

CARL HANSER VERLAG

Ralf Siegel

Programmierhandbuch ActionScript

Leitfaden für die Programmierung von Flash MX-Anwendungen

3-446-22116-6

www.hanser.de



Inhalt

1	Die etwas andere Einführung	19
1.1	Aufwärmphase	19
1.1.1	Trockenübungen: Was ist ein Programm?	19
1.2	Sprung ins kalte Wasser	21
1.2.1	Ein erstes ActionScript-Programm	21
1.2.2	Kommentare	22
1.2.3	Ausgabefenster	23
1.2.4	Zeilenabschluss	23
1.3	Abtrocknen	23
2	Ausloten der Entwicklungsumgebung	27
2.1	Expertenmodus	27
2.2	Formatoptionen	28
2.3	Voreinstellungen	28
2.4	Codehinweise	29
2.5	Referenz	30
3	Variablen	33
3.1	Was sind Variablen?	33
3.2	Deklaration	33
3.2.1	Deklaration einer Variablen	33
3.2.2	Deklaration mehrerer Variablen	35
3.3	Initialisierung	35
3.3.1	Die speziellen Werte undefined und null	36
3.3.2	Initialisierung mit einer Zuweisungskette	37
3.4	Bezeichner	38
3.4.1	Konventionen	38
3.4.2	Groß- und Kleinschreibung	39
3.4.3	Erlaubte Zeichen	39
3.4.4	Reservierte Wörter (keywords)	40

3.5	Gültigkeitsbereich und Lebensdauer von Variablen	40
3.5.1	Lokale Variablen	40
3.5.2	Zeitleistenvariablen	41
3.5.3	Globale Variablen	45
3.5.4	Bereichskette	46
3.5.5	Vergleichstabelle	48
3.6	Textfeld-Variablen	48
3.7	Vordefinierte Variablen	49
3.7.1	\$version	49
3.7.2	Definition über die URL	50
3.7.3	Definition über die FlashVars Eigenschaft	50
3.8	Variablenaustausch zur Laufzeit	50
3.9	Beispiel	51
4	Datentypen und Operatoren	55
4.1	Die Notwendigkeit von Datentypen	55
4.2	Übersicht zu den Datentypen in ActionScript	56
4.3	Die einfachen Datentypen, Literale und Operatoren	57
4.3.1	String (Zeichenketten)	57
4.3.1.1	String-Literale	58
4.3.1.2	Operatoren für den Datentyp string	60
4.3.2	Boolean (Wahrheitswerte)	61
4.3.2.1	Logische Operatoren	62
4.3.3	Number (Zahlen)	63
4.3.3.1	Zahlenliterale	63
4.3.3.2	Gleitkomma-Genauigkeit	64
4.3.3.3	Konstanten und spezielle Werte	65
4.3.3.4	Operatoren für den Datentyp number	66
4.4	Rangfolge der Operatoren	72
5	Funktionen	77
5.1	Wieso Funktionen grün sind	77
5.2	Eigene Funktionen definieren und aufrufen	78
5.3	Funktionen mit Parametern	80
5.4	Funktionen mit Rückgabewerten	82
5.5	Funktionen mit variabler Anzahl von Parametern	83
5.6	Das Objekt Arguments – callee & caller	85
5.7	Mit rekursiven Funktionen ans Limit	86
5.8	Unheimliches aus der Wunderlampe	88
5.8.1	Einen Verweis auf sich selbst zurückgeben	88
5.8.2	Funktionsdeklaration mit einer bedingten Anweisung	88

