



# Das JavaScript Handbuch

➤ Einführung, Praxis und Referenz

# Inhalt

<b>Vorwort</b> .....	23
Über den Autor .....	24
<b>Teil I Elementare Grundlagen</b> .....	30
<b>1 Einleitung</b> .....	31
1.1 Über das Buch .....	32
1.1.1 An wen sich das Buch wendet und was Sie schon wissen sollten	32
1.1.2 Wie dieses Buch organisiert ist .....	33
1.1.3 Was Sie in dem Buch lernen .....	34
1.1.4 Schreibkonventionen .....	35
1.1.5 Die Quellcodes .....	35
1.2 Was ist JavaScript? .....	36
1.2.1 Die Geschichte von JavaScript .....	37
1.2.2 Die Versionszyklen von JavaScript .....	38
1.2.3 Was funktioniert wo? .....	40
1.3 Der Aufbau von Skriptsprachen und die Rolle des Interpreters .....	42
1.4 Zusammenfassung .....	43
<b>2 Die Voraussetzungen schaffen</b> .....	45
2.1 Hardware und Betriebssystem .....	45
2.1.1 Verschiedene Betriebssysteme als Testumgebungen .....	46
2.2 Die Browser .....	48
2.2.1 Benchmark your Webbrowser .....	51
2.3 Der Webserver .....	53
2.4 Die Entwicklungs-Tools .....	56
2.4.1 Integrierte Entwicklungsumgebungen .....	58
2.4.2 Pimp my Firefox – Browser auf Basis von Mozilla aufrüsten ....	59
2.4.3 Pimp my Internet Explorer? .....	64
2.5 Sonstige Software .....	66
2.5.1 Grafikprogramme .....	66
2.5.2 Übertragungssoftware .....	67
2.6 Zusammenfassung .....	69

## Inhalt

<b>3</b>	<b>Auf die Plätze, fertig, los</b>	71
3.1	Projekte – nichts als Projekte	72
3.2	Erste praktische Beispiele	74
3.2.1	Identifizierung eines Browsers	75
3.2.2	Die Bildschirmauflösung eines Besuchers ermitteln	78
3.2.3	Anzeige des aktuellen Datums und der aktuellen Zeit	79
3.2.4	Ein neues Browserfenster öffnen	81
3.2.5	Eine Grafik in einer Webseite dynamisch austauschen	82
3.3	JavaScript in Webseiten einbinden	85
3.3.1	Die Inline-Referenz	86
3.3.2	Die direkte Notation eines Skriptcontainers in der Webseite	87
3.3.3	Der <noscript>-Container	95
3.3.4	JavaScripts über externe Dateien referenzieren	95
3.4	Kann ich testen, ob bei einem Browser JavaScript aktiviert ist?	98
3.5	Zusammenfassung	100
<b>4</b>	<b>JavaScript-Grundstrukturen</b>	101
4.1	Kommentare in JavaScript	102
4.1.1	Mehrzeilige Kommentare	102
4.1.2	Einzeilige Kommentare	103
4.2	Token und Parser	103
4.2.1	Token	104
4.2.2	Parser	104
4.3	Schlüsselwörter in JavaScript	104
4.4	Datentypen, Variablen und Literale	107
4.4.1	Variablen	107
4.4.2	Literale	107
4.4.3	Datentypen und Typisierung	108
4.4.4	Variablen anlegen	112
4.4.5	Ein vollständiges Beispiel zu den Details mit Variablen	115
4.5	Funktionen, Prozeduren und Methoden	117
4.5.1	Form 3 eines Unterprogramms – Methoden	118
4.5.2	Verwendung von vordefinierten Funktionen	119
4.5.3	Die Definition eigener Funktionen	122
4.5.4	Der Aufruf eigener Funktionen	124
4.5.5	Ein Beispiel für selbst definierte Funktionen	125
4.5.6	Verwendung von externen JavaScript-Dateien als Funktionsbibliotheken	126
4.5.7	Lokale Variablen in Funktionen	129

