenschaft) 181
 _ymouse (Eigenschaft) 247
 _yscale (Eigenschaft) 254
 _y (Eigenschaft) 184
 || 78

A

Absolut 40, 130
 ActionScript
 -Ebene 232
 Ereignisse
 enterFrame 37
 keyDown 37
 keyUp 37
 load 37
 mouseDown 37
 mouseMove 37
 mouseUp 37
 unload 37
 Export für 255, 279
 Strikte Typisierung 70
 Was lässt sich steuern? 38
 Wo? 34
 addListener() (Methode) 174, 248
 addPage() (Methode) 143
 Optionen 143
 Adressierung 40
 Absolut 40
 Bildschirme 130
 Level 313
 Relativ 40
 Schaltflächen 42
 AIFF 276
 Aktion 35
 Aktionen-Bedienfeld 34, 43
 Aktion fixieren 44
 Ansichtsoptionen 48
 Auto-Format 47
 Auto-Format-Optionen 49
 Codehinweis zeigen 47
 Debug-Optionen 48
 Ersetzen 47
 Hilfe 48
 Menü 48
 Referenz 48
 Schnellzugriff 45
 Skriptfeld 45
 Suchen 46
 Syntax überprüfen 47
 Werkzengleiste 46
 Zeilennummer 46
 Animieren 218
 in der Praxis 237
 mit ActionScript 218
 mit Zufall 235
 Arithmetische Operatoren 75

Array 69, 101
-Zugriffsoperator 102
assoziativ 105
bearbeiten 103
Datentypen 102
Index 102
in der Praxis 189
Methoden 103
mit Schleifen 104
multidimensional 106
multidimensionaler 387
Array() (Funktion) 73
AS-Kommunikationsdatei 61
ASCII 174
Assoziatives Array 105
attachMovie() (Methode) 255,
297, 304
attachSound() (Methode) 279, 280
Auswahlliste 346

B

backgroundColor (Eigenschaft) 156,
158
background (Eigenschaft) 156, 158
Balance 280
Bedienfelder
Aktionen 34, 43
Film-Explorer 41
Hilfe 49
Komponenten-Inspektor 60
Komponenten 60
Verhalten 27
Web Services 61
Zielpfad-Editor 40, 131
Bedingung 79
beginFill() (Methode) 231
beginGradientFill() (Methode) 232
Begrenzung
Drag & Drop 227
Bezeichner 66
bframe (Begrenzung) 140
Bibliothek 23
Auf Bühne exportieren 255
Video exportieren 297
Bildmarkierungen 116
Bildschirme 58, 128
Adressierung 130
Hierarchie 60, 130
Name 129
Übergänge 135

bmax (Begrenzung) 140
bmovie (Begrenzung) 140
Bogenmaß 202
Boolean() (Funktion) 73
Boolean (Datentyp) 69
borderColor (Eigenschaft) 158
border (Eigenschaft) 158
Breakpoints 54
break (Anweisung) 85
Bühne 24

C

case (Fallunterscheidung) 83
change (Ereignis) 340, 348, 384
Checkbox 343
Eigene 365
clearInterval() (Funktion) 163, 234
click (Ereignis) 345, 351
Client-Server-Prinzip 372
Codehinweise 70
eigene 71
zeigen 47
Color (Klasse) 218, 306
Computerlogik 211
Countdown 194
createEmptyMovieClip() (Methode)
230, 304
createTextField() (Methode) 165
CSS 160, 362, 367

D

data (Eigenschaft) 345, 347
data (Ereignis) 37
data (Parameter) 346
Datentyp 69
ändern 72
Endungen 72
feststellen 71
Date (Klasse) 97, 188
instanziiieren 188
Methoden 190
Datum 188
auf Deutsch 189
Daten voneinander abziehen 196
eigenes 190
Debugger (von Flash) 53
Debugging 51
Deckkraft 225
Deklarieren 64

Dekrement 75
 in der Praxis 162
 delete (Anweisung) 143, 159
 DirectX 292
 do-while (Schleife) 88
 Dokumentebene 313
 dragOut (Ereignis) 36
 dragOver (Ereignis) 36
 Drag & Drop 222, 270
 Begrenzung 227
 Drehregler 287, 288
 Drucken 140
 Begrenzung 140
 bestimmtes Bild 142
 Dialogfeld 143
 mit Schaltfläche 141
 duplicateMovieClip() (Methode) 250
 Duplizieren 250
 duration (Eigenschaft) 280
 DV (Digital Video) 292
 Dynamisches Textfeld 150