5.8.3	Aufruf vor Deklaration.....	89
5.8.4	Funktionen ohne Bezeichner (anonyme Funktionen).....	91
5.9	Anmerkungen zu vordefinierten Funktionen.....	91
5.9.1	escape/unescape.....	91
5.9.2	eval().....	92
6	Bedingte Anweisungen.....	95
6.1	Wer stellt hier die Bedingungen?.....	95
6.2	Verzweigungen mit der <i>if</i> -Anweisung.....	95
6.3	Auswahl mit der <i>switch</i> -Anweisung.....	100
6.4	Der <i>?:</i> Operator.....	104
7	Schleifen.....	107
7.1	Jetzt wird Achterbahn gefahren.....	107
7.2	Die <i>while</i> -Schleife.....	107
7.3	Die <i>do-while</i> -Schleife.....	112
7.4	Die <i>for</i> -Schleife.....	113
7.5	Die <i>for-in</i> -Schleife.....	117
8	Objekte und Klassen.....	123
8.1	Sind Objekte der heilige Gral von Flash?.....	123
8.2	Haben Objekte auch wirklich Klasse?.....	126
8.2.1	Konstruktor.....	126
8.2.2	Überblick über vordefinierte Klassen in ActionScript.....	127
8.2.3	Instanzen mit dem Schlüsselwort <i>new</i> erzeugen.....	128
8.2.4	Konstruktor-Funktion mit Argumenten.....	129
8.2.5	Eigenschaften und Methoden.....	130
8.2.6	Zugriff auf Eigenschaften und Methoden.....	131
8.2.6.1	Der gepunktete Weg – die DOT-Syntax.....	131
8.2.6.2	Mit allen gleichzeitig – die Aktion <i>with</i>	132
8.2.6.3	Zugriff auf alles – der Zugriffsoperator [].....	132
8.2.7	Strenge Verweise, oder der Letzte knipst das Objekt aus.....	133
8.2.8	Überblick über vordefinierte Objekte in ActionScript.....	137
8.3	Die eigene Klasse.....	138
8.3.1	Das Schlüsselwort <i>this</i>	140
8.3.2	Die Vorlage bekommt Methoden.....	141
8.3.3	Die Eigenschaft <i>prototype</i>	143
8.3.4	Funktionen sind auch Objekte.....	144
8.3.5	Die Eigenschaft <i>__proto__</i>	145

8.3.6	Feuerwehr – Löschen von Objekten mit einer Objektmethode.....	148
8.3.7	Die ideale Klasse – eine Zusammenfassung für die Praxis	150
8.4	Eine Klasse erweitern – Hopp oder OOP?.....	151
8.4.1	Prototype-basierte Vererbung	152
8.4.2	Überschreiben von Eigenschaften und Methoden.....	154
8.4.3	Macromedia hebt den Finger: new vs. <code>__constructor__</code>	155
8.4.4	Der <i>super</i> -Operator	157
8.4.5	Schweres Erbe – eine Zusammenfassung	161
8.5	Bestehende Klassen erweitern	161
8.6	Klassen mit Grafik-Symbolen assoziieren.....	162
8.6.1	Das Traumhaus	163
8.6.2	Registrierung mit <code>Object.registerClass()</code>	164
8.6.3	Es kommt Leben in die vier Wände – Methoden und Eigenschaften.....	165
8.6.4	Richtfest – das Init-Objekt.....	166
8.6.5	Zwischen <code>#initclip</code> und <code>#endinitclip</code>	167
8.6.6	Vererbungs-Strategie für visuelle Klassen.....	169
9	Ereignisse.....	177
9.1	Was geht denn hier ab?	177
9.2	Große Ereignisse werfen ihre Schatten voraus.....	178
9.2.1	Eine, die immer redet	178
9.2.2	Ereignis-basierte Ausführung.....	179
9.2.3	Stars und Journalisten – Ereignisquellen und Empfänger	180
9.3	Listener	181
9.3.1	Zuhörer gewinnen – <code>addListener()</code>	181
9.3.2	Zuhörer verlieren – <code>removeListener()</code>	184
9.3.3	Zuhörer-Zombies	185
9.4	Sport frei – Zuhörer in Aktion	187
9.4.1	Bob anschieben – Key Listener.....	188
9.4.2	Mäuse machen – Mouse Listener	190
9.4.3	Shopping mit Zuhörern – TextField Listener	193
9.4.4	Gut gewählt – Selection Listener	196
9.4.4.1	Alte Schule: Fokus-Ereignisse im Spiegel betrachtet	196
9.4.4.2	Neue Schule: Selection Listener.....	203
9.4.4.3	Hohe Schule: Selection – Textauswahl.....	205
9.4.5	Bühne frei – Stage Listener.....	207
9.4.6	Geheime Botschaften – <code>broadcastMessage</code>	209
9.4.7	Vorsorgen für die Zukunft – <code>ASBroadcaster</code>	210
9.5	Was sonst noch passiert und passieren kann	212
9.5.1	Ruhe in Frieden: <code>onClipEvent()</code> und <code>on()</code>	213
9.5.2	Vertrauter Käse für Movieclips und Buttons – Mausereignisse.....	213

