

Auf einen Blick

1	Apps	23
2	HTML5 in der mobilen Webentwicklung	37
3	Konzeption und Gestaltungsprinzipien	109
4	HTML5 als Designwerkzeug	149
5	Positionsbestimmung	239
6	Auslesen des Bewegungssensors mit JavaScript	275
7	Offline – damit eine App in jedem Winkel der Welt funktioniert	317
8	WebApps mit Frameworks entwickeln	341
9	Native Anwendungen und App-Stores	435
10	Windows Phone 8	491
11	Weiterführende Informationen	507

Inhalt

Vorwort 13

Bevor's richtig losgeht – Installation von XAMPP 15

1 Apps 23

1.1 Native Apps vs. WebApps 23

 1.1.1 WebApps allgemein 24

 1.1.2 Native Apps auf mobilen Geräten 24

 1.1.3 Apples App Store, Google Play und Co. 27

 1.1.4 WebApps auf mobilen Geräten 29

 1.1.5 Native App oder WebApp? 30

1.2 Warum ist es sinnvoll, auf Touch-Geräten zu entwickeln? 30

 1.2.1 Der Mobilsektor boomt 30

 1.2.2 Browsertechnik auf Mobilgeräten 32

1.3 Showcase – das können HTML5-Apps 33

2 HTML5 in der mobilen Webentwicklung 37

2.1 HTML5 – Definition und aktueller Stand 37

 2.1.1 HTML als Standard 38

 2.1.2 HTML5 als Marketinginstrument 38

 2.1.3 HTML5 als Flash-Ablösung? 38

2.2 Was HTML5 bietet 39

 2.2.1 Der Aufbau einer einfachen HTML5-Seite 39

 2.2.2 Semantische Elemente in HTML5 41

2.3 CSS3 – Grundlagen und Neuerungen 44

 2.3.1 Was ist eigentlich CSS? 44

 2.3.2 Styling mit einer CSS-Datei 45

 2.3.3 Gestalten mit CSS – Tipps und Tricks 50

 2.3.4 Die Neuerungen von CSS3 – welche sind heute schon einsetzbar? 55

2.4 Desktop vs. Mobile – die Unterschiede	65
2.4.1 Möglichkeiten und Limitationen von HTML5 auf einer mobilen Plattform	70
2.4.2 HTML und CSS für kleine Bildschirme	71
2.4.3 Von der Webseite zur WebApp	80
2.5 Entwickeln mit JavaScript-Frameworks – Zepto und Co.	85
2.5.1 Die JavaScript-Syntax	86
2.5.2 Einführung in Zepto	90
2.5.3 Wichtige Funktionen in Zepto	96
2.6 JavaScript auf dem iPhone testen	100
2.7 Die Geschwindigkeit einer WebApp verbessern	101
2.7.1 Kompression	101
2.7.2 Die richtige Reihenfolge	105
2.7.3 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	114
3.1.3 Auf der Couch oder in der Bahn	123
3.1.4 Alles eine Frage der Ausrichtung	128
3.2 Eine App konzipieren	134
3.2.1 Vorüberlegungen – was genau soll meine WebApp können?	135
3.2.2 Gedanken und Inhalte ordnen	136

4 HTML5 als Designwerkzeug 149

4.1 Ordnung halten mit CSS-Rastersystemen	149
4.1.1 Einsatz des Less CSS Frameworks	151
4.1.2 Das Raster für einen speziellen Einsatz – Magazine Grid	160
4.2 Mal was anderes als Arial – Webfonts	170
4.2.1 Schriften im Web	172
4.2.2 Webfonts	174
4.2.3 Font-Services	175
4.2.4 Vor- und Nachteile von Webfonts	179

