

## B Literaturempfehlungen

Dieser Anhang fasst noch einmal alle in den »Literaturempfehlungen« der einzelnen Kapitel aufgeführten Links zusammen, sodass Sie sie bequem nachschlagen können. Eine stets aktuelle und gegebenenfalls um neue und interessante Links erweiterte Version dieser Liste finden Sie auf der Website zum englischen Buch (<http://modernwebbook.com>).

### Einleitung

Die in der Einleitung genannten Statistiken stammen aus vielen verschiedenen Quellen, vor allem *The Mobile Industry in Numbers* von VisionMobile auf <http://www.visionmobile.com/blog/2012/10/infographic-the-mobile-industry-in-numbers/> sowie »The Internet of Things« von Cisco auf <http://blogs.cisco.com/news/the-internet-of-things-infographic/>.

Eine gute Einführung in das Internet of Things finden Sie im *The Next Web*-Artikel *Why 2013 will be the year of the Internet of Things*: <http://thenextweb.com/insider/2012/12/09/the-future-of-the-internet-of-things/>.

David Storey schrieb einen hervorragenden Beitrag über das nicht auf Smartphones ausgerichtete mobile Web: *See your site like the rest of the world does. On the Nokia X2-01*. Sie finden ihn auf <http://generatedcontent.org/post/31441135779/mobileweb-row/>.

Der beste Artikel, den ich über die Webentwicklung für mobile Geräte gelesen habe und aus dem ich in diesem Kapitel zitiere, ist *The 10 Principles of Mobile Interface Design* von Jonathan Stark: <http://www.netmagazine.com/features/10-principles-mobile-interface-design/>. Jason Grigsbys hervorragender Artikel *Responsive Design for Apps* ist ein guter Einstieg in die Gestaltung für unterschiedliche Bildschirmabmessungen und -darstellungen: <http://blog.cloudfour.com/responsive-design-for-apps-part-1/>. Der im *UX Magazine* erschienene Artikel *Five Lessons from a Year of Tablet UX Research* von Brennan Browne enthält herausragende Erkenntnisse über die Nutzung von Tablets: <http://uxmag.com/articles/five-lessons-from-a-year-of-tablet-ux-research/>.

Ein guter Einstieg für Anna Debenhams Untersuchungen über Spielekonsolen-Browser ist ihr Artikel *Testing Websites in Game Console Browsers* auf *A List Apart*: <http://www.alistapart.com/articles/testing-websites-in-game-console-browsers/>.

Noch einmal Jason Grigsby: Er hielt den wunderbaren Vortrag *The Immobile Web* über die Entwicklung für TV-Geräte. Das Video finden Sie auf <http://vimeo.com/44444464/>, die zugehörigen Präsentationsfolien auf <http://www.slideshare.net/grigs/the-immobile-web/>.

Die komplette Untersuchung zur Multi-Device-Nutzung finden Sie im Google-Blog-Beitrag *Navigating the new multi-screen world* auf <http://googlemobileads.blogspot.co.uk/2012/08/navigating-new-multi-screen-world.html>.

Es ist immer gut, zukunftsweisende Websites zu gestalten. Mehr auf <http://futurefriend.ly/>.

## Kapitel 1

Die *Liste der Technologien, die die Webplattform ausmachen*, finden Sie auf <http://platform.html5.org/>. Bruce Lawson schlug den Begriff NEWT in seinem Blog vor: <http://www.brucelawson.co.uk/2010/meet-newt-new-exciting-web-technologies/>.

Die *HTML5-Spezifikation des W3C* finden Sie auf <http://www.w3.org/TR/html5/> und die *Live-Spezifikation des WHATWG* auf <http://whatwg.org/html>. Es gibt dort auch eine nützliche *Edition for Web Developers*, in der gewisse obskure Sprachelemente weggelassen werden und die deshalb besser lesbar ist: <http://developers.whatwg.org/>.

Die *vollständige Formulierung des HTML5-Standards* gibt es auf <http://html5boilerplate.com/>. Denken Sie daran: Verwenden Sie nur die von Ihnen benötigten Teile; Sie müssen nicht alles wörtlich umsetzen.