4.5.8	Flexible Anzahl an Funktionsargumenten und der Token arguments . . . . .	132
4.5.9	Rekursive Funktionsaufrufe . . . . .	140
4.5.10	Funktionsreferenzen . . . . .	143
4.5.11	Benannte versus anonyme Funktion . . . . .	145
4.5.12	Innere Funktionen . . . . .	146
4.5.13	Callbacks . . . . .	147
4.6	Anweisungen . . . . .	154
4.6.1	Blöcke . . . . .	154
4.6.2	Ausdrücke . . . . .	155
4.6.3	Operatoren . . . . .	156
4.6.4	Boolesche Operatoren . . . . .	159
4.6.5	Logische Operatoren . . . . .	160
4.6.6	Zuweisungsoperatoren . . . . .	162
4.6.7	Bitweise Operatoren . . . . .	162
4.6.8	Die Operatorenrangfolge . . . . .	169
4.7	Steuerzeichen bei Zeichenketten . . . . .	170
4.8	Kontrollstrukturen in JavaScript . . . . .	171
4.8.1	Die if-Bedingung . . . . .	171
4.8.2	Die switch-Fallunterscheidung . . . . .	173
4.8.3	Die while-Schleife . . . . .	176
4.8.4	Die do-while-Schleife . . . . .	180
4.8.5	Die for-Schleife und die for-in-Schleife . . . . .	182
4.8.6	Gezielte Abbrüche von Schleifen mit break und continue . . . . .	184
4.9	Ein erster Blick auf Objekte und das DOM-Konzept . . . . .	185
4.9.1	Was sind Objekte? . . . . .	186
4.9.2	Wie entstehen Objekte? . . . . .	187
4.10	Zusammenfassung . . . . .	189
<b>5</b>	<b>Eventhandler . . . . .</b>	<b>191</b>
5.1	Was versteht man unter einem Ereignis? . . . . .	192
5.1.1	Eventhandler in HTML bzw. XHTML . . . . .	193
5.1.2	Eventhandler in JavaScript . . . . .	195
5.1.3	HTML- bzw. XHTML- versus JavaScript-Eventhandler . . . . .	196
5.2	Verschiedene Ereignisse . . . . .	197
5.2.1	Laden und Verlassen der Webseite . . . . .	197
5.2.2	Aufruf beim Klick auf ein sensibles Element . . . . .	201
5.2.3	Aufruf beim Überstreichen mit dem Mauszeiger . . . . .	207
5.2.4	Reaktion auf Mausbewegungen . . . . .	209
5.2.5	Die Tastaturereignisse . . . . .	209

## Inhalt

5.2.6	Selektieren von Text .....	212
5.2.7	Unterbrechung eines Bildladevorgangs .....	214
5.2.8	Formularereignisse .....	215
5.2.9	Reaktion auf Zurücksetzen oder Absenden eines Formulars . . . .	217
5.3	Zusammenfassung .....	220
<b>Teil II</b>	<b>Objekte</b> .....	<b>221</b>
<b>6</b>	<b>Objekte und Klassen in JavaScript – Grundlagen</b> .....	<b>223</b>
6.1	Was sind Objekte? .....	224
6.1.1	Die grundsätzliche Definition der objektorientierten Programmierung .....	224
6.2	Klassen, Instanzen und Vererbung .....	225
6.2.1	Vererbung .....	225
6.3	JavaScript-Objekte erzeugen .....	226
6.3.1	Test auf die Existenz eines Objekts .....	230
6.4	Object – ganz oben im Baum .....	231
6.4.1	Eigenschaften von Object .....	231
6.4.2	Methoden von Object .....	232
6.5	Klassenmethoden und Klasseneigenschaften verwenden .....	236
6.6	Zusammenfassung .....	238
<b>7</b>	<b>Native Objekte und Klassenelemente in JavaScript</b> .....	<b>239</b>
7.1	Datenfelder .....	240
7.1.1	Wozu benötigt man Datenfelder? .....	240
7.1.2	Datenfelder erzeugen .....	241
7.1.3	Zugriff auf Elemente im Datenfeld .....	242
7.1.4	Die Array-Notation .....	245
7.1.5	Methoden und Eigenschaften von Datenfeldern .....	249
7.2	Objekte vom Typ Boolean .....	255
7.3	Datumsoperationen mit Date .....	256
7.3.1	Die Methoden von Date .....	258
7.3.2	Erzeugen eines Datumsobjekts mit einem vorgegebenen Datum	261
7.3.3	Ein Schaltjahr bestimmen .....	263
7.3.4	Festlegen des Ablaufdatums für den Inhalt einer Webseite . . . .	265
7.3.5	Berechnen der Zeitspanne bis zu einem bestimmten Termin. . . .	268
7.3.6	Das Erstellungsdatum einer Webseite per JavaScript faken . . . .	270
7.3.7	Veränderte Inhalte auf Grund des Datums .....	273