E

Ebene
 per ActionScript 232
 ECMAScript 20
 editable (Eigenschaft) 339, 341
 Eigenschaften 95
 Eingabefeld 150, 151
 Einstellungen (für ActionScript) 49
 else (Fallunterscheidung) 81
 else if (Fallunterscheidung) 80
 endFill() (Methode) 232
 Endlosschleife 88
 enterFrame (Ereignis) 37, 173, 227, 323
 zum Animieren? 234
 Ereignis-Sound 277
 Ereignisorientierung 34
 Ereignisprozedur 37, 123
 Ereignisse 34
 Schaltflächen- 36
 Erkennung des Flash-Player 331
 Event-Listener 174
 für die Maus 247
 für MovieClipLoader 317
 Expertenmodus (Aktionen-Bedienfeld) 43
 Export für ActionScript 22
 Externen Film
 Preloader 330
 Externer Code 48
 Externe Bilder 318
 Externe Filme 312
 in der Praxis 315
 in MovieClips laden 314
 Ladevorgang überwachen 317

F

Fallunterscheidungen 79
 Farbe 218
 ändern 219
 Transformation 306
 zufällig wählen 236
 Film-Explorer 41
 Filmsteuerung 112
 für MovieClips 121
 für Szenen 118
 Film testen 29, 52
 Flash
 Besonderheiten in MX 2004
 Professional 58
 für Mac 30
 für Windows 30
 MX 2004 Professional 128
 Projekt 61
 Sound einbinden 276
 Versionen 29
 Video integrieren 292
 Flash-JavaScript-Datei 61
 Flash-Player 55
 Erkennung 331
 FLV 293
 Folien
 Zugriff 128
 Folienpräsentation 58
 fontFamily (Eigenschaft) 368
 fontSize (Eigenschaft) 368
 fontStyle (Eigenschaft) 368
 fontWeight (Eigenschaft) 368
 for-in (Schleife) 105
 Formularanwendung 58
 in der Praxis 351
 Formulare 336
 Auswahlliste 346
 Checkbox 343
 eigene Formularelemente 363
 Kontakt- 375
 Kontrollkästchen 343
 Lieferdaten 391
 Mailversand 379

- Mehrzeiliges Textfeld 341
- Optionsfelder 344
- Radiobuttons 344
- Schaltflächen 350
- Textfelder 338
- Umfrage 351
- Umfrage ohne MX 2004
 - Professional 357
- Versenden 372
- Vollständigkeitsüberprüfung 353
- Zugriffskontrolle 366
- for (Schleife) 86, 104
- fromCharCode() (Methode) 176
- fscommand() (Funktion) 136
- function (Schlüsselwort) 89
- Funktionen 89
 - Aufruf 90
 - in der Praxis 209
 - mehrere Rückgaben 92
 - Parameter 91
 - rekursiv 94
 - Rückgabewert 92
 - von ActionScript 95
 - Globale 95

G

- GET 373
- getAscii() (Methode) 175
- getBytesLoaded() (Methode) 280, 322
- getBytesTotal() (Methode) 280
- getCode() (Methode) 175
- getDate() (Methode) 189
- getDay() (Methode) 189
- getFullYear() (Methode) 189
- getMonth() (Methode) 189
- getPan() (Methode) 280
- getRGB() (Methode) 219
- getTransform() (Methode) 280
- getURL() (Funktion) 373
- getVolume() (Methode) 280
- Globale Funktionen 95
- Globale Variablen 74
- globalToLocal() (Methode) 260
- GMT 188
- gotoAndPlay() (Filmsteuerung) 112, 115
- gotoAndStop() (Filmsteuerung) 115, 118, 136, 295
- gotoNextSlide() (Methode) 129
- gotoPreviousSlide() (Methode) 129

- gotoSlide() (Methode) 134
- Grafik-Symbol 21
- groupName (Eigenschaft) 345
- Gültigkeit 73

H

- hasAudioEncoder (Eigenschaft) 281
- hasAudio (Eigenschaft) 281
- hasMP3 (Eigenschaft) 281
- Hexadezimal 219
 - Umwandlung 236
- hide() (Methode) 247
- hideChild (Ereignis) 135
- hide (Ereignis) 135
- Hierarchie
 - MovieClips 40
- Hilfe 49
- hitArea (Eigenschaft) 265
- hitTest() (Methode) 260
- hscroll (Eigenschaft) 162
- htmlText (Eigenschaft) 156
- html (Eigenschaft) 158, 341

