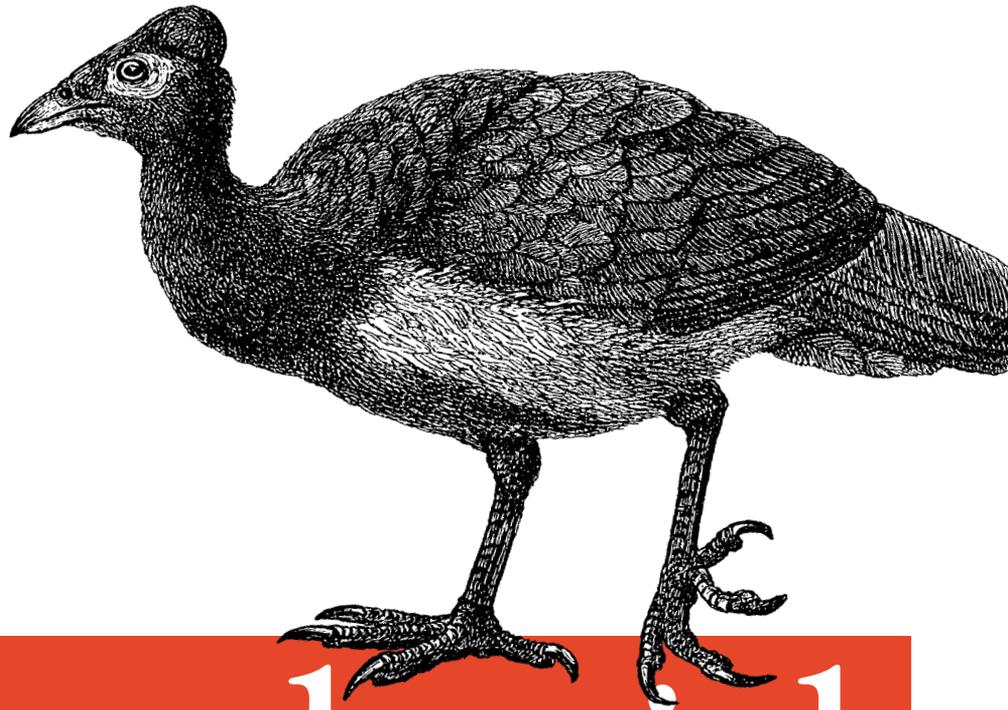


Mit Standard-Web-Tools zur nativen App



Android Apps

mit HTML, CSS und JavaScript

O'REILLY®

Jonathan Stark

Deutsche Übersetzung von Lars Schulten

Inhalt	VII
Einleitung	IX
1 Erste Schritte	1
Web-Apps im Vergleich zu nativen Apps	1
Kurzeinführung in die Webprogrammierung	3
2 Elementares Styling	13
Sie haben keine Website?	13
Erste Schritte	14
Das Android-CSS hinzufügen	20
Das Android-Look-and-Feel einbringen	23
Mit jQuery erste Verhalten einführen	25
Was Sie gelernt haben	31
3 Fortgeschrittenes Styling	33
Einen Spritzer Ajax einbringen	33
Der Verkehrspolizist	34
Etwas Schnickschnack	39
Dem Home-Screen ein Symbol hinzufügen	53
Was Sie gelernt haben	54
4 Animationen	55
Mit etwas Hilfe von unserem Freund	55
Nach Hause rutschen	55
Die Seite »Tage« hinzufügen	59
Die Seite »Tag«	61

Die Seite »Neuer Eintrag«	62
Die Seite »Einstellungen«.	65
Die Teile zusammenfügen	67
jQTouch anpassen	69
Was Sie gelernt haben	72
5 Clientseitige Datenspeicherung	73
Web Storage.	73
Web SQL Database.	79
Was Sie gelernt haben	94
Referenz zu den Web Database-Fehlercodes	94
6 Offline gehen	95
Die Grundlagen des Offline Application Caches	95
Die Online-Whitelist- und -Fallback-Optionen	98
Eine dynamische Manifest-Datei erstellen	101
Debugging	107
Was Sie gelernt haben	110
7 Native Apps.	113
PhoneGap	113
Das Android SDK herunterladen	115
PhoneGap herunterladen.	118
Die Umgebung einrichten	119
Ein virtuelles Android-Gerät erstellen	121
KiloGap erstellen	123
KiloGap im Emulator installieren	126
KiloGap auf Ihrem Gerät installieren	131
Das Gerät mit JavaScript steuern	132
Was Sie gelernt haben	144
8 Die App auf dem Android Market einreichen	145
Die Release-Version der App vorbereiten	145
Die App auf den Android Market hochladen.	149
Apps direkt verteilen	151
Weitere Lektüre	152
Anhang: Browsererkennung mit WURFL.	155
Index.	159

Elementares Styling

Unser endgültiges Ziel ist, mit HTML, CSS und JavaScript eine native Android-App zu erstellen. Der erste Schritt auf dieser Reise besteht darin, dass wir uns damit vertraut machen, wie man HTML so stylt, dass es wie eine mobile App wirkt. In diesem Kapitel werde ich Ihnen zeigen, wie Sie CSS-Styles einsetzen, um bestehenden HTML-Seiten eine Gestalt zu geben, die für die Darstellung auf einem Android-Gerät angemessen ist. Sie nähern sich damit nicht nur dem Ziel, eine native Anwendung aufzubauen, sondern erwerben zugleich eine praktische (und wertvolle) Fertigkeit, die Sie unmittelbar einsetzen können.