Mehr über den Implementierungsgrad verschiedener Funktionen bieten Alexis Deverias Site *Can I Use* auf <http://caniuse.com/>, die Community-Site *HTML5 Please* auf <http://html5please.com/> sowie *The HTML5 Test* auf <http://html5test.com/>.

Die Website *LabUp!* ist eine Ressource für Open Device Labs: <http://lab-up.org/>. Der Testleiter der BBC, David Blooman, beschäftigte sich in dem langen und ausführlichen Artikel *Testing for Dummies* damit, wie ein großes Unternehmen Multi-Device-Tests durchführt: <http://mobiletestingfordummies.tumblr.com/post/20056227958/testing>.

Patrick Meenans Präsentationsfolien für seinen Vortrag *Taming the Mobile Beast* enthalten eine Fülle von Links und Informationen über Tests auf mobilen Geräten: <http://www.slideshare.net/patrickmeenans/velocity-2012-taming-the-mobile-beast/>. Und Anna Debenhams Artikel für A List Apart *Testing Websites in Game Console Browsers* handelt von ... na ja, der Titel ist ziemlich selbsterklärend: <http://www.alistapart.com/articles/testing-websites-in-game-console-browsers/>.

Opera bietet eine *detaillierte Anleitung zum Remote-Debugging*: <http://www.opera.com/dragonfly/documentation/remotel/>. *weinre* können Sie sich von <http://people.apache.org/~pmuellr/weinre/docs/latest/> herunterladen. Weitere *Informationen über Adobe Edge Inspect* erhalten Sie auf <http://html.adobe.com/edge/inspect/>.

## Kapitel 2

HTML5 Doctor ist die beste Quelle für Informationen über die meisten HTML5-Themen. Sie finden beispielsweise im folgenden *Artikel von Mike Robinson* die meiner Meinung nach beste Definition der neuen Gliederungsalgorithmen: <http://html5doctor.com/outlines/>. Das in *Abb. 2-1* gezeigte *Schaubild* können Sie von <http://html5doctor.com/resources/#flowchart/> herunterladen.

Lesen Sie auch *Derek Johnsons Artikel* im Smashing Magazine: <http://coding.smashingmagazine.com/2011/08/16/html5-and-the-document-outlining-algorithm/>.

Für tiefere Einblick in das Problem der strukturierenden Elemente in HTML5 (Sectioning Elements) empfehle ich Luke Stevens' Buch *The Truth About HTML5*. Sie finden es auf <http://www.truthabouthtml5.com/>. Wenn Sie die *vollständige HTML5-Spezifikation* studieren möchten, um selbst zu entscheiden, rate ich Ihnen zur Entwicklerversion auf <http://developers.whatwg.org/sections.html>.

Die *vollständige WAI-ARIA-Spezifikation* finden Sie auf <http://www.w3.org/TR/wai-aria/>. Für Informationen über die Barrierefreiheit von HTML5 lohnt es sich, den *Paciello Group Blog* zu lesen. Wichtig ist in diesem Zusammenhang der Beitrag <http://www.paciello.com/blog/2010/10/using-wai-aria-landmark-roles/>.

*Divya Manians Artikel über Semantik* erschien im Smashing Magazine auf <http://coding.smashingmagazine.com/2011/11/11/our-pointless-pursuit-of-semantic-value/>. Wenn Sie mehr über Aboutness und die Bedeutung der Semantik lesen möchten, empfehle ich unbedingt das Buch *Ambient Findability: What We Find Changes Who We Become* von Peter Morville (O'Reilly, 2005). Die Website <http://webdatacommons.org/> liefert *Informationen und Statistiken über Sites, die strukturierte Daten nutzen*.

Lesen Sie *alles über Mikroformate* auf <http://microformats.org/>. Eine Überarbeitung der Syntax, Microformats 2.0, begann im Jahr 2010 und ist immer noch in Entwicklung begriffen; erfahren Sie mehr darüber auf <http://microformats.org/wiki/microformats-2>.

Möchten Sie mehr über das RDFa-Format lesen, hat das W3C eine exzellente *Einführung* veröffentlicht: <http://www.w3.org/TR/xhtml-rdfa-primer/>. Die beste *Ressource über Mikrodaten* stammt ebenfalls vom HTML5 Doctor: <http://html5doctor.com/microdata/>. Sind Sie ganz masochistisch veranlagt und möchten Sie die *Spezifikation im Ganzen* lesen, finden Sie diese auf <http://www.w3.org/TR/microdata/>.