7.3.8	Die zeitgesteuerte optische Veränderung einer Seite . . . . .	276
7.3.9	Eine Datumsangabe in den IETF-Standard oder eine lokale Darstellung umwandeln . . . . .	278
7.3.10	Die Berechnung regelmäßiger Termine . . . . .	282
7.3.11	Den Unterschied zwischen der lokalen Zeit und Greenwich Mean Time ermitteln. . . . .	283
7.4	String-Operationen. . . . .	284
7.4.1	Methoden von Zeichenketten . . . . .	285
7.4.2	Die Länge einer Zeichenkette und verschiedene String-Methoden . . . . .	289
7.4.3	Die Unicode-Kodierung von Zeichen bestimmen . . . . .	290
7.4.4	Aus Unicode Zeichenketten erzeugen . . . . .	292
7.4.5	Einen Teil eines Strings extrahieren . . . . .	294
7.4.6	Einen String an einem definierten Trennzeichen in ein Datenfeld aufspalten. . . . .	296
7.5	Reguläre Ausdrücke . . . . .	297
7.5.1	Was sind reguläre Ausdrücke? . . . . .	298
7.5.2	Objekte vom Typ RegExp. . . . .	303
7.5.3	Test auf eine gültige E-Mail-Adresse und einen korrekten Usernamen mit test() . . . . .	307
7.5.4	Verwertung des resultierenden Datenfelds beim Durchsuchen eines Strings mit exec(). . . . .	309
7.5.5	Extrahierung der Versionsnummer eines Browsers mit exec() und Auswertung von geklammerten Teilen . . . . .	311
7.5.6	Suchen und Ersetzen von Suchpattern mit search() und replace() . . . . .	313
7.5.7	Sperrern bestimmter Eingaben in einem Webformularfeld mit regulären Ausdrücken und der match()- beziehungsweise replace()-Methode . . . . .	314
7.6	Objekte vom Typ Function . . . . .	317
7.7	Nutzen von Klassenelementen aus Math . . . . .	320
7.7.1	Eigenschaften bzw. Konstanten . . . . .	320
7.7.2	Methoden . . . . .	320
7.8	Objekte und Klassenelemente von Number. . . . .	323
7.8.1	Eigenschaften von Number . . . . .	323
7.8.2	Methoden von Number . . . . .	325
7.9	Zusammenfassung . . . . .	329