9.5.3	Vorsicht Falle – Gültigkeitsbereich und <code>_parent</code>	215
9.5.4	Bereiche an der Kette – Scope Chain	216
9.5.5	Aktiv, Aktiviert, Aktivierungs-Objekt	219
9.5.6	Frisch machen für eigenen Mauszeiger – <code>updateAfterEvent()</code>	222
9.5.7	Im Gleichtakt – Intervalle	223
9.5.7.1	<code>setInterval()</code>	223
9.5.7.2	Intervall-ID und <code>clearInterval()</code>	224
10	Filmsequenzen	229
10.1	Film ab	229
10.2	Die Geburt eines MovieClip-Objekts	230
10.2.1	Das Initialisierungsobjekt	232
10.2.1.1	Anhänglich: <code>attachMovie()</code>	232
10.2.1.2	Doppelgänger: <code>duplicateMovieClip()</code>	233
10.2.2	An der Leine – Referenz zur neuen Instanz	235
10.3	Tiefenmessung	236
10.4	Plätze tauschen mit Drag'n'Drop	239
10.4.1	<code>swapDepth()</code>	240
10.4.2	<code>startDrag()</code> und <code>stopDrag()</code>	242
10.4.3	<code>_droptarget</code>	244
10.5	Maskierter Täter	245
10.6	Farbige Aussichten	246
10.6.1	Color-Klasse	246
10.6.2	Alles wieder zurück	249
10.7	Vorsicht Schieber	250
10.7.1	Klasse Knopf mit Begrenzung	250
10.7.2	Aktiver Knopf	251
10.7.3	Klasse Slider mit gefährlichen Ecken	252
10.7.4	Passive Krönung – Listener-Schnittstelle einschieben	255
10.7.5	Stellung einnehmen	256
10.8	Malbretter	258
10.8.1	Linien zeichnen	258
10.8.2	Überlappende Füllungen	259
10.8.3	Gefährliche Kurven	261
10.9	Anecken – Squash	262
10.9.1	<code>getBounds()</code>	262
10.9.2	<code>hitTest()</code>	264
10.9.3	Das etwas andere Menü mit <code>hitTest()</code>	265
10.9.4	Kollisionserkennung mit <code>getBounds()</code> und <code>hitTest()</code>	266
10.9.5	Bildeintritt vs. Intervall-Auftritt	271
10.9.6	Game over	271

10.10	Nachladen	272
10.10.1	Ärger mit onData()/onLoad().....	273
10.10.2	Ladestapel mit getBytesLoaded()/getBytesTotal()	274
10.10.3	Fortschrittsanzeige mit Tücken	277
10.10.4	Bilder laden	278
10.10.5	Nicht gefunden.....	278
11	Datenaustausch.....	281
11.1	Völlig losgelöst.....	281
11.2	Externe Dateien und Unicode	281
11.3	LoadVars – Austausch über Variablen	282
11.3.1	Daten laden.....	283
11.3.2	Alles da – Daten anzeigen	285
11.3.3	Ein Klasse-Ticker.....	286
11.3.4	Formular abschicken	289
11.3.4.1	Der ActionScript-Teil.....	289
11.3.4.2	ASP / JScript.....	294
11.3.4.3	PHP.....	295
11.3.4.4	CFML.....	295
11.3.4.5	JSP.....	296
11.3.5	Content-Length: Lange Nase mit getBytesTotal().....	296
11.3.5.1	ASP/JScript.....	297
11.3.5.2	PHP.....	297
11.3.5.3	CFML.....	297
11.3.5.4	JSP	298
11.4	XML – Tauschen mit allen.....	298
11.4.1	DOM – Ausloten des XML-Objekts.....	298
11.4.1.1	Ding – Erzeugen eines XML-Objekts.....	299
11.4.1.2	Dong – Anhängliche Elemente und Textknoten.....	303
11.4.1.3	Peng – Knoten entfernen	305
11.4.1.4	Analogie XML DOM und Flash DOM.....	306
11.4.2	Daten laden.....	306
11.4.3	Alles da – Daten anzeigen	308
11.4.4	Schleifen lassen – der XML-Ticker.....	310
11.4.5	Rekursive Selbstfindung – XML in ein Array transformieren.....	313
11.4.6	Schnellzugriff mit dem Attribut „id“	315
11.4.7	XML Parser – Fehlerbehandlung	316
11.4.8	Achtung Umleitung – XML Newsfeed.....	319
11.4.8.1	ASP/JScript.....	321
11.4.8.2	PHP.....	321
11.4.8.3	CFML.....	321