Weitere *Informationen über Schema.org* erhalten Sie auf <http://schema.org/>, und die *Google-Dokumentation über Rich Snippets* befindet sich auf <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=99170>. Das *Test-Tool* finden Sie auf <http://www.google.com/webmasters/tools/richsnippets/>.

John Resig schrieb in seinem Blog <http://ejohn.org/blog/html-5-data-attributes/> eine einprägsame *Einführung zu den Datenattributen*, und die `data()`-Methode wird vollständig auf der jQuery-Website auf <http://api.jquery.com/data/> dokumentiert.

## Kapitel 3

Die erste Anlaufstelle, wenn Sie mehr über Media Queries erfahren möchten, sollte Zoe Mickley Gillenwaters Post *Essential Considerations for Crafting Quality Media Queries* sein: <http://zomigi.com/blog/essential-considerations-for-crafting-quality-media-queries/>.

Die maßgebliche Autorität für Mobilgeräte ist Peter-Paul Koch. Wenn Sie mehr über virtuelle und physische Pixel erfahren möchten, empfehle ich Ihnen für den Einstieg seinen Artikel *A Pixel Is Not a Pixel*: [http://www.quirksmode.org/blog/archives/2010/04/a\\_pixel\\_is\\_not.html](http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html). Wikipedia bietet eine *Auflistung der gängigen Geräteauflösungen und Pixeldichten*: [http://en.wikipedia.org/wiki/List\\_of\\_displays\\_by\\_pixel\\_density](http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density).

Patrick Lauke schrieb einen Artikel über benutzerkontrollierte DPR: *devicePixelRatio in Opera Mobile*. Sie finden ihn auf <http://my.opera.com/ODIN/blog/2012/07/05/devicepixelratio-in-opera-mobile>. Matt Wilcox's Artikel *The Responsive Design Process* enthält ein gutes Glossar mit Schlüsselbegriffen sowie zahlreiche praktische Design-Hinweise: <http://mattwilcox.net/archive/entry/id/1078/>.

Mehr darüber, wie unterschiedliche Browser Dezimalstellen runden, erfahren Sie in John Albin Wilkins' Beitrag *Responsive Design's Dirty Little Secret*: <http://www.palantir.net/blog/responsive-design-s-dirty-little-secret/>.

Paul Irishs Blogbeitrag *box-sizing: border-box FTW* legt seine Gründe dar, warum diese Eigenschaft global gesetzt werden sollte: <http://paulirish.com/2012/box-sizing-border-box-ftw/>.

Luke Wroblewskis Buch *Mobile First* ist bei A Book Apart erschienen: <http://www.abookapart.com/products/mobile-first/>.

*Mehr über Inhalts-Breakpoints* erfahren Sie in dem Artikel des australischen Webdesign-Studios Jor design (<http://www.jordesign.com/blog/responsive-breakpoints-from-the-content-out/>) und dem Artikel des Entwicklers Thierry Koblentz (<http://coding.smashing-magazine.com/2012/03/22/device-agnostic-approach-to-responsive-web-design/>).

Die *Geschichte des momentan wichtigsten Vorschlags für responsive Bilder und Informationen zu seinem aktuellen Status* finden Sie auf der Website der Responsive Images Community Group: <http://www.w3.org/community/respimg/>.

Matt Wilcox' *Tool für adaptive Bilder* finden Sie auf <http://adaptive-images.com/>.

## Kapitel 4

Das stets zuverlässige MozDev enthält eine klare Einführung zu mehrspaltigen Layouts, *Using CSS multi-column layouts*: [https://developer.mozilla.org/en-US/docs/CSS/Using\\_CSS\\_multi-column\\_layouts/](https://developer.mozilla.org/en-US/docs/CSS/Using_CSS_multi-column_layouts/).

Die Flexbox-Syntax hat sich so oft geändert, dass es quasi keine aktuelle Online-Ressource mehr gibt! Auch wenn sich Stephen Hays Artikel *Learn You a Flexbox for Great Good!* auf <http://www.the-haystack.com/2012/01/04/learn-you-a-flexbox/> auf eine veraltete Syntax bezieht, sollten Sie ihn lesen, denn Stephens Kenntnisse des CSS-Layouts sind unvergleichlich.