Sie haben keine Website?

Sollten Sie Ihre Webseiten auf Ihrem lokalen Rechner testen, können Sie sie nicht auf Ihrem Android-Gerät betrachten. Dazu müssen Sie einen Server aufsetzen. Folgende Optionen stehen Ihnen zur Verfügung:

- Sie können Ihre Webseiten auf einem Webserver hosten und sich mit Ihrem Android-Gerät mit ihm verbinden. Es ist recht wahrscheinlich, dass Ihr *Internet Service Provider* (ISP) zusätzlich Webhosting bietet. In der Regel unterstützt das allerdings nur grundlegende Funktionen wie HTML. Wenn wir Kapitel 6, »Offline gehen«, erreichen, werden wir PHP nutzen müssen. PHP ist eine Skriptsprache, die auf einem Webserver läuft. Sie sollten deswegen nach einem preiswerten Webhoster suchen. Viele Unternehmen, wie Laughing Squid (<http://laughingsquid.us/>) bieten grundlegende Webdienste mit PHP-Unterstützung für zwischen 5 und 10 € im Monat.
- Sie können Ihre Webseiten auf einem Webserver hosten, der auf Ihrem eigenen Rechner läuft, und sich mit Ihrem Android-Gerät mit diesem Webserver verbinden. Das funktioniert aber nur, wenn sich Ihr Android-Gerät und Ihr Rechner im gleichen Drahtlosnetzwerk befinden.

Dieses Kapitel ist so aufgebaut, dass Sie die Beispiele ausprobieren können, während Sie es durcharbeiten. Ganz gleich, welche Option zur Betrachtung der Webseiten Sie wählen, laden Sie sie einfach neu in einem Browser (vorzugsweise dem Android-Browser), wenn Sie einem der Beispiele etwas Neues hinzufügen. Sie sollten allerdings darauf achten, dass

Sie Ihre Seiten im Texteditor gespeichert haben, bevor Sie sie mit dem Browser neu laden, denn sonst werden Sie keine Änderungen sehen.

Einen Webserver lokal ausführen

Alle aktuell relevanten Betriebssysteme (Linux, Windows, Mac OS X) bringen irgendeine Form von Webserver mit. Öffnen Sie unter Mac OS X die Systemeinstellungen, wählen Sie Sharing, und aktivieren Sie WEB-SHARING. Haben Sie das Web-Sharing gestartet, zeigen Ihnen die Web-Sharing-Einstellungen die URL Ihrer persönlichen Website an (diese schließt alles ein, dass Sie im *Websites*-Unterverzeichnis Ihres Benutzerverzeichnisses abgelegt haben). Die URL hat die Form `http://lokaler_Hostname/~Ihr_Benutzername`.

Unter einigen Linux-Versionen wie Ubuntu müssen Sie einige zusätzliche Schritte vornehmen, um einen Webserver zu installieren und zu aktivieren. Erst installieren Sie über die Kommandozeile mit dem Befehl `sudo aptitude install apache2 Apache`. Dann aktivieren Sie das Benutzerverzeichnis-Modul mit `sudo a2enmod userdir`. Haben Sie das erledigt, starten Sie Apache mit folgendem Befehl neu: `sudo /etc/init.d/apache2 restart`. Anschließend können Sie in Ihrem Benutzerverzeichnis ein Verzeichnis mit dem Namen *public_html* erstellen und auf alle Dateien darin über eine URL der Form `http://lokaler_Hostname/~Ihr_Benutzername` zugreifen.

Unter Windows ist die Sache etwas komplizierter. Sie müssen eine Windows-Version nutzen, die die *Internet Information Services* (IIS) enthält. Diese aktivieren Sie über die **WINDOWS-KOMPONENTEN HINZUFÜGEN ODER ENTFERNEN**-Option der Systemsteuerungskomponente **SOFTWARE**. Nachdem Sie das getan haben, können Sie Ihre Webdokumente ins Dokumentenwurzelverzeichnis des IIS stecken, das sich üblicherweise unter `C:\inetpub\wwwroot` befindet. Wenn Sie auch unter Windows den Apache-Webserver vorziehen, sollten Sie sich eines der vorkonfigurierten Pakete wie EasyPHP (<http://www.easyphp.org/>) ansehen oder einen Blick auf die Wikipedia-Seite zu diesem Thema werfen: http://en.wikipedia.org/wiki/Comparison_of_WAMPs.