4.2.5	Vorsicht! Schriften und deren Lizenzen	180
4.2.6	Webschriften in Aktion	180
4.3	Hinzufügen von Audio und Video	182
4.3.1	Video	182
4.3.2	Audio	192
4.4	Animationen mit CSS	198
4.4.1	CSS-Transitions und CSS-Animations	199
4.4.2	Animieren mit -webkit-transition	199
4.4.3	Animieren mit -webkit-animation	203
4.4.4	Manipulation mit -webkit-transform	206
4.5	Animationen mit Adobe Edge Animate	211
4.5.1	Adobe Edge Animate	211
4.5.2	Ihr erster eigener Teaser	211
4.5.3	Die Oberfläche von Adobe Edge Animate	212
4.5.4	Einstimmung, erste Vorüberlegungen und Unterteilung des Teasers	213
4.5.5	Voreinstellungen: Die Einrichtung der Bühne	213
4.5.6	Teil 1: Textelemente anlegen und animieren	214
4.5.7	Teil 2: Fakten, schnell und knackig!	217
4.5.8	Teil 3: Der Abbinder und der »Call to Action«	218
4.5.9	»Call to Action« – Erstellung eines Buttons	219
4.5.10	Der HTML-Code von Adobe Edge Animate im Detail	221
4.5.11	Abschließende Worte zu Adobe Edge Animate	222
4.6	Zeichnen mit JavaScript – das canvas-Element	223
4.6.1	Erster Schritt: Canvas und Context registrieren	223
4.6.2	Grundlegende Funktionen	224
4.6.3	Zeichnen per Touch-Eingabe	231
4.6.4	In der Praxis – Canvas oder CSS?	234
4.6.5	Canvas-Bibliotheken	235

5 Positionsbestimmung 239

5.1	Die Positionsbestimmung mit HTML5	240
5.1.1	Weitere Eigenschaften der Positionsbestimmung	243
5.2	Die Where-to-Eat-App	244
5.2.1	Position auf einer statischen Karte eintragen	246
5.2.2	Interaktive Karten mit der Google Maps JavaScript-API einbinden	253
5.2.3	Liste der Restaurants laden und ausgeben	255

5.2.4	Den Abstand zwischen zwei Koordinaten berechnen	258
5.2.5	Die Programmoberfläche	260
5.2.6	Eine Symbolleiste einfügen	262
5.2.7	Die Restaurantliste erstellen	265
5.2.8	Der letzte Schliff	267
5.3	More Fun mit Geodaten	270
5.3.1	Yelp-API	271
5.3.2	foursquare-API	271
5.3.3	Google Places	272
5.3.4	Twitter-API	272
5.3.5	Flickr-API	273

6 Auslesen des Bewegungssensors mit JavaScript 275

6.1	Diese Daten liefert der Bewegungssensor	276
6.2	Vorüberlegungen zur App »Shake it like a Polaroid picture«	279
6.3	»Shake it like a Polaroid picture« – die Umsetzung	280
6.3.1	Die HTML-Datei	280
6.3.2	Laden eines Bildes aus dem Fotodienst Flickr via JSON	283
6.3.3	Die CSS-Datei	289
6.3.4	Das Laden eines Polaroids	290
6.3.5	Das Sahnehäubchen – eine leichte Rotation des Polaroids	295
6.4	Die Anzeige eigener Bilder mit dem Input-Element	300
6.4.1	Einführung in den Dateiupload ab iOS 6 und Android 2.2	300
6.4.2	Erweitern der Shake-it-WebApp um eigene Fotos – Übersicht	302
6.4.3	Abschließende Worte/Einschränkungen	315

7 Offline – damit eine App in jedem Winkel der Welt funktioniert 317

7.1	Abhilfe schafft das Cache-Manifest	319
7.1.1	Die Struktur der Cache-Manifest-Datei	319
7.1.2	Ihre erste Cache-Manifest-Datei	320
7.1.3	Wann werden welche Daten gecached?	322
7.1.4	Die Cache-Manifest-Datei im Einsatz	322

7.2	Der Local Storage – die Offline-Datenbank	324
7.2.1	localStorage-Funktionen	324
7.2.2	Temporäre Speicherung von Daten im Session Storage	325
7.3	Die Offline-To-do-App	326
7.3.1	Welche Anforderungen muss Ihre To-do-App erfüllen?	326
7.3.2	Der Wireframe der To-do-App	327
7.3.3	Funktionsbezogene Animation	336
7.3.4	Zusätzliche EventHandlerler	338
7.4	Checkliste zum Überprüfen der Offline-Fähigkeit einer WebApp	339