Die beste Erklärung des Grid-Layout-Moduls, zumindest hinsichtlich der Implementierung im Internet Explorer 10, finden Sie im »*Internet Explorer 10 Guide for Developers*«: <http://msdn.microsoft.com/en-us/library/ie/hh673533%28v=vs.85%29.aspx/>.

## Kapitel 5

Die *Illustration in Abb. 5–1* stammt aus Peter Beverloos Blog: <http://peter.sh/experiments/asynchronous-and-deferred-javascript-execution-explained/>.

Christian Heilmann hat eine fundierte *Einführung in JavaScript-Ereignisse* für das Smashing Magazine geschrieben: <http://coding.smashingmagazine.com/2012/08/17/javascript-events-responding-user/>. Die *PointerEvents-Bibliothek* finden Sie bei GitHub unter <https://github.com/toolkitchen/PointerEvents/>.

Die jQuery-Website <http://jquery.com/> liefert *Hinweise für den Einstieg*, die ausgezeichnete Dokumentation finden Sie unter [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page). Die statistischen Erhebungen zur Verwendung von jQuery entstammen dem Blogbeitrag *jQuery Now Runs on Every Second Website* unter [http://w3techs.com/blog/entry/jquery\\_now\\_runs\\_on\\_every\\_second\\_website/](http://w3techs.com/blog/entry/jquery_now_runs_on_every_second_website/).

Alle *Bibliotheken für Mobilgeräte* sind vollständig dokumentiert: jQuery Mobile unter <http://jquerymobile.com/>, Zepto.js unter <http://zeptojs.com/> und jQTouch unter <http://jqtouch.com/>.

*YepNope.js* finden Sie unter <http://yepnopejs.com/> und eine gute *Einführung* unter <http://net.tutsplus.com/tutorials/javascript-ajax/easy-script-loading-with-yepnope-js/>.

Die Modernizr-Website <http://modernizr.com/> bietet eine *vollständige Dokumentation* und ein *konfigurierbares Build-Tool*. Den Leitfaden *The All-In-One Entirely-Not-Alphabetical No-Bullshit Guide to HTML5 Fallbacks* (ich habe den Titel nicht erfunden) finden Sie unter <https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills/>.

Christopher Coenraets hat ein hervorragendes *Einstiegstutorial für Mustache* geschrieben, allerdings hat sich die Syntax geringfügig geändert: <http://coenraets.org/blog/2011/12/tutorial-html-templates-with-mustache-js/>. Die *vollständige Dokumentation zu Mustache.js* befindet sich hier: <https://github.com/janl/mustache.js/>.

Es gibt zahlreiche *Entwicklertools zum Experimentieren und Debuggen*, ganz hervorragend sind sowohl <http://jsbin.com/> als auch <http://jsfiddle.net/>.

Die *aktuellste Version des Grid-Layout-Moduls* finden Sie unter <http://dev.w3.org/csswg/css-grid>.

Lesen Sie *Mark Boultons offenen Brief* zur Grid-Layout-Terminologie auf <http://www.markboulton.co.uk/journal/comments/open-letter-to-w3c-css-working-group-re-css-grids>.

## Kapitel 6

Dive Into HTML5 liefert unter <http://diveintohtml5.info/geolocation.html> eine ausführliche *Erklärung der Geolocation-API*, während der MozDev-Artikel *Orientation and Motion Data Explained* einen guten Überblick zur dreidimensionalen Ausrichtung und Bewegung gibt: [https://developer.mozilla.org/en-US/docs/DOM/Orientation\\_and\\_motion\\_data\\_explained/](https://developer.mozilla.org/en-US/docs/DOM/Orientation_and_motion_data_explained/).

Die Fullscreen-API wird im Sitepoint-Artikel *How to Use the HTML5 Full-Screen API* von Craig Buckler erklärt, allerdings wurde die API geringfügig überarbeitet, während ich dies schrieb; einige Objekt-namen oder Eigenschaften könnten sich also geändert haben. Den Artikel finden Sie unter <http://www.sitepoint.com/html5-fullscreen-api/>.