Erste Schritte

Theorie ist nett, aber da ich eher der »Red' nicht, zeig's mir«-Typ bin, sollten wir jetzt doch langsam mal zur Sache kommen.

Stellen Sie sich vor, Sie haben eine Webseite, die Sie »mobil machen« wollen (Abbildung 2-1). Wenn das der Fall ist, können Sie einige einfache Dinge tun, um eine Site für Android zu optimieren. In diesem Kapitel werde ich die Möglichkeiten durchgehen, die Ihnen zur Verfügung stehen.

Abbildung 2-2 zeigt, wie diese Webseite auf einem Android-Gerät aussieht. Man kann sie nutzen, aber auf die Anforderungen eines Android-Geräts ist sie eigentlich nicht zugeschnitten.

Beispiel 2-1 zeigt eine verkürzte Version der Website in Abbildung 2-2. Das ist das HTML, mit dem wir in diesem Kapitel arbeiten werden. Sie können es von der Webseite zum Buch (<http://www.oreilly.de/catalog/9783897215733>) herunterladen, wenn Sie sich selbst im Stylen versuchen wollen, während Sie das Kapitel durchgehen. Das Desktop-Stylesheet (*screen.css*) wird hier nicht gezeigt, da es nicht relevant ist. Aber Sie können das Stylesheet aus dem letzten Kapitel nehmen, wenn Sie etwas zum Spielen brauchen.

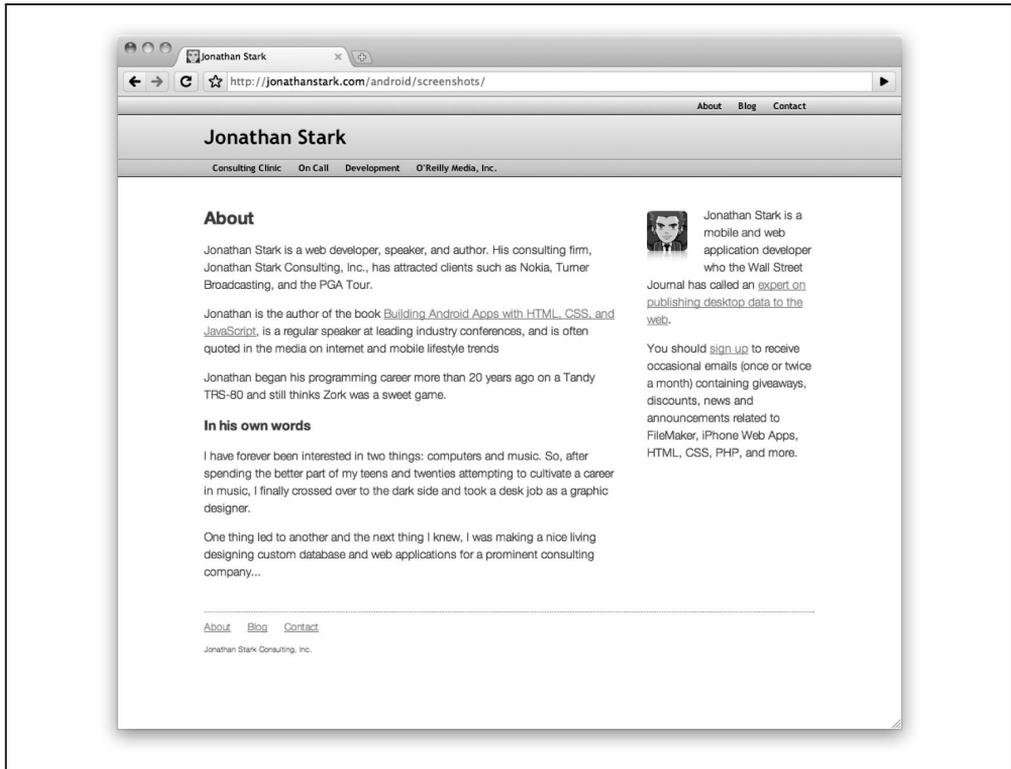


Abbildung 2-1: Die Desktop-Version einer Standard-Webseite in Chrome auf dem Desktop