I

- icon (Eigenschaft) 351
- ID3 284
- if (Fallunterscheidung) 79
- Index 102
- Initialisieren 65
- Inkrement 75
 - in der Praxis 162
- Instanz 23
 - name 39
 - Textfeld 155
- Instanzieren 97
- isDown() (Methode) 173
- italic (Eigenschaft) 158

J

- JavaScript 20
- JPEG
 - per ActionScript laden 318
- JScript 20

K

- Key.DOWN (Eigenschaft) 184
- Key.ENTER (Eigenschaft) 173

Key.LEFT (Eigenschaft) 183
 Key.UP (Eigenschaft) 184
 keyDown (Ereignis) 37, 174
 keyPress (Ereignis) 36, 168
 keyUp (Ereignis) 37, 340
 Key (Klasse) 168, 173
 Kilobyte 325
 Klasse 96
 Kollisionstest
 hitTest() 260
 in einer Anwendung 266
 per Hand 260
 Kommentare 74
 Komponenten 60
 anpassen 362
 Auswahlliste 383
 Button 350
 CheckBox 343
 List 347
 RadioButton 345
 TextArea 161, 341
 TextInput 338
 UI- 336
 Komponenten-Bedienfeld 60
 Komponenten-Inspektor 60
 Komponenten;anpassen 367
 Konkatenieren 100
 Konstanten 200
 Konstruktor 97
 Kontrollstrukturen 79
 Koordinaten, zufällige wählen 235
 Koordinatensystem 24
 lokal und global 260
 Ursprung 24
 Kosinus 201

L

labelPlacement (Eigenschaft) 343,
 345, 351
 labels (Eigenschaft) 347
 label (Eigenschaft) 343, 345, 351
 length (Eigenschaft) 103
 Level 313
 absolute Adressierung 313
 Zugriff 313
 LineStyle() (Methode) 231
 lineTo() (Methode) 231
 load() (Methode) 160
 loadClip() (Methode) 317
 loadClip (Ereignis) 317
 loadMovie() 318
 loadMovie() (Methode) 312
 loadMovieNum() (Methode) 313
 loadSound() (Methode) 280
 loadVariables() (Funktion) 373
 load (Ereignis) 37, 226
 localToGlobal() (Methode) 260
 Logische Operatoren 77
 Lokale Variablen 73
 Loops 277

M

Mac OS X 30
 Mailversand 379
 Math.abs() (Methode) 200
 Math.ceil() (Methode) 206
 Math.cos() (Methode) 202
 Math.exp() (Methode) 200
 Math.E (Konstante) 200
 Math.floor() (Methode) 206
 in der Praxis 196
 Math.log() (Methode) 201
 Math.max() (Methode) 206
 Math.min() (Methode) 206
 Math.PI (Konstante) 200
 Math.pow() (Methode) 201, 263
 Math.random() (Methode) 206
 Math.round() (Methode) 206
 Math.sin() (Methode) 202
 Math.tan() (Methode) 202
 Math (Klasse) 200
 Konstanten 200
 Maus 246
 -bewegungen 246
 verfolgen 269
 maxhscroll (Eigenschaft) 162
 maxscroll (Eigenschaft) 162
 Methoden 95
 mouseDown (Ereignis) 37, 246
 mouseMove (Ereignis) 37, 246
 mouseUp (Ereignis) 37, 246
 Mouse (Klasse) 247
 moveTo() (Methode) 231
 MovieClip 21, 35
 Adressierung 39
 als Schaltfläche 35
 aus Bibliothek laden 255
 Deckkraft 225
 duplizieren 250
 Ereignisse 36

- Filmsteuerung für 121
- für Animationen 218
- Hierarchie 40
- leeren erzeugen 230
- SWF laden 314
- umfärben 220
- Zugriff 39
- MovieClipLoader (Klasse) 317
- MovieClip (Klasse) 97, 165, 191, 218, 247, 250
- MP3 276
- MPEG 292
- Multidimensionales Array 106
- multiline (Eigenschaft) 158
- multipleSelection (Eigenschaft) 347

N

- new (Konstruktor) 97
- nextFrame() (Filmsteuerung) 114, 136
- nextScene() (Filmsteuerung) 119
- Normalmodus (Aktionen-Bedienfeld) 43
- Null (Datentyp) 69
- Number() (Funktion) 73, 195
- Number (Datentyp) 69