11.4.8.4	JSP	322
11.4.8.5	Die ausgedienten Redirect-Skripte	322
11.4.9	Snapshot abschicken	323
11.4.9.1	Der ActionScript-Teil	323
11.4.9.2	ASP/JScript	327
11.4.9.3	PHP	327
11.4.9.4	CFML	328
11.4.9.5	JSP	329
11.5	Mixed Pickles – Variablen streuen, XML ernten	329
11.5.1	Der ActionScript-Teil	329
11.5.2	ASP/JScript	331
11.5.3	PHP	332
11.5.4	CFML	332
11.5.5	JSP	333
11.6	Flash Remoting	334
11.6.1	Zufalls-Service – einfache Werte anfordern	335
11.6.2	Info-Service – komplexe Objekte anfordern	337
11.6.3	Auswahl-Service – XML-Objekt laden	339
11.6.4	SOAP – Webservice konsumieren	342
11.6.4.1	Währung – Umtauschraten	342
11.6.4.2	Babelfish – Übersetzen	344
11.6.4.3	Google	345
11.6.5	Bumerang – FLASH Scope in ColdFusion	349
11.6.6	SSAS – Server Side ActionScript	351
11.6.6.1	Klassen-Vergleich	351
11.6.6.2	CF.http()	353
11.6.6.3	CF.query() – RecordSet-Objekte	354
11.7	XMLSocket – Austausch in Echtzeit	357
12	Audio und Video	361
12.1	Mit Pauken und Trompeten	361
12.2	Sound	361
12.2.1	Das Sound-Objekt abhören	362
12.2.1.1	Klasse – ein erstes ernsthaftes Gespräch	362
12.2.1.2	Einer nach dem anderen – onSoundComplete()	363
12.2.1.3	Meisterhaft – Globale Sounds und andere Kanäle	365
12.2.1.4	Organisation – Virtuelles Mischpult-Objekt	367
12.2.1.5	Wissenswert – Laden eines SWF in ein anderes Level	368
12.2.1.6	Globalisierung – im Sinne von Planet Erde	370
12.2.1.7	Ruhe	371
12.2.2	Sound Loops	374