Beispiel 2-1: Das HTML-Dokument, das wir stylen werden

```
<html>
<head>
  <link rel="stylesheet" href="screen.css" type="text/css" />
  <meta charset="utf-8" />
  <title>Jonathan Stark</title>
</head>
<body>
<div id="container">
  <div id="header">
    <h1><a href=".">Jonathan Stark</a></h1>
    <div id="utility">
      <ul>
        <li><a href="about.html">Info</a></li>
```

```

        <li><a href="blog.html">Blog</a></li>
        <li><a href="contact.html">Kontakt</a></li>
    </ul>
</div>
<div id="nav">
    <ul>
        <li><a href="consulting-clinic.html">Consulting-Klinik</a></li>
        <li><a href="on-call.html">Bereitschaftsdienst</a></li>
        <li><a href="development.html">Entwicklung</a></li>
        <li><a href="http://www.oreilly.com">O'Reilly Media, Inc.</a></li>
    </ul>
</div>
</div>
<div id="content">
    <h2>Info</h2>
    <p>Jonathan Stark ist Webentwickler, Redner und Autor. Zu den Kunden seines Consulting-
    Unternehmen, Jonathan Stark Consulting, Inc., zählen unter anderem Staples, Turner
    Broadcasting und die PGA Tour.
    ...
    </p>
</div>
<div id="sidebar">
    
    <p>Jonathan Stark ist Entwickler für Mobil- und Webanwendungen und wurde vom Wall Street
    Journal als Experte für die Veröffentlichung von Desktop-Daten im Web bezeichnet.</p>
</div>
<div id="footer">
    <ul>
        <li><a href="services.html">Dienste</a></li>
        <li><a href="about.html">Info</a></li>
        <li><a href="blog.html">Blog</a></li>
    </ul>
    <p class="subtle">Jonathan Stark Consulting, Inc.</p>
</div>
</div>
</body>
</html>

```



Jahrelang nutzten Webentwickler Tabellen, um Elemente in einem Raster anzuordnen. Fortschritte in CSS und HTML haben dieses Verfahren nicht nur überflüssig, sondern gar unerwünscht gemacht. Heutzutage nutzen wir hauptsächlich `div`-Elemente (mit einer Vielzahl von Attributen), um bei besseren Steuerungsmöglichkeiten das Gleiche zu erreichen. Obwohl eine vollständige Erklärung `div`-basierter Layouts den Horizont dieses Buches deutlich übersteigt, werden Sie beim Lesen dieses Buches viele Beispiele dafür finden. Wenn Sie mehr wissen wollen, sollten Sie das Buch *Designing with Web Standards* von Jeffrey Zeldman (New Rider Press) lesen, das das Thema mit großer Ausführlichkeit behandelt.



Abbildung 2-2: Desktop-Versionen von Webseiten sehen auf Android-Geräten ordentlich aus, aber das geht noch erheblich besser.

Ein separates Android-Stylesheet vorbereiten

Auch wenn ich ebenso DRY wie alle anderen bin, ist es im wahren Leben häufig empfehlenswert, einen klaren Trennstrich zwischen Desktop-Stylesheets und Android-Stylesheets zu ziehen. Glauben Sie mir einfach, und legen Sie zwei eigenständige Dateien an – das wird Sie ruhiger schlafen lassen. Die Alternative ist, alle CSS-Regeln in ein einziges Stylesheet zu zwingen, aber das ist aus mehreren Gründen wenig empfehlenswert. Der offensichtlichste Grund ist, dass Sie einen Berg irrelevanter Desktop-Style-Regeln an ein Handy versenden und damit wertvolle Bandbreite und Speicherplatz vergeuden.



DRY steht für »Don't Repeat Yourself« (»Vermeiden Sie Wiederholungen«) und ist ein Prinzip der Softwareentwicklung, das besagt: »Jedes Wissens-element muss eine einzige, eindeutige und autoritative Repräsentation in einem System haben!« Der Begriff wurde von Andrew Hunt und David Thomas in ihrem Buch *The Pragmatic Programmer* (Addison-Wesley Professional) geprägt.

Um ein spezielles Stylesheet für Android anzugeben, ersetzen Sie das Link-Tag in unserem HTML-Beispiel durch eines mit den folgenden Ausdrücken:

```
<link rel="stylesheet" type="text/css"
      href="android.css" media="only screen and (max-width: 480px)" />
<link rel="stylesheet" type="text/css"
      href="desktop.css" media="screen and (min-width: 481px)" />
```

Symbole

: (Doppelpunkt) 98, 120
{ } (geschweifte Klammern) 6, 10
. (Punkt) 7, 105
= (Gleichheitszeichen) 10
(Doppelkreuzzeichen) 6–7
| (Pipe) 43
+ (Pluszeichen) 10
; (Semikolon) 10, 120
\$ (Shell-Prompt) 108
/ (Schrägstrich) 3, 105