O

- Object() (Funktion) 73
- Object (Datentyp) 69
- Object (Klasse) 174, 248, 260
- Objekte instanziiieren 97
- onClipEvent (Schlüsselwort) 35, 235
- onEnterFrame (Ereignisprozedur) 204, 262
 - zum Animieren? 234
- onKeyDown (Ereignisprozedur) 174
- onLoadComplete (Ereignisprozedur) 317
- onMouseMove (Ereignisprozedur) 246
- onRelease (Ereignisprozedur) 123
- onSoundComplete (Ereignisprozedur) 284
- on (Schlüsselwort) 35
- Operatoren 75
 - Arithmetische 75
 - Logische 77
 - Vergleichs- 76
- Optionsfelder 344

P

- Parameter 91, 96
- parseInt() (Methode) 388
- password (Eigenschaft) 158, 339
- Passwortfelder 339
- Passwort prüfen 366
- Pendelbewegung 203
- PHP 372
- play() (Filmsteuerung) 114, 296
- play() (Methode) 96
- pop() (Methode) 104
- position (Eigenschaft) 280
- POST 373
- Präsentation
 - mit Flash MX 2004 Pro 133
 - ohne Pro 135
- Preloader 322
 - als externer Film 330
 - Fortschrittsbalken 324
 - Ladegeschwindigkeit berechnen 329
 - mit Bildern 323
 - mit Bytes 325
 - Restzeit berechnen 327
- press (Ereignis) 36, 224
- prevFrame() (Filmsteuerung) 114, 136
- prevScene() (Filmsteuerung) 119
- print() (Funktion) 140
- printAsBitmap() (Funktion) 141
- printAsBitmapNum() (Funktion) 141
- PrintJob (Klasse) 143
- printNum() (Funktion) 141
- push() (Methode) 103

Q

- QuickTime 276, 292

R

- Radiobuttons 344
 - eigene 363
- random() (Funktion, veraltet) 207
- Registrierungspunkt 22, 25, 203
- Regler 225, 287
- Rekursive Funktionen 94
- Relativ 40, 130
- releaseOutside (Ereignis) 36, 224
- release (Ereignis) 36, 224

removeListener() (Methode) 248
 removeMovieClip() (Methode) 254
 replaceText() (Methode) 159
 return (Schlüsselwort) 92
 revealChild (Ereignis) 135
 reveal (Ereignis) 135
 reverse() (Methode) 104
 rollOut (Ereignis) 36, 163
 rollOver (Ereignis) 36, 163
 rowHeight (Eigenschaft) 347
 Rückgabewert 92

S

Schaltfläche 21, 35
 als Komponente 350
 eine für mehrere Zustände 300
 eine für zwei Zustände 297
 Ereignisse 36
 mit ActionScript hervorheben 295
 Zustand 36
 Schieberegler 225
 für Lautstärke 282
 horizontal 302
 Registrierungs­punkt 225
 Schleifen 86
 Schlüsselbild 34
 Schlüsselwort 64
 Übersicht aller 67
 Scrollen, horizontal 162
 scroll (Eigenschaft) 162
 selectable (Eigenschaft) 158
 selectedData (Eigenschaft) 346
 selectedIndex (Eigenschaft) 347
 selectedIndices (Eigenschaft) 349
 selectedItem (Eigenschaft) 347
 selected (Eigenschaft) 343, 345, 351
 send() (Methode) 143
 set() (Funktion) 73
 setInterval() (Funktion) 163, 192, 338
 zum Animieren? 234
 setMonth() (Methode) 190
 setNewTextFormat() (Methode) 159
 setPan() (Methode) 280
 setProperty() (Funktion) 253
 setRGB() (Methode) 219
 setStyle() (Methode) 160, 368, 386
 setTextFormat() (Methode) 159
 setTimeout() (JavaScript) 234
 setTransform() (Methode) 280, 306
 setVolume() (Methode) 280
 shift() (Methode) 104
 show() (Methode) 247
 Sinus 201
 Verlauf 202
 size (Eigenschaft) 158
 skip intro 382
 Sonderzeichen 99
 Sorenson Sparc-Codec 293
 Sound 276
 Balance 280, 287
 Ereignis- 277
 Export für ActionScript 279
 extern 283
 Formate 276
 hinzufügen 277
 Lautstärke 280
 Loops 277
 mehrere Sounds 284
 mit Verhalten 277
 MP3 laden 280
 Schleifen 277
 steuern 278
 Streaming- 277
 transformieren 280
 soundComplete (Ereignis) 284
 SoundDesigner II 276
 Sound (Klasse) 279, 301
 Methoden und Eigenschaften 281
 Spiele
 17 und 4 208
 Computerlogik 211
 Friseur 268
 Kollisionstest für 260
 split() (Methode) 195
 sqrt() (Methode) 201
 start() (Methode) 143, 279, 280
 startDrag() (Methode) 222
 Statisches Textfeld 150
 Stile
 in der Praxis 386
 stop() (Filmsteuerung) 113, 295
 stop() (Methode) 280, 301
 stopAllSounds() (Funktion) 278
 stopDrag() (Methode) 223
 Streaming-Sound 277
 Strikte Typisierung 70
 für Funktionen 91
 String
 Besonderheiten 98
 mehrere zusammenfügen 100
 vergleichen 101