Die *Battery-Status-API* wird von David Walsh unter <http://davidwalsh.name/battery-api/> gut erklärt, und eine *Diskussion der früheren und neu aktualisierten Network-Information-API* findet sich hier: <http://nostrongbeliefs.com/a-quick-look-network-information-api/>.

HTML5 Rocks bietet im Artikel *Capturing Audio & Video in HTML5* die beste Erklärung von `getUserMedia()`: <http://www.html5rocks.com/en/tutorials/getusermedia/intro/>. Die vollständige *Zielsetzung des WebRTC-Projekts* ist unter <http://www.webrtc.org/> aufgelistet.

(Wieder einmal) MozDev liefert eine prägnante *Einführung in die Web-Storage-API*: <https://developer.mozilla.org/en-US/docs/DOM/Storage/>.

Der am besten verständliche *Leitfaden zur Drag-and-Drop-API*, den ich finden konnte, stammt von den HTML5 Doctors unter <http://html5doctor.com/native-drag-and-drop/>, während das fünfteilige *Working with Files in JavaScript* von Nicholas Zakas eine exzellente Ressource zur File-API darstellt: <http://www.nczonline.net/blog/2012/05/08/working-with-files-in-javascript-part-1/>.

Die *APIs, aus denen das Firefox-OS-Projekt besteht*, sind unter <https://wiki.mozilla.org/WebAPI/> aufgeführt, und die Folien der Präsentation *WebAPIs and Apps* von Robert Nyman liefern einen tollen Überblick darüber: <http://www.slideshare.net/robnyman/webapis-apps-mozilla-london/>. *Are We Mobile Yet?* zeigt aktuell und auf einen Blick, wie es um die Implementierung von APIs bestellt ist: <http://arewemobileyet.com/>.

## Kapitel 7

Eine hervorragende *SVG-Einführung* finden Sie auf der Website SVG Basics: <http://www.svgbasics.com/>, und der *SVG-Primer* des W3C geht noch mehr in die Tiefe: <http://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>. MDN bietet eine ziemlich vollständige *Elemente- und Attributliste*: <https://developer.mozilla.org/en-US/docs/SVG/>.

Die *Technik zum Stapeln von SVG-Symbolen* stammt von Erik Dahlström und wird im Simurai-Blog beschrieben: <http://simurai.com/post/20251013889/svg-stacks/>. Robert O'Callahan von Mozilla warnt vor *möglichen Veränderungen der Stapeltechnik*: <http://robert.ocallahan.org/2012/10/impending-doom-for-svg-stacks-sort->

of.html. Ich führe die `svgView()`-Method in meinem Blog *Broken Links* weiter aus: <http://www.broken-links.com/2012/08/14/better-svg-sprites-with-fragment-identifiers/>.

Die IE-Testdrive-Site bietet ein praktisches *Tool zum Experimentieren mit SVG-Filtereffekten*: [http://ie.microsoft.com/testdrive/graphics/hands-on-css3/hands-on\\_svg-filter-effects.htm](http://ie.microsoft.com/testdrive/graphics/hands-on-css3/hands-on_svg-filter-effects.htm).

Dirk Schulze hat eine gute, prägnante *Einführung zu den geplanten Funktionen von SVG2* geschrieben: <http://dschulze.com/blog/articles/8/new-features-in-svg2/>.

Eine gute Auswahl an *Canvas-Anleitungen* (vom Einsteiger- bis zum Fortgeschrittenenniveau) finden Sie unter <http://www.html5canvas-tutorials.com/>, einen hervorragenden *Spickzettel mit den Eigenschaften und Methoden* auf Jacob Seidelins Blog unter <http://blog.nihilogic.dk/2009/02/html5-canvas-cheat-sheet.html> (letzte Aktualisierung 2009, aber trotzdem noch relevant).

HTML5 Rocks liefert unter <http://www.html5rocks.com/en/tutorials/canvas/imagefilters/> eine *Anleitung zu fortgeschrittenen Bildbearbeitungseffekten*.

Tim Taubert erklärt in seinem Blog-Bertrag *Building a Live Green Screen with getUserMedia() and MediaStreams* die Grundlagen der Live-Bearbeitung von Videobildern: <http://timtaubert.de/blog/2012/10/building-a-live-green-screen-with-getusermedia-and-mediastreams/>.