A

a-Tag (Anker, HTML) 4
Abbrechen-Button 63
absolute URLs 96
:active-Pseudoklasse 51
adb-Tool 128
addGlossToIcon-Eigenschaft 70
Ajax (Asynchronous JavaScript and XML)
 Definition 33
 Stellwerkseite 34
alert-Funktion 132
Alpha-Wert (Farbe) 27
Android Market
 App hochladen auf 149
 Apps direkt verteilen 151
 Release-Version einer App 145
Android SDK
 herunterladen 115
 weitere Lektüre 152
Android-Emulator
 erstellen 121
 gesamte Bildschirmhöhe 129
 KiloGap installieren 126
 Symbol anpassen 130
 Testen, Fortschrittsanzeige 40
Android-Gerät
 Apps testen 13
 KiloGap installieren 131

 mit JavaScript steuern 132
 Seitenskalierung steuern 19
 Standardvergrößerungsstufe 19
 wenig bekannte Funktionen 2
Anfragen steuern 37
anhängende Ellipse 43
Animationen
 jQuery-Bibliothek und 55, 69
 Kalorienzähler-App erstellen 55
 Kalorienzähler-App, Code 67
anonyme Funktionen 48, 134
Apache Ant 123
Apps testen
 Chrome 6, 18, 28
 Fortschrittsanzeige 40
 Hosting 13
Apps versionieren 146
Äquivalenzoperator, logisch 10
Attribute (HTML)
 angepasste Selektoren und 7
 Definition 4
automatisch zum Seitenanfang scrollen 45

B

background-color-Eigenschaft 7
backSelector-Eigenschaft 70
beep-Funktion 132
BEGIN-Anweisung 94
Beschleunigungssensor 142
Block-Tags (HTML) 4
Body-Element
 CSS-Darstellung 6
 Padding-Probleme 23
 PhoneGap-Beispiel 129
border-width-Eigenschaft 27, 49
Browser *siehe* spezifische Browser

C

Cache-Manifest-Datei
 absolute URLs und 96

- dynamische erstellen 101
- elementarer Überblick 95
- Syntaxfehler und 109
- cacheGetRequests-Eigenschaft 70
- Cascading Style Sheets *siehe* CSS
- checkBudget()-Funktion 133
- Chrome-Browser
 - Apps testen 6
 - clientseitige Datenbanken und 82
 - Entwicklertools 83
 - Testüberlegungen 18, 28
- class-Attribut
 - angepasste Selektoren und 7
 - id-Attribut, Vergleich 8
 - Kalorienzähler-App 64
- click()-Funktion 78, 90
- Click-Handler 38
- clone()-Funktion 88
- color-Eigenschaft
 - Beispiel 6
 - Zurück-Button 49
- COMMIT-Anweisung 94
- CONSTRAINT_ERR-Fehlercode 94
- CREATE TABLE-Anweisung 82, 136
- createEntry()-Funktion 83, 134, 137
- Crockford, Douglas 38
- CSS (Cascading Style Sheets) 13
 - anhängende Ellipse 43
 - Einführung 6
 - hinzufügen 20
 - Look-and-Feel ergänzen 23
 - Stylesheets anwenden 8
- cubeSelector-Eigenschaft 70

D

- Database.changeVersion()-Methode 94
- DATABASE_ERR-Fehlercode 94
- DatabaseSync.changeVersion()-Methode 94
- Datenbanken
 - Ergebnismengen verarbeiten 86
 - erstellen 80
 - Zeilen auswählen 86
 - Zeilen einfügen 83
 - Zeilen löschen 90
- Datenspeicherung
 - Fehlercode-Referenz 94
 - Web SQL Database 79
 - Web Storage 73

- Debugging 108
 - JavaScript-Konsole 108
 - Offline Application Cache 107
- Deckkraft (Farbe) 27
- deleteEntryById()-Funktion 91
- digitale Signaturen 146
- dissolveSelector-Eigenschaft 70
- Document Object Model (DOM) 10
- Document-Ready-Funktion 29, 37
- DOM (Document Object Model) 10
- Doppelkreuzzeichen (#) 6–7
- Doppelpunkt (:): 98, 120
- Droid-Schriftfamilie 21
- droidgap-App 123
- DRY-Akronym 17
- dupeEntryById()-Funktion 144

E

- E Text Editor 5
- eckige Klammern, Notation 10
- Eigenschaften, CSS 6
- Einrücken von Text 22
- Ellipse, anhängende 43
- em-Tag (Hervorhebung, HTML) 4
- emacs (Texteditor) 5
- End-Tag (HTML) 3
- entryClickHandler()-Funktion 140
- Ergebnismengen verarbeiten 86
- errorHandler()-Funktion 85
- executeSql()-Methode 84, 88, 133

F

- fadeSelector-Eigenschaft 70
- FALLBACK:-Schlüsselwort 98
- Farbcode-Notation (HTML) 6
- Fehlerbehandlung
 - Einfügen von Zeilen 85
 - Web SQL Database 85, 94
- :first-child-Pseudoklasse 25
- fixedViewport-Eigenschaft 70
- flipSelector-Eigenschaft 70
- font-weight-Eigenschaft 49
- formSelector-Eigenschaft 70
- Fortschrittsanzeige 39
- fullScreen-Eigenschaft 71
- fullScreenClass-Eigenschaft 71