String() (Funktion) 73
String (Datentyp) 69
String (Objekt) 98
styleSheet (Eigenschaft) 161
StyleSheet (Klasse) 160
Style Sheets 362, 367
Sun AU 276
SWF 28
 laden 312
 vorladen 322
SWiSH 29
switch-case (Fallunterscheidung) 83
 in der Praxis 236
Symbole 21
 Export für ActionScript 22
 In Symbol konvertieren 21
System.capabilities 281
System7 276
Szenen 117
 Filmsteuerung 118

T

Tangens 202
target-Attribut (HTML) 374
Tastensteuerung 168
 Ereignis 168
 Mehrere Tasten abfangen 178
 Probleme beim Testen 172
Testen 52
textColor (Eigenschaft) 157
Textfelder 150
 Datentyp 151
 dynamisch 150
 dynamisch erzeugen 165
 Eingabe- 150
 formatieren 155
 Inhalte bearbeiten 152
 mehrzeilig mit Komponente 341
 mit Komponente 338
 mit MovieClip-Eigenschaften 157
 mit Rahmen und Hintergrund 156
 scrollen 161
 statisch 150
 Variable festlegen 150
TextField (Klasse) 156
TextFormat (Klasse) 157
textHeight (Eigenschaft) 157
Textwerkzeug 150
textWidth (Eigenschaft) 157
text (Eigenschaft) 156, 339, 341

this (Schlüsselwort) 41, 123, 251
Tiefe 232
toggle (Eigenschaft) 351
toString() (Methode) 72, 196
trace() (Anweisung) 53
 beim Veröffentlichen ignorieren 53
Transparenz 225
Trigonometrie 201
 Einheitskreis 201
 Koordinatensystem 202
Tweens 218
type (Eigenschaft) 155

U

Übergänge 135
Uhrzeit 188
 mit Uhrzeigern 191
UI-Komponenten 336
Umfärben 220
Umwandlungsfunktionen 95
Undefined (Datentyp) 69
undefined (Wert) 65
Unix-Zeitstempel 194
unloadMovie() (Methode) 314
unload (Ereignis) 37
unshift() (Methode) 104
Ursprung 24
Usability 336, 362
UTC 188

V

Variable 64
 Datentyp 69
 deklarieren 64
 für ein Bild 120
 für Textfeld 150
 global 74
 Gültigkeit 73
 Gültigkeitsbereich in der Praxis 93
 initialisieren 65
 Kurzformen 73
 lokal 73
 Namen dynamisch bilden 252
 Zeitleisten- 73
Variablenamen 66
 Konventionen 68
var (Schlüsselwort) 64
Vergleichsoperatoren 76
Verhalten 27

- für Bildschirme 131
- für Sound 277
- für Videos 293
- Veröffentlichen 28
 - Einstellungen für Veröffentlichungen 29
 - ohne trace() 53
- Versenden von Formularen 372
- Video 292
 - aus Bibliothek 297
 - dynamisch verändern 303
 - einbetten 292
 - Ende überprüfen 301
 - Farbeändern 306
 - Format ändern 305
 - in Flash 292
 - mit ActionScript 294
 - mit Soundsteuerung 301
 - mit Verhalten 293
 - positionieren 297
 - steuern 294
 - transparent 307
 - verknüpfen 292
- Video (Container) 304
- Voreinstellungen (für ActionScript) 50

W

- Warenkorb 382
 - Artikelseiten 383
 - Bestätigung 391
 - Funktionalität 387
- WAV 276, 292
- Webserver 372
- Web Services 60
- Web Services-Bedienfeld 61
- while (Schleife) 87
- Windows 30
- Windows Media 292
- with () (Anweisung) 180, 232
- wordWrap (Eigenschaft) 341

Z

- Zählervariable 86, 262
- Zeichnen mit ActionScript 230
- Zeichnungsmethoden 230
- Zeitleiste 20
- Zeitleisten-Variablen 73
- Zielpfad-Editor 40, 131
- Zufallszahl 206
 - zum Animieren 235
 - zur Spielentwicklung 208
- Zugriffsoperator 102
- Zustand (von Schaltflächen) 21