## Inhalt

<b>8</b>	<b>DOM – das Mysterium der JavaScript-Welt</b>	331
8.1	Was ist das DOM-Konzept?	332
8.1.1	Grundsätzliches zum Weg durch den DOM	333
8.2	Die verfügbaren DOM-Objekte	337
8.2.1	Das Konzept der Objektfelder	339
8.3	Generelle Zugriffsmöglichkeiten auf DOM-Objekte	340
8.3.1	Der Zugriff über einen Namen – <code>getElementsByName()</code>	341
8.3.2	Zugriff über eine ID – <code>getElementById()</code>	343
8.3.3	Zugriff über den Elementnamen – <code>getElementsByTagName()</code>	344
8.3.4	Der kurze Dienstweg – <code>this</code>	345
8.4	Zugriff auf Inhalte von Elementen in der Webseite	347
8.4.1	Zugriff auf Textinhalte	347
8.4.2	Zugriff auf Formularinhalte	348
8.4.3	Zugriff auf klassische HTML-Attribute	348
8.4.4	Zugriff über <code>nodeValue</code>	353
8.5	Das Objekt <code>document</code>	353
8.5.1	Die Eigenschaften	353
8.5.2	Die Methoden von <code>document</code>	371
8.6	Das <code>node</code> -Objekt	378
8.6.1	Knotenarten	378
8.6.2	Die Eigenschaften eines <code>node</code> -Objekts	380
8.6.3	Die Methoden eines <code>node</code> -Objekts	387
8.7	Zusammenfassung	392
<b>9</b>	<b>Objekterweiterung, Prototyping und die Erstellung eigener Konstruktormethoden</b>	393
9.1	Ein einzelnes Objekt erweitern	394
9.2	Was zeichnet eine Methode gegenüber einer normalen Funktion aus?	396
9.3	Was ist Prototyping?	397
9.4	Erstellen von Konstruktormethoden	406
9.4.1	Was zeichnet eine Konstruktormethode aus?	406
9.4.2	<code>this</code> bei Konstruktormethoden	407
9.4.3	Vererbung mit Prototyping simulieren	412
9.5	Datenkapselung	414
9.6	Zusammenfassung	419

<b>Teil III</b>	<b>Erweiterte JavaScript-Techniken</b>	421
<b>10</b>	<b>Erweiterte JavaScript-Techniken</b>	423
10.1	Was umfasst dieser Abschnitt?	424
10.2	Zusammenfassung	426
<b>11</b>	<b>Fenster und Frames</b>	427
11.1	Der Toplevel und die reservierten Fensternamen	428
11.2	Die Eigenschaften und Methoden des window-Objekts	428
11.2.1	Eigenschaften	429
11.2.2	Die Methoden von window	430
11.3	Anwendungen mit Fenstern	434
11.3.1	Öffnen und Schließen eines Fensters	434
11.3.2	Positionieren und in der Größe festlegen	446
11.3.3	Automatische Weiterleitung von Besuchern	450
11.3.4	Standarddialogfenster von window	451
11.3.5	Die Startseite des Browsers aufrufen	454
11.3.6	Zugriff auf die Statuszeile des Browsers	455
11.3.7	Eine Aktion zeitverzögert aufrufen	457
11.3.8	Zugriff auf die Adresszeile des Webbrowsers	460
11.4	Frames, IFrames und das Objektfeld frames	469
11.4.1	Die Eigenschaften und Methoden von Frame-Objekten	470
11.4.2	Gleichzeitiges Aktualisieren mehrerer Frames	471
11.4.3	Daten aus IFrames laden – ein unsichtbarer Datenspeicher	473
11.4.4	Daten in einem IFrame speichern	478
11.4.5	Verhindern, dass eine Datei in einem Frameset angezeigt wird	480
11.5	Zugriff auf die History des Webbrowsers	481
11.5.1	Die Anzahl der Einträge in der History eines Besuchers	481
11.5.2	Aufruf der zuletzt besuchten Seite	481
11.6	Zugriff auf Bildschirminformationen eines Besuchers	482
11.6.1	Die Bildschirmauflösung	482
11.6.2	Die Farbauflösung bei einem Besucher	484
11.7	Zusammenfassung	485