Der Blog Learning WebGL stellt *Anfängerlektionen zur Arbeit in einem dreidimensionalen Kontext* in canvas bereit unter <http://learning-webgl.com/lessons/>, und WebGL.com bietet regelmäßig *Zusammenstellungen von Demos, Anleitungen und Entwicklertreffen*: <http://www.webgl.com/>.

## Kapitel 8

Das stets hilfreiche MDN liefert einen präzisen und vollständigen *Leitfaden zu den neuen Eingabetypen*, zu finden ist er unter <https://developer.mozilla.org/en-US/docs/HTML/Element/Input/>. PPK bietet detaillierte *Tabellen zur Desktop-Unterstützung* unter <http://www.quirksmode.org/html5/inputs.html>. Das Ganze gibt es auch noch für Mobilgeräte unter [http://www.quirksmode.org/html5/inputs\\_mobile.html](http://www.quirksmode.org/html5/inputs_mobile.html).

Ryan Seddon hat ein *Polyfill für HTML5-Formularfunktionen in Browsern ohne native Unterstützung* geschrieben: Zu finden ist es unter <https://github.com/ryanseddon/H5F/>.

Bruce Lawsons Diskussion zu Autofokus und Barrierefreiheit, *The Accessibility of HTML 5 Autofocus* finden Sie auf seinem Blog unter <http://www.brucelawson.co.uk/2009/the-accessibility-of-html-5-autofocus/>.

HTML5 Rocks hat einen guten *Überblick über die Constraint-Validation-API*, zu finden unter <http://www.html5rocks.com/en/tutorials/forms/constraintvalidation/>.

Rubular ist eine hilfreiche *Anwendung zum Testen regulärer Ausdrücke*. Sie zielt zwar auf Ruby ab, funktioniert aber ebenso gut für JavaScript und HTML5-Formulare: <http://rubular.com/>.

Und Ihr bescheidener Autor hat bei HTML5 Doctor eine *Einführung in CSS3-Pseudoklassen* geschrieben: <http://html5doctor.com/css3-pseudo-classes-and-html5-forms/>.

## Kapitel 9

Den gesamten Text von Steve Jobs *Thoughts on Flash* finden Sie unter <http://www.apple.com/hotnews/thoughts-on-flash/>.

*Hinweise zur Barrierefreiheit von Audio und Video* im Zusammenhang mit automatischer Wiedergabe erhalten Sie auf der Audio-Control-Seite der WCAG unter <http://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-dis-audio.html>.

MDN liefert unter [https://developer.mozilla.org/en-US/docs/Media\\_formats\\_supported\\_by\\_the\\_audio\\_and\\_video\\_elements/](https://developer.mozilla.org/en-US/docs/Media_formats_supported_by_the_audio_and_video_elements/) *Tabellen zum aktuellen Implementations-Status von Medienformaten in verschiedenen Browsern*, und Kroc Camens Technik und Anmerkungen zur Implementation für verschiedene Browser *Video for Everybody!* gibt es auf [http://camendesign.co.uk/code/video\\_for\\_everybody/](http://camendesign.co.uk/code/video_for_everybody/).

Die beste *Einführung zu WebVTT und das track-Element* finden Sie auf Dev.Opera unter <http://dev.opera.com/articles/view/an-introduction-to-webvtt-and-track/>, und auf Anne van Kesterens Website wartet ein nützliches *WebVTT-Validierungswerkzeug*: <http://quuz.org/webvtt/>.

Ich habe in diesem Kapitel drei Codierungswerkzeuge angesprochen: Das *Kommandozeilen-Tool FFmpeg* gibt es auf <http://ffmpeg.org/>, den *Miro Video Converter* unter <http://www.mirovideoconverter.com/> und *HandBrake* unter <http://handbrake.fr/>.

Die *Media-Fragments-Spezifikation des W3C* finden Sie unter <http://www.w3.org/TR/media-frags/>.

MDN bietet die besten *Dokumentationen für die Media-API und Ereignisse*, die ich gefunden habe: <https://developer.mozilla.org/en-US/docs/DOM/HTMLMediaElement/> beziehungsweise [https://developer.mozilla.org/en-US/docs/DOM/Media\\_events/](https://developer.mozilla.org/en-US/docs/DOM/Media_events/).