12.2.2.1	Organisation	374
12.2.2.2	Start-Synchronisation	376
12.2.2.3	Einblenden und Ausblenden	377
12.2.3	Externe MP3-Dateien laden	378
12.2.3.1	Streaming	378
12.2.3.2	Event-Sounds.....	379
12.2.3.3	Lade- und Positionsanzeige	380
12.2.3.4	Streamen und Abspielen.....	385
12.2.3.5	Playliste mit XML-Speisung.....	386
12.2.4	ID3-Tag für MP3-Dateien auslesen.....	387
12.3	Mikrofon.....	388
12.3.1	Alle mal herhören.....	388
12.3.2	Pegelanzeige.....	389
12.4	Kamera.....	390
12.4.1	Alle mal herschauen!.....	390
12.4.2	Bewegungsmelder.....	391
12.5	Video.....	392
13	Komponenten.....	395
13.1	Bauklötze staunen.....	395
13.2	Die einfachste Komponente der Welt.....	396
13.2.1	Ein bekanntes Skript.....	396
13.2.2	Ein eigenes Icon.....	398
13.3	WordChecker – Das Rundum-Sorglos-Paket.....	399
13.3.1	_targetInstanceName und horizontal Eigenschaften.....	399
13.3.2	Die Komponente entwerfen.....	400
13.3.3	Der erste Ansatz.....	401
13.3.4	addProperty() und Get/Set-Eigenschaften.....	403
13.3.4.1	Das Problem.....	404
13.3.4.2	Das Get/Set-Prinzip.....	406
13.3.4.3	Get/Set-Eigenschaften am Prototype der Klasse.....	409
13.3.4.4	Get/Set-Ausgangswerte im Prototype definieren.....	412
13.3.5	Zugriffskontrolle, Optimieren und Verstecken.....	413
13.3.6	Live-Vorschau und das xch-Austauschobjekt.....	415
13.3.7	Ab ins Körbchen.....	417
13.3.8	Verpacken für Exchange: die MXI-Datei-Definition.....	418
13.3.9	Custom Actions.....	420
13.4	Andere Komponenten.....	423

14	Lokale Spezialitäten.....	427
14.1	Shared Objects – Cookies-Ersatz.....	427
14.1.1	Eigenschaften, Methoden und Ereignisse.....	428
14.1.2	SharedObject-Objekt ausloten.....	428
14.1.3	Datum des letztes Besuchs ermitteln.....	429
14.1.4	Benutzer-Rollenspiel.....	430
14.1.5	Löschzug.....	432
14.2	Local Connection – JavaScript-Zahnersatz.....	433
14.2.1	Eigenschaften, Methoden und Ereignisse.....	433
14.2.2	LocalConnection-Objekt ausloten.....	434
14.2.3	Ping – auf Sendung!.....	434
14.2.4	Schluss – Aus.....	436
15	Käfersuche.....	439
15.1	Krabbeln ja, aber nicht hier.....	439
15.2	Syntaxfehler und Warnmeldungen (Compiler).....	440
15.2.1	Syntaxfehler.....	440
15.2.2	Warnungen.....	441
15.3	Laufzeitfehler (Interpreter).....	442
15.4	Logische Fehler.....	443
15.4.1	Top 10 – die hartnäckigsten Käfer.....	443
15.4.1.1	Zuweisungsoperator statt Vergleichsoperator.....	443
15.4.1.2	Funktionsausführung statt Zuweisung einer Referenz.....	444
15.4.1.3	Verwendung falscher Datentypen/unerwartete Typenumwandlung.....	445
15.4.1.4	Fehlende Angabe von <i>this</i> in einem Event Handler.....	445
15.4.1.5	Der Einser-Wahnsinn.....	446
15.4.1.6	Das erste Element eines Arrays bekommt eine 0.....	446
15.4.1.7	Sortieralbtraum.....	447
15.4.1.8	Alles verdreht nach <code>Array.reverse()</code>	448
15.4.1.9	Schleifendurchlauf bringt zu viel ans Licht.....	448
15.4.1.10	Wer warten kann, ist klar im Vorteil.....	449
15.5	Player-, Plattform- und Performance-Probleme.....	449
15.6	Sicherheitsmängel.....	450
15.7	Kammerjägerausrüstung.....	450
15.7.1	<code>trace()</code>	450
15.7.2	Variablen/Objekte auflisten.....	451
15.7.3	Snapshot von Objekten.....	451
15.7.4	Geschwindigkeitsvergleiche.....	452
15.7.5	FLASM.....	452

Anhang	455
A Werkzeugkasten	455
A.1 #include – Externe Skripte einbinden.....	455
A.2 ASSetPropFlags() – Attribute für Eigenschaften setzen.....	456
A.3 ASNative() – die interne Verdrahtung.....	457
A.4 Function.extends().....	458
A.5 Reguläre Ausdrücke.....	458
A.6 WDDX.....	458
A.7 Skript-Schnipsel für alle Lebenslagen.....	458
Index	459