## Inhalt

<b>12 Browserauswertung</b> .....	487
12.1 Das Objekt navigator .....	488
12.2 Die Browserkennung .....	489
12.2.1 Suche nach spezifischen Informationen in navigator.appName, navigator.appVersion und navigator.userAgent. ....	490
12.2.2 Indirekte Identifizierung eines Browsers unter Ausnutzung eines Mangels .....	494
12.2.3 Was ist die sinnvollste Identifizierungstechnik? .....	496
12.2.4 Die Spracheinstellung eines Browsers .....	497
12.3 Java-Unterstützung bei einem Browser testen .....	499
12.4 Bestimmung, welche Plug-ins ein Browser unterstützt .....	500
12.5 Bestimmung, welche MIME-Types ein Browser unterstützt .....	504
12.6 Eine Browserweiche erstellen .....	508
12.7 Zusammenfassung .....	515
<b>13 Zugriff auf Webformulare</b> .....	517
13.1 Das Formularobjekt .....	518
13.1.1 Die Eigenschaften eine Formularobjekts .....	518
13.1.2 Die Methoden .....	519
13.1.3 Die Elemente in dem Webformular .....	519
13.1.4 Das Objektfeld options .....	521
13.2 Zugriff auf ein Webformular per JavaScript .....	522
13.2.1 Zugriff auf den Wert eines Formularelements .....	522
13.3 Verschicken und Zurücksetzen von Formulardaten .....	537
13.4 Plausibilisieren von Formularen .....	538
13.4.1 Abbruch von Aktionen .....	540
13.4.2 Fehlerreaktion .....	540
13.4.3 Ein Beispiel mit Plausibilisierung .....	541
13.5 Zusammenfassung .....	545
<b>14 Ereignisbehandlung und das event-Objekt</b> .....	547
14.1 Grundlagen zu Ereignisobjekten .....	548
14.1.1 Blubbern .....	548
14.1.2 Die Informationen im Ereignisobjekt .....	549
14.2 Das Ereignisobjekt verwenden .....	549
14.2.1 Das Netscape-Ereignismodell .....	550
14.2.2 Das Microsoft-Ereignismodell .....	551
14.2.3 Die konkrete Verwertung eines Ereignisobjekts .....	552
14.3 Eine universelle Ereignisbehandlung .....	560

14.4	Globale Ereignisbehandlung in JavaScript	564
14.4.1	Registrierung eines Listeners	564
14.4.2	Das Alternativkonzept im Microsoft-Modell	567
14.4.3	Eine universelle Funktion zur Registrierung eines Listener	568
14.5	Zusammenfassung	570
<b>15</b>	<b>Ausnahmebehandlung</b>	<b>571</b>
15.1	Was ist eine Ausnahme?	572
15.1.1	Wozu dient ein Ausnahmekonzept?	573
15.1.2	Die verschiedene Ausnahmetypen	576
15.2	Das Auffangen einer Ausnahme	577
15.3	Selbst definierte Ausnahmen erzeugen und verwenden	584
15.3.1	Erstellen einer selbst definierten Ausnahme	585
15.3.2	Auswerfen von Ausnahmen	585
15.3.3	Freie Benutzereingaben mit dem Ausnahmekonzept absichern	590
15.4	Mehrere Ausnahmen behandeln	594
15.5	Zusammenfassung	595
<b>16</b>	<b>DHTML und Animationstechniken</b>	<b>597</b>
16.1	DHTML mit reinem JavaScript	598
16.1.1	Eine Animation durch Manipulation von Grafiken	602
16.2	Verbinden von JavaScript und Stylesheets über style	606
16.2.1	Die Eigenschaften von style	606
16.2.2	Die Methoden von style	611
16.2.3	Mit style eine Webseite formatieren	612
16.2.4	Den Mauszeiger verändern	613
16.2.5	Ein Farbwähler für die Hintergrundfarbe	616
16.2.6	Die Webseite dynamisch verändern	622
16.2.7	Eine Animation durch Manipulation von Grafiken (style-Version)	626
16.2.8	Eine Interaktionskomponente dynamisieren	629
16.2.9	Ein blinkender Text als Animation mit style	632
16.3	Zusammenfassung	634
<b>17</b>	<b>JavaScript jenseits der Version 1.5 und ECMAScript 5</b>	<b>635</b>
17.1	Neue Datenfeldmethoden in JavaScript 1.6	636
17.1.1	Methoden zum Indexzugriff	636
17.1.2	Iterative Methoden für Datenfelder	638
17.2	E4X – XML in JavaScript	641
17.2.1	Erstellen eines E4X XML-Objekts	641
17.2.2	Variablen in E4X	644