G

- gedit (Texteditor) 5
- Geolocation 136
- Geräteemulator *siehe* Android-Emulator
- geschweifte Klammern {} 6, 10
- getCurrentPosition()-Funktion 137
- getDate()-Funktion 78
- getFilename()-Funktion 105
- getMonth()-Methode 78
- Gleichheitszeichen (=) 10
- goBack()-Funktion 75
- Google Goggles 151
- goTo()-Methode 141
- Gradienten (CSS) 24
- gt()-Funktion 87

H

- h1-Tags (HTML)
 - Text einbetten in 3
- herunterladen
 - Android SDK 115
 - PhoneGap-Entwicklungswerkzeug 118
- hexadezimale Notation 6
- hijackLinks()-Funktion 37, 48
- Home-Screen, Symbole hinzufügen zu 53
- Hosting, Einrichtung 13
- :hover-Pseudoklasse 51
- href-Attribut
 - absoluter Pfad, Beispiel 8
 - Funktion 4
 - relative Pfade, Beispiel 8
- .htaccess-Datei 96–97
- HTML 13
 - Einführung 3
 - Kalorienzähler-App 56
 - Web SQL Database-Spezifikation und 79
- html-Element
 - Funktion 4
 - manifest-Attribut 96
- HTML-Farbcode-Notation 6
- HTML-Tags
 - CSS-Formatierung 7
 - Hyperlinks und 4
 - übliche Struktur 3
 - unterstützte Kategorien 4
- Hunt, Andrew 17
- Hyperlinks
 - CSS-Formatierung 7
 - HTML-Tags 4
 - lokale abfangen 45

I

- icon-Eigenschaft 71
- id-Attribut 7–8
- IF NOT EXISTS-Klausel 82
- IIS (Internet Information Services) 14
- !important-Direktive 129
- index.html-Seite 95
- Inline-Tags (HTML) 4
- innerWidth-Eigenschaft (window) 28
- INSERT-Anweisung 94, 137
- insertEntry()-Funktion 137
- Internet Explorer-Browser 18
- Internet Information Services (IIS) 14
- Internet Service Provider (ISP) 13
- ipfw-Befehl 40
- ISP (Internet Service Provider) 13

J

- JAVA_HOME-Umgebungsvariable 115
- JavaScript 108
 - Anfragesteuerung 37
 - beep()-, vibrate()-, alert()-Funktionen 132
 - Beschleunigungssensor 142
 - Debugging 108
 - Einführung 9
 - Geolocation 136
 - Geräte steuern 132
 - Syntax 10
- jQuery-Bibliothek
 - Abbrechen-Button 63
 - anpassen 69
 - Benutzereinstellungen speichern 76
 - Browser-Toolbar und 129
 - Einführung 55
 - goTo()-Methode 141
 - Kalorienzähler-App 57
- jQuery-Bibliothek
 - click()-Funktion 78, 90
 - Document-Ready-Funktion 29, 37
 - Einführung 11
 - Fortschrittsanzeige 39
 - goBack()-Funktion 75
 - gt()-Funktion 87
 - Kalorienzähler-App 58
 - load()-Funktion 37
 - OnClick-Aktionen kapern 34
 - Seitentitel festlegen 41
 - slideUp()-Methode 91
 - toggleClass()-Funktion 30
 - val()-Funktion 75

- Verhalten gestalten 25
- Zurück-Button 51

K

- Kalorienzähler-App
 - Einstellungen-Seite 65, 74
 - Info-Seite 59
 - Neuer Eintrag-Seite 61
 - Startseite 55
 - Tag-Seite 61, 77
 - Tage-Seite 59, 77
 - vollständiger HTML-Code 67
- Kaneda, David 55
- keytool-Befehl 148
- Kilo-App *siehe* Kalorienzähler-App
- KiloGap-App
 - erstellen 123
 - installieren 126, 131
- Kommandozeilenumgebung 114

L

- :last-child-Pseudoklasse 25
- Lesezeichen für Apps 53
- li-Tag (Listenelement, HTML)
 - Beispiel 3
 - Pseudoclasses und 25
- line-height-Eigenschaft 49
- Linux-Betriebssystem
 - Android SDK herunterladen 115
 - Apache Ant-Unterstützung 123
 - Debugging 108
 - Kommandozeilenunterstützung 114
 - PHP-Skripten ausführen 102
 - Testen, Fortschrittsanzeige 40
 - Texteditor 5
 - Umgebungsvariablen 120
 - Websserver-Unterstützung 14
- load()-Funktion 37
- loadPage()-Funktion
 - Anfragesteuerung 37
 - automatisch an Anfang scrollen 45
 - Zurück-Button 48
- loadSettings()-Funktion 76
- localStorage-Attribut
 - Benutzereinstellungen speichern 74
 - Funktionalität 73
 - PhoneGap-Beispiel 133
- logEvent-Funktion 109
- Logo-Link 21
- Löschen von Zeilen 90