CreativeJS liefert unter <http://creativejs.com/resources/web-audio-api-getting-started/> einen tollen *Einführungsartikel zur Web-Audio-API*.

Über WebRTC werden Sie in Zukunft sicherlich noch viel mehr hören, Sam Dutton hat bereits eine gute Einführung dazu auf *HTML5 Rocks* geschrieben: <http://www.html5rocks.com/en/tutorials/webrtc/basics/>. Microsofts *Einführung zum CU-RTC-Web-Vorschlag* finden Sie im Interoperability-Blog unter <http://blogs.msdn.com/b/interoperability/archive/2012/07/28/customizable-ubiquitous-real-time-communication-over-the-web-cu-rtc-web.aspx>.

## Kapitel 10

Der *Chrome Web Store* hat eine detaillierte Dokumentation zur Vorbereitung der einzureichenden Apps und natürlich auch zur Manifest-Datei: [https://developers.google.com/chrome/web-store/docs/get\\_started\\_simple/](https://developers.google.com/chrome/web-store/docs/get_started_simple/).

*Näheres zur Manifest-Datei für Firefox Marketplace* finden Sie unter <https://developer.mozilla.org/docs/Apps/Manifest/>. Auch ein nützliches *Tool zur Überprüfung von Manifesten* wird angeboten: <https://marketplace.firefox.com/developers/validator/>. Robert Nyman's Vortrag *Web APIs and Apps* liefert eine hervorragende Einführung zur Entwicklung von Apps für den Firefox Marketplace: <http://www.slideshare.net/robnyman/web-apis-apps-mozilla-london/>.

Die letzte Version der *Widgets-Spezifikation des W3C* liegt unter <http://w3.org/TR/widgets/>, und eine *hilfreiche Einführung* finden Sie auf Peter-Paul Kochs Site – auch wenn sie bereits vor einigen Jahren geschrieben wurde und sich die Spezifikation seither in einigen kleinen Details verändert hat: [http://quirksmode.org/blog/archives/2009/04/introduction\\_to.html](http://quirksmode.org/blog/archives/2009/04/introduction_to.html).

Das *PhoneGap-Projekt* finden Sie unter <http://phonegap.com/>, und die *API* ist unter <http://docs.phonegap.com/> dokumentiert. Den PhoneGap-Artikel *Beliefs, Goals, and Philosophy* können Sie unter <http://phonegap.com/2012/05/09/phonegap-beliefs-goals-and-philosophy/> nachlesen.

*Mehr zum Titanium-Projekt* finden Sie unter <http://appcelerator.com/platform/>.

Jede Smart-TV-Plattform hat ihr eigenes Entwicklerforum, ein gutes *Beispiel für den Einstieg* bietet die Samsung-Site unter <http://www.samsungdforum.com/>.

Der Opera TV Store fasst immer mehr Fuß; mehr darüber erfahren Sie unter <http://business.opera.com/partners/tv/store/>. Die Dev.Opera-Site bietet einige tolle *Referenzartikel zur Gestaltung und Entwicklung für Fernsehgeräte*; diese können sie unter <http://dev.opera.com/tv/> nachlesen.

Das *Webinos-Projekt*, das eine einheitliche Geräte-API anstrebt, befindet sich auf <http://www.webinos.org/>.

Das Mozilla Developer Center verfügt unter [https://developer.mozilla.org/en-US/docs/HTML/Using\\_the\\_application\\_cache/](https://developer.mozilla.org/en-US/docs/HTML/Using_the_application_cache/) über eine hervorragende *AppCache-Dokumentation*, und Mark Christian und Peter Lubbers haben eine nützliche *Seite mit Fakten über AppCache* ins Leben gerufen: <http://appcachefacts.info/>. Alle Pferdefüße von AppCache und Tipps und Techniken zur Verwendung sind in Jake Archibalds Artikel *Application Cache is a Douchebag* bei *A List Apart*: <http://www.alistapart.com/articles/application-cache-is-a-douchebag/> aufgeführt.

## Kapitel 11

Web Components sind in diesem Moment noch ziemlich neu, es gibt dazu also noch nicht so viele Ressourcen. Als Erstes sollten Sie sich dazu die *entwicklerfreundliche Einführung der Verfasser der Spezifikation* ansehen und danach dann vielleicht die *Präsentation von Eric Bidelman