## Inhalt

17.2.3	Zugriff auf Attribute über das E4X-XML-Objekt . . . . .	646
17.2.4	Die Arbeit mit E4X-XML-Objekten . . . . .	648
17.2.5	Methoden eines E4X-XML-Objekts . . . . .	651
17.2.6	Der Umgang mit Objekten vom Typ XMLList. . . . .	652
17.3	JavaScript 1.7 und 1.8. . . . .	654
17.3.1	JavaScript 1.7. . . . .	654
17.3.2	JavaScript 1.8. . . . .	656
17.4	ECMAScript 5 . . . . .	656
17.4.1	JSON. . . . .	657
17.4.2	Objekte schützen . . . . .	658
17.4.3	Ein paar neue offizielle Datenfeldmethoden. . . . .	658
17.4.4	Strenger Verarbeitungsmodus. . . . .	658
17.4.5	Neue Art der Objekteigenschaften und Zugriffsbeschränkungen auf Objekte . . . . .	659
17.4.6	Multithreading über Web Workers. . . . .	660
17.5	Zusammenfassung . . . . .	662
<b>18</b>	<b>Fehlersuche und Fehler vermeiden . . . . .</b>	<b>663</b>
18.1	Welche Fehler gibt es? . . . . .	664
18.1.1	Typografische Fehler. . . . .	665
18.1.2	Syntaktische Fehler. . . . .	666
18.1.3	Laufzeitfehler. . . . .	666
18.2	Gegenmaßnahmen – Fehler verhindern, abfangen und korrekte Versionen sichern . . . . .	667
18.2.1	Vernünftige Sicherungsmaßnahmen – Versionierung . . . . .	667
18.2.2	Fehler zur Laufzeit abfangen . . . . .	667
18.3	Fehler suchen . . . . .	668
18.3.1	Fehler ohne Debugger finden. . . . .	668
18.3.2	Fehlersuche mit einem Debugger . . . . .	670
18.4	Fehler beheben . . . . .	681
18.5	Zusammenfassung . . . . .	682
<b>19</b>	<b>JsUnit – testgetriebene Entwicklung mit JavaScript . . . . .</b>	<b>683</b>
19.1	Was sind Unit-Tests? . . . . .	685
19.1.1	Wie arbeitet man grundsätzlich bei testgetriebener Entwicklung mit xUnit? . . . . .	685
19.1.2	Vorzüge von TDD mittels Unit-Testing. . . . .	686
19.1.3	Probleme von TDD. . . . .	687
19.1.4	Fazit pro und contra. . . . .	688

19.2	Das Ergebnis eines xUnit-Tests. ....	688
19.3	Download und Installation des JsUnit-Frameworks ....	688
19.4	Testseiten und Testfunktionen. ....	690
19.4.1	Der Aufbau von Testfunktionen ....	691
19.4.2	Ausführen eines Tests über den JsUnit Test Runner ....	694
19.5	Bessere Tests ....	699
19.5.1	Level der Verfolgung ....	699
19.5.2	Die Standardfunktionen zur Verfolgung ....	700
19.6	Zusammenfassung ....	706
<b>20</b>	<b>Sicherheit und Optimierung. ....</b>	<b>707</b>
20.1	Welche Sicherheitslücken weist JavaScript auf? ....	708
20.1.1	Signierte Skripte und Sicherheitseinstellungen der Browser. ...	709
20.1.2	Das Motiv ....	713
20.1.3	Was geht wirklich mit JavaScript? ....	714
20.2	Wie kann man seine Skripte schützen? ....	717
20.2.1	Diebstahl geistigen Eigentums. ....	718
20.2.2	Veränderung. ....	721
20.2.3	Deaktivierung durch den Anwender. ....	721
20.3	Optimierung. ....	721
20.3.1	Interpreterprinzip = langsam – JIT = schnell ....	722
20.3.2	Reduzierung der Skriptgröße ....	723
20.3.3	Externe JavaScript-Dateien einsetzen und Kodierung kennzeichnen. ....	724
20.3.4	Auf die überflüssige Erstellung von Objekten verzichten und insbesondere keine Funktionsobjekte benutzen ....	724
20.3.5	Überflüssige Ausnahmenbehandlung vermeiden ....	724
20.3.6	Kein eval(). ....	724
20.3.7	Die Anweisung with() mit Bedacht einsetzen ....	725
20.3.8	Variablen bei der Deklaration immer mit einem Wert versehen und nach Möglichkeit explizit als lokal kennzeichnen ....	725
20.3.9	Die Anweisung for-in kann ineffektiv sein. ....	725
20.3.10	Bei setTimeout() und setInterval() Funktionsreferenzen oder anonyme Funktionen verwenden ....	726
20.3.11	Den DOM der Webseite nicht permanent dynamisch verändern .	726
20.4	Zusammenfassung ....	726