8 WebApps mit Frameworks entwickeln 341

8.1	Entwicklung einer WebApp mit jQuery Mobile	341
8.1.1	Zum Heiteren Hering – ein Fischrestaurant als WebApp	342
8.1.2	Die Startseite	343
8.1.3	Kurze Kaffeepause – ein Blick hinter die Kulissen von jQuery Mobile	354
8.1.4	À la Carte – die Menüseite	355
8.1.5	Implementierung einer Google Maps-Karte inklusive Routenplanung	366
8.1.6	Ein Kontaktformular mithilfe von PHP	371
8.1.7	jQuery Mobile-Apps mit dem ThemeRoller und dem Interface-Builder von Codiqa noch schneller entwickeln	378
8.2	Die Entwicklung einer App mit Sencha Touch	380
8.2.1	Installation und erste Schritte mit Sencha Touch	381
8.2.2	Interaktive Diagramme mit Sencha Touch Charts – die Wetterstation	393
8.2.3	HTML5-Tablet-Magazin mit Sencha Touch	402
8.3	Responsive Apps mit Twitter Bootstrap	411
8.3.1	Was ist das Responsive Webdesign?	411
8.3.2	Was ist Bootstrap?	411
8.3.3	Grundlagen	412
8.3.4	Die Instagram-App	420
8.3.5	Ressourcen	430

9.1 Eine WebApp wird zur nativen Anwendung	435
9.1.1 Die Vor- und Nachteile einer nativen App	435
9.1.2 Die Erstellung einer nativen App mit PhoneGap (Apache Cordova)	437
9.1.3 »Shake it like a Polaroid picture« – die native Version	438
9.1.4 Die kompilierten App-Dateien mithilfe von PhoneGap Build erstellen	444
9.1.5 Apple-Zertifikate und Profile	446
9.1.6 Einbinden von Zertifikat und Profil in PhoneGap Build	450
9.1.7 Endlich! Das Hochladen der Shake-it-like-a-Polaroid- picture-Daten	451
9.1.8 Die App-Datei auf dem Endgerät installieren	452
9.1.9 Icon- und App-Namen in PhoneGap Build anpassen	455
9.1.10 Abschließende Worte zu PhoneGap und PhoneGap Build	455
9.2 Ein Spiel mit PhoneGap	456
9.2.1 Die Welt der Spiele	457
9.2.2 Die Grundausstattung	458
9.2.3 Runde für Runde – JavaScript, die Erste	460
9.2.4 Die Positionierung einer Schießscheibe	463
9.2.5 Das Resultat einblenden	464
9.2.6 Hinzufügen der PhoneGap-Funktionen	465
9.2.7 Lasset die Duelle beginnen	469
9.3 Eine native Magazin-App mit Laker	469
9.4 Die Veröffentlichung einer App in einem App-Store	472
9.4.1 Das Veröffentlichen einer App in Apples App Store	472
9.4.2 Das Veröffentlichen einer App in Google Play	479
9.5 Publizieren der WebApp in einem Store oder als Webseite	483
9.5.1 Auf den Vollbildmodus hinweisen	483
9.5.2 Die Alternative zu App-Stores – OpenAppMarket	485
9.5.3 Weitere Möglichkeiten	490

10 Windows Phone 8

10.1 Was ist Windows Phone?	491
10.1.1 Der Browser	492
10.1.2 Gestaltungsprinzipien des Modern UI	493

10.1.3	Grundlagen – Raster	495
10.1.4	Die To-do-App	499
10.1.5	Zum Home-Bildschirm hinzufügen	503
10.1.6	Weiterführende Infos	504

A	Weiterführende Informationen	507
A.1	HTML5 und CSS3	507
A.2	JavaScript und nützliche Tools	508
A.3	iOS, Apple	508
A.4	Android, Google	509
A.5	Technologie und Gadgets	509
A.6	Digital Publishing	510
A.7	Design und Inspiration	510
A.8	Sonstiges	511
A.9	Interessante Twitter-Kontakte	511
	Index	513