

Auf einen Blick

1	Apps	21
2	HTML5 in der mobilen Webentwicklung	35
3	Konzeption und Gestaltungsprinzipien	109
4	HTML5 als Designwerkzeug	147
5	Positionsbestimmung	225
6	Auslesen des Bewegungssensors mit JavaScript	261
7	Offline – damit eine App in jedem Winkel der Welt funktioniert	287
8	WebApps mit Frameworks entwickeln	311
9	Native Anwendungen und App Stores	381

Inhalt

Vorwort	11
1 Apps	21
1.1 Native Apps vs. WebApps	21
1.1.1 WebApps allgemein	22
1.1.2 Native Apps auf mobilen Geräten	22
1.1.3 App Store, Google Play und Co.	25
1.1.4 WebApps auf mobilen Geräten	27
1.1.5 Native App oder WebApp?	28
1.2 Warum ist es sinnvoll, auf Touch-Geräten zu entwickeln?	28
1.2.1 Der Mobilsektor boomt	28
1.2.2 Browsertechnik auf Mobilgeräten	30
1.3 Showcase – das können HTML5-Apps	31
2 HTML5 in der mobilen Webentwicklung	35
2.1 HTML5 – Definition und aktueller Stand	35
2.1.1 HTML als Standard	36
2.1.2 HTML5 als Marketinginstrument	36
2.1.3 HTML5 als Flash-Ablösung?	36
2.2 Was HTML5 bietet	37
2.2.1 Der Aufbau einer einfachen HTML5-Seite	37
2.2.2 Semantische Elemente in HTML5	39
2.3 CSS3 – Grundlagen und Neuerungen	42
2.3.1 Was ist eigentlich CSS?	42
2.3.2 Styling mit einer CSS-Datei	43
2.3.3 Gestalten mit CSS – Tipps und Tricks	48
2.3.4 Die Neuerungen von CSS3 – welche sind heute schon einsetzbar?	53
2.4 Desktop vs. Mobile – die Unterschiede	63
2.4.1 Möglichkeiten und Limitationen von HTML5 auf einer mobilen Plattform	68

2.4.2	Die neuen HTML- und CSS-Features in iOS5	69
2.4.3	Von der Webseite zur WebApp	78
2.5	Entwickeln mit JavaScript-Frameworks – jQuery & Co.	83
2.5.1	Die JavaScript-Syntax	84
2.5.2	Einführung in jQuery	88
2.5.3	Wichtige Funktionen in jQuery	94
2.5.4	Nachteile von jQuery	98
2.5.5	Eine Alternative zu jQuery – Zepto.js	99
2.6	Die Geschwindigkeit einer WebApp verbessern	101
2.6.1	Kompression	101
2.6.2	Die richtige Reihenfolge	105
2.6.3	So wenig Anfragen wie möglich	105
3	Konzeption und Gestaltungsprinzipien	109
3.1	Touch-typisches Design in der Praxis	109
3.1.1	Der feine Unterschied zwischen Berühren und Klicken	109
3.1.2	Der Anwender erwartet so einiges – Aufbau einer App	113
3.1.3	Auf der Couch oder in der Bahn	122
3.1.4	Alles eine Frage der Ausrichtung	127
3.2	Eine App konzipieren	134
3.2.1	Vorüberlegungen – was genau soll meine WebApp können?	134
3.2.2	Gedanken und Inhalte ordnen	135
4	HTML5 als Designwerkzeug	147
4.1	Ordnung halten mit CSS-Rastersystemen	147
4.1.1	Einsatz des Less CSS Frameworks	149
4.1.2	Das Raster für einen speziellen Einsatz – MagazineGrid	158
4.2	Mal was anderes als Arial – Webfonts	168
4.2.1	Schriften im Web	169
4.2.2	Webfonts	171
4.2.3	Font-Services	173
4.2.4	Vor- und Nachteile von Webfonts	177

- 4.2.5 Vorsicht! Schriften und deren Lizenzen 178
 - 4.2.6 Webschriften in Aktion 178
- 4.3 Hinzufügen von Audio und Video 180
 - 4.3.1 Video 180
 - 4.3.2 Audio 190
- 4.4 Animationen mit CSS 196
 - 4.4.1 CSS-Transitions und CSS-Animations 196
 - 4.4.2 Animieren mit -webkit-transition 197
 - 4.4.3 Animieren mit -webkit-animation 201
 - 4.4.4 Manipulation mit -webkit-transform 203
- 4.5 Zeichnen mit JavaScript – das canvas-Element 208
 - 4.5.1 Erster Schritt: Canvas und Context registrieren 208
 - 4.5.2 Grundlegende Funktionen 209
 - 4.5.3 Zeichnen per Touch-Eingabe 216
 - 4.5.4 In der Praxis – Canvas oder CSS? 219
 - 4.5.5 Canvas-Bibliotheken 220