<b>Teil IV</b>	<b>Kommunikation zwischen Client und Server</b>	729
<b>21</b>	<b>Diener und Herr</b>	731
21.1	Die Beziehung zwischen Webserver und Browser	732
21.1.1	Der grundsätzliche Ablauf einer Kommunikation zwischen Herr und Diener – HTTP	732
21.1.2	Verbindungsorientierte Datenkommunikation	734
21.1.3	Zustandslose Datenkommunikation	734
21.1.4	Die HTTP-Details und die HTTP-Header	734
21.1.5	Der HTTP-Request	735
21.1.6	Der HTTP-Response	737
21.1.7	Die Methoden der Datenübertragung per HTTP	738
21.2	Die Meldungen eines Webserver	740
21.3	Zusammenfassung	741
<b>22</b>	<b>Das Datenformat</b>	743
22.1	Binäre Ressourcen	744
22.2	Klartextdatenformate	744
22.2.1	Vollständige HTML- und XHTML-Dateien	745
22.2.2	HTML-Fragmente	745
22.2.3	Reiner Klartext	746
22.2.4	Klartext mit Struktur und beliebiger Logik	746
22.3	Zusammenfassung	754
<b>23</b>	<b>AJAX und das Web 2.0</b>	755
23.1	Was ist AJAX und was bedeutet Web 2.0?	756
23.1.1	AJAX als technische Basis des Web 2.0	757
23.2	Der grundsätzliche Ablauf einer AJAX-Anfrage und das XMLHttpRequest-Objekt (XHR)	761
23.2.1	Der grundsätzliche Ablauf	761
23.2.2	Ein XMLHttpRequest-Objekt erzeugen	762
23.3	Die Anforderung von Daten	765
23.3.1	Die Methoden eines XHR-Objekts	765
23.3.2	Die Eigenschaften eines XMLHttpRequest-Objekts	767
23.4	Verschiedene Datentypen vom Server per AJAX nachfordern	769
23.4.1	Eine reine Textdatei mit AJAX nachfordern	769
23.4.2	Übergabe von Werten an den Webserver	778
23.4.3	Nachfordern von XML-Daten	780

23.4.4	JSON-Daten anfordern und verarbeiten . . . . .	788
23.4.5	Ein praktisches Beispiel mit Auswertung der HTTP-Header . . . .	791
23.5	Statusinformationen für den Anwender . . . . .	793
23.5.1	Die Auswertung des Statuscodes des Servers . . . . .	793
23.6	Zusammenfassung . . . . .	797

**Teil V APIs, Frameworks & Toolkits . . . . . 799**

**24 Frameworks, Toolkit und APIs . . . . . 801**

24.1	Was versteht man unter den Definitionen Framework, Toolkit und API? . .	802
24.1.1	Framework vs. Toolkit . . . . .	802
24.1.2	Web APIs. . . . .	802
24.1.3	Für wen lohnen sich Frameworks, Toolkits und fremde APIs? . . .	803
24.1.4	AJAX als Titel für APIs, Frameworks und Toolkits – nur Mogelpackungen? . . . . .	805
24.2	Verschiedene Ansätze für Frameworks und Toolkits . . . . .	805
24.3	Zusammenfassung . . . . .	806