M

- Mac-Betriebssystem
 - Android SDK herunterladen 115
 - Apache Ant-Unterstützung 123
 - Chrome-Browser 6
 - Debugging 108
 - .htaccess-Datei und 97
 - PHP-Skripten ausführen 102
 - Terminal-Anwendung 114
 - Testen, Fortschrittsanzeige 40
 - Texteditor 5
 - Umgebungsvariablen 120
 - Websserver-Unterstützung 14
- Manifest-Datei *siehe* Cache-Manifest-Datei
- max-device-width-Eigenschaft 18
- max-width-Eigenschaft 18, 45, 49
- md5_file-Funktion 107
- Meyer, Eric 9
- MIME-Typen 96
- min-device-width-Eigenschaft 18
- min-width-Eigenschaft 18

N

- nano (Texteditor) 5, 120
- native Apps 123
 - Entwicklung, Vor-/Nachteile 2
 - Merkmale 1
 - richtigen Ansatz wählen 2
- Navigation
 - erstellen 21
 - Menüs abgerundete Ecken geben 24
 - Zurück-Button 46
- netspeed-Kommandozeilenoption 40
- NETWORK:-Schlüsselwort 98
- Nitobi-Entwicklungswerkzeuge 113, 123
- Notepad 5

O

- Offline Application Cache
 - Debugging 107
 - Definition 95
 - dynamische Manifest-Datei erstellen 101
 - elementarer Überblick 95
 - Whitelisting und 98
- onclick-Attribut
 - Aktionen abfangen 34
 - JavaScript-Beispiel 11
- overflow-Eigenschaft 45, 49

P

- p-Tag (HTML-Absatz-Tag) 3
- Padding, Inhalt hinzufügen 22
- pageAnimationBegin-Event 143
- pageAnimationEnd-Event 143
- PATH-Umgebungsvariable 115, 119–120
- PhoneGap-Entwicklungswerkzeug
 - Android SDK herunterladen 115
 - beep()-, vibrate()-, alert()-Funktionen 132
 - Beschleunigungssensor-App 142
 - Einführung 113
 - Geolocation-App 136
 - Geräte mit JavaScript steuern 132
 - herunterladen 118
 - KiloGap erstellen 123
 - KiloGap installieren 126, 131
 - Umgebung einrichten 119
 - virtuelles Android-Gerät erstellen 121
- PHP-Skriptsprache
 - Hosting-Überlegungen 13
 - md5_file-Funktion 107
 - MIME-Typen und 96
 - Skripten ausführen 102
- pico-Editor 97
- Pipe (|) 43
- Pluszeichen (+) 10
- popSelector-Eigenschaft 71
- preloadImages-Eigenschaft 71
- Premium-Modell 2
- preventDefault()-Methode (Event) 38
- Pseudoklassen 25
- Punkt (.) 7, 105

Q

- QR-Code 151
- QUOTA_ERR-Fehlercode 94

R

- refreshEntries()-Funktion
 - Funktion 78
 - PhoneGap-Beispiel 139, 142
 - Zeilen auswählen 86
 - Zeilen einfügen 85
 - Zeilen löschen 90
- reguläre Ausdrücke 46
- REPLACE-Anweisung 94
- Reverse Domain Name-Syntax 124
- RhoMobile-Projekt 114
- Rich-Text-Editing 5

- ROLLBACK-Anweisung 94
- Ruby-Sprache 120

S

- saveSettings()-Funktion 75–76
- Schrägstrich (/) 3, 105
- Schriften, mobile Geräte 21
- script-Element 108
- scrollTo()-Befehl 45
- Seitenskalierung steuern 19
- Seitentitel festlegen 41
- SELECT-Anweisung 142
- Selektoren (CSS)
 - Definition 6
 - Hyperlink-Beispiel 7
 - Navigation aufbauen 21
 - Pseudoklassen und 25
- Semikolon (;) 10, 120
- sessionStorage-Attribut
 - ausgewähltes Datum speichern 77
 - currentDate-Wert 78, 84, 87
 - Funktionalität 73
 - PhoneGap-Beispiel 133
- setDate()-Funktion 78
- Shell-Prompt (\$) 108
- Skripten ausführen 102
- slideInSelector-Eigenschaft 71
- slideUp()-Methode 91
- slideUpSelector-Eigenschaft 71
- Speicherung
 - Fehlercode-Referenz 94
 - Web SQL Database 79
 - Web Storage 73
- Start-Tag (HTML) 3–4
- Starter-Symbole 53
- startupScreen-Eigenschaft 71
- startWatchingShake()-Funktion 143
- statusBar-Eigenschaft 72
- stopPropagation()-Methode 140
- stopWatchingShake()-Funktion 143
- String-Verkettungsoperator 10
- Stylen von HTML-Seiten
 - Ajax 33–34
 - automatisch zum Seitenanfang scrollen 45
 - CSS einbinden 20
 - erste Schritte 14
 - Fortschrittsanzeige 39
 - jQuery-Unterstützung 25

- lange Titel bewältigen 43
- lokale Links abfangen 45
- Look-and-Feel ergänzen 23
- Seitenskalierung steuern 19
- Seitentitel festlegen 41
- seitenweite Styles definieren 20
- Stellwerkseite 34
- Stylesheets vorbereiten 17
- Symbole zu Home-Screens hinzufügen 53
- Testüberlegungen 13
- Zurück-Button 46
- Stylesheets
 - anwenden 8
 - vorbereiten 17
- submitSelector-Eigenschaft 72
- swapSelector-Eigenschaft 72
- Symbole
 - anpassen 130
 - zu Home-Screens hinzufügen 53
- SYNTAX_ERR-Fehlercode 94

T

- Testen von Apps
 - Chrome 6, 18, 28
 - Fortschrittsanzeige 40
 - Hosting 13
- Text einrücken 22
- Text Wrangler (Texteditor) 5
- text-align-Eigenschaft 49
- text-overflow-Eigenschaft 45, 49
- text-shadow-Eigenschaft 23, 49
- TextEdit 5
- Texteditor, auswählen 5
- TextMate (text editor) 5
- Thomas, David 17
- TIMEOUT_ERR-Fehlercode 94
- Titanium Mobile-Projekt 114
- Titelleiste, klickbare 21
- title-Element 5
- toggleClass()-Funktion 30
- toggleMenu()-Funktion 30
- TOO_LARGE_ERR-Fehlercode 94
- Traffic Cop-App 34
- typeof-Operator 129

U

- Überschriften-Tags (HTML)
 - Hyperlink-Beispiel 7
- ul-Tag (ungeordnete Liste, HTML)
 - Elemente verbergen 26

- Funktionalität 3
- Navigation aufbauen 21
- Pseudoclasses und 25
- Umgebungsvariablen 115, 119
- UNKNOWN_ERR-Fehlercode 94
- UPDATE-Anweisung 94
- updateready-Event 97
- url.match-Funktion 46
- useAnimations-Eigenschaft 72

V

- val()-Funktion 75
- var-Schlüsselwort 10
- Variablen
 - deklarieren 10
 - Umgebung 115, 119
- Vergrößerungsstufe, Standard 19
- Verkettungsoperator 10
- VERSION_ERR-Fehlercode 94
- Versionieren von Apps 146
- vi (Texteditor) 5
- vibrate-Funktion 132
- viewport-Meta-Tag 19
- virtuelle Geräte *siehe* Android-Emulator

W

- Web SQL Database
 - Datenbanken erstellen 80
 - Ergebnismengen verarbeiten 86
 - Fehlercode-Referenz 85, 94
 - Funktionalität 79
 - Zeilen auswählen 86
 - Zeilen einfügen 83
 - Zeilen löschen 90
- Web Storage
 - Funktionalität 73
 - localStorage-Attribut 73–74, 133
 - sessionStorage-Attribut 73, 77, 84, 87, 133
- Web-Apps 95
 - Entwicklung, Vor-/Nachteile 2
 - Kennzeichen von 1
 - richtigen Ansatz wählen 2
- Webbrowser *siehe* spezifische Browser
- WebChromeClient-Klasse 154
- webkit-border-image-Eigenschaft 27, 49, 52
- webkit-border-radius-Eigenschaft 24
- webkit-gradient()-Funktion 24
- webkit-tap-highlight-color-Eigenschaft 51

- Webprogrammierung
 - CSS-Einführung 6
 - HTML-Einführung 3
 - JavaScript-Einführung 9
- Webserver
 - Hosting-Überlegungen 13
 - Log-Dateien überwachen 108
 - lokal ausführen 14
 - PHP-Skripten ausführen 102
- WebSettings-Klasse 154
- WebView-Klasse 154
- WebViewClient-Klasse 154
- white-space-Eigenschaft 45, 49
- Whitelisting 98
- window-Objekt
 - applicationCache-Eigenschaft 97
 - innerWidth-Eigenschaft 28
- Windows-Betriebssystem
 - Android SDK herunterladen 115
 - Apache Ant-Unterstützung 123
 - Chrome-Browser 6
 - Eingabeaufforderung 114
 - PHP-Skripten ausführen 102
 - Testen, Fortschrittsanzeige 40
 - Texteditoren 5
 - Umgebungsvariablen 120
 - Webserver-Unterstützung 14
- Wireless Universal Resource File (WURFL)
 - 18, 155
- WordPad 5
- WURFL (Wireless Universal Resource File)
 - 18, 155

Z

- Zeilen
 - auswählen 86
 - einfügen 83
 - löschen 90
- Zeldman, Jeffrey 17
- Zurück-Button 46
- Zuweisungsoperator 10