5 Positionsbestimmung 225

- 5.1 Die Positionsbestimmung mit HTML5 226
 - 5.1.1 Weitere Eigenschaften der Positionsbestimmung 229
- 5.2 Die »Where to Eat«-App 230
 - 5.2.1 Position auf einer statischen Karte eintragen 231
 - 5.2.2 Interaktive Karten mit der Google Maps JavaScript API einbinden 238
 - 5.2.3 Liste der Restaurants laden und ausgeben 240
 - 5.2.4 Den Abstand zwischen zwei Koordinaten berechnen 244
 - 5.2.5 Die Programmoberfläche 246
 - 5.2.6 Eine Symbolleiste einfügen 248
 - 5.2.7 Die Restaurantliste erstellen 250
 - 5.2.8 Der letzte Schliff 253
- 5.3 More Fun mit Geodaten 256
 - 5.3.1 Qype-API 257
 - 5.3.2 foursquare-API 257
 - 5.3.3 Google Places 258
 - 5.3.4 Twitter-API 258
 - 5.3.5 Flickr-API 260

6 Auslesen des Bewegungssensors mit JavaScript 261

6.1	Diese Daten liefert der Bewegungssensor	262
6.2	Vorüberlegungen zur App »Shake it like a Polaroid picture«	264
6.3	»Shake it like a Polaroid picture« – die Umsetzung	266
6.3.1	Die HTML-Datei	266
6.3.2	Laden eines Bildes aus dem Fotodienst Flickr via JSON	269
6.3.3	Die CSS-Datei	275
6.3.4	Das Laden eines Polaroids	276
6.3.5	Das Sahnehäubchen – eine leichte Rotation des Polaroids	281

7 Offline – damit eine App in jedem Winkel der Welt funktioniert 287

7.1	Abhilfe schafft das Cache-Manifest	289
7.1.1	Die Struktur der Cache-Manifest-Datei	289
7.1.2	Ihre erste Cache-Manifest-Datei	290
7.1.3	Wann werden welche Daten gecacht?	291
7.1.4	Die Cache-Manifest-Datei im Einsatz	292
7.2	localStorage, die Offline-Datenbank	294
7.2.1	localStorage-Funktionen	294
7.2.2	Temporäre Speicherung von Daten im sessionStorage	295
7.3	Die Offline-Todo-App	296
7.3.1	Welche Anforderungen muss Ihre Todo-App erfüllen?	296
7.3.2	Der Wireframe der Todo-App	297
7.3.3	Funktionsbezogene Animation	306
7.3.4	Zusätzliche EventHandlerler	308
7.4	Checkliste zum Überprüfen der Offline-Fähigkeit einer WebApp	309

8 WebApps mit Frameworks entwickeln 311

8.1	Entwicklung einer WebApp mit jQuery Mobile	311
8.1.1	Zum heiteren Hering – ein Fischrestaurant als WebApp	312
8.1.2	Die Startseite	313

8.1.3	Kurze Kaffeepause – ein Blick hinter die Kulissen von jQuery Mobile	324
8.1.4	À la carte – die Menüseite	325
8.1.5	Implementierung einer Google-Maps-Karte inklusive Routenplanung	336
8.1.6	Ein Kontaktformular mithilfe von PHP	340
8.1.7	jQuery-Mobile-Apps mit dem Themeroller und dem Interface-Builder von Codiqa noch schneller entwickeln	347
8.2	Die Entwicklung einer App mit Sencha Touch	350
8.2.1	Installation und erste Schritte mit Sencha Touch	351
8.2.2	Interaktive Diagramme mit Sencha Touch Charts – die Wetterstation	361
8.2.3	HTML5-Tablet-Magazin mit Sencha Touch	371
9	Native Anwendungen und App Stores	381
9.1	Eine WebApp wird zur nativen Anwendung	381
9.1.1	Die Vor- und Nachteile einer nativen App	381
9.1.2	Die Erstellung einer nativen App mit PhoneGap (Apache Cordova)	383
9.1.3	»Shake it like a Polaroid picture« – die native Version	384
9.1.4	Die kompilierten App-Dateien mithilfe von PhoneGap Build erstellen	390
9.1.5	Apple-Zertifikate und Profile	391
9.1.6	Einbinden von Zertifikat und Profil in PhoneGap Build	395
9.1.7	Endlich! Das Hochladen der »Shake it like a Polaroid picture«-Daten	396
9.1.8	Die App-Datei auf dem Endgerät installieren	398
9.1.9	Icon- und App-Namen in PhoneGap Build anpassen	399
9.1.10	Abschließende Worte zu PhoneGap und PhoneGap Build	400
9.2	Eine native Magazin-App mit Laker	401
9.3	Die Veröffentlichung einer App in einem App Store	403
9.3.1	Das Veröffentlichen einer App in Apples App Store	404
9.3.2	Das Veröffentlichen einer App in Google Play	411
9.4	Publizieren der WebApp in einem Store oder als Webseite	416
9.4.1	Auf den Vollbildmodus hinweisen	416

9.4.2 Die Alternative zum App Store – OpenAppMarket und Zeewe 418

9.4.3 Weitere Möglichkeiten 425

A Weiterführende Informationen 427

A.1 HTML5 und CSS3 427

A.2 iOS, Apple 427

A.3 Android, Google 428

A.4 Technologie und Gadgets 428

A.5 Digital Publishing 428

A.6 Design und Inspiration 428

A.7 Sonstiges 429

A.8 Interessante Twitter-Kontakte 429

Index 431