**25 JavaScript-Frameworks . . . . . 807**

25.1	jQuery . . . . .	808
25.1.1	Download und Installation . . . . .	808
25.1.2	Zugriff auf Elemente der Webseite . . . . .	810
25.1.3	Die Webseite durchlaufen oder manipulieren . . . . .	812
25.1.4	Praktische jQuery-Beispiele . . . . .	813
25.1.5	AJAX und jQuery . . . . .	820
25.1.6	Das jQuery UI . . . . .	824
25.2	Das Dojo-Toolkit . . . . .	826
25.2.1	Download des Dojo Toolkits oder direkte Einbindung aus dem Internet. . . . .	828
25.2.2	Ein Überblick . . . . .	828
25.2.3	Die Datei dojo.js – die Verwaltungszentrale . . . . .	830
25.2.4	Der Einstieg ins Geschäft . . . . .	830
25.2.5	Das Grundgerüst einer Dojo-Applikation . . . . .	833
25.2.6	Die require()-Methode und das Dojo-Paketsystem . . . . .	834
25.2.7	Namensräume . . . . .	835
25.2.8	Schutz des DOM – die Methode dojo.addOnLoad() . . . . .	838
25.2.9	Der Dojo-Typ und das gerenderte Widget . . . . .	839
25.2.10	Ein Beispiel mit Dojo-Widgets . . . . .	840
25.2.11	Das Dijit – Dojo Widget Library . . . . .	843

## Inhalt

25.3	Prototype	843
25.3.1	Der Einstieg	844
25.3.2	Shortcuts	845
25.3.3	Try.these() – Ausnahmebehandlung mit Prototype	848
25.3.4	document.getElementsByClassName()	849
25.3.5	Prototype und AJAX	849
25.4	Zusammenfassung	852
<b>26</b>	<b>Google Web APIs und das Google Web Toolkit – GWT</b>	<b>855</b>
26.1	Die Google Web APIs	856
26.1.1	Die gemeinsame Basis	857
26.1.2	Das Google Maps API	858
26.1.3	Das Google AJAX Feed API	869
26.1.4	Das Google AJAX Search API	872
26.2	Was ist das GWT?	878
26.2.1	Warum mit dem GWT arbeiten?	879
26.2.2	Kontra GWT	882
26.2.3	Der Aufbau eines typischen GWT-Entwicklungszyklus	883
26.2.4	Download und Installation	884
26.2.5	Die GWT-Beispielapplikationen ausführen	888
26.2.6	Der GWT Development Mode	888
26.2.7	Aufruf von Applikationen im GWT Development Mode	888
26.2.8	Erstellen einer eigenen Applikation	892
26.2.9	Vorhandene Beispiele erweitern und modifizieren	895
26.3	Zusammenfassung	900
<b>27</b>	<b>Web-APIs von Microsoft – Bing</b>	<b>901</b>
27.1	Das Bing API	902
27.2	Die Windows Live ID und die AppID	903
27.3	Das Bing API aus einer Webapplikation verwenden	904
27.3.1	Der Umgang mit der Antwort	906
27.4	Zusammenfassung	910
<b>28</b>	<b>Das YUI und die Yahoo! Web APIs und Web Services</b>	<b>911</b>
28.1	Die Web APIs und Web Services	911
28.2	Das YUI	912
28.2.1	Ein Überblick	914
28.2.2	Die Bestandteile von YUI	914
28.2.3	Die Struktur der YUI-Bibliothek	915

28.2.4	Die konkrete Anwendung des YUI .....	916
28.2.5	Wie gehen Sie zum Einstieg am besten vor? .....	917
28.2.6	Exemplarisches Anpassen eines Beispiels .....	918
28.3	Zusammenfassung .....	922
<b>A</b>	<b>Anhang</b> .....	<b>923</b>
A.1	Quellen .....	923
A.2	Die Buch-CD .....	926
A.3	HTML und XHTML – grundlegende Details .....	926
A.3.1	Steueranweisungen .....	926
A.3.2	Parameter .....	927
A.3.3	Strukturierung einer Webseite mit HTML .....	928
A.4	Grundlagen Stylesheets .....	929
A.4.1	Kaskadierung von Regeln .....	929
A.4.2	Die konkrete Syntax von CSS-Deklarationen .....	929
A.5	Das hexadezimale Zahlensystem .....	930
A.6	Der ISO-Latin-1-Zeichensatz .....	930
A.7	Add-ons für Firefox .....	939
<b>Index</b>	.....	<b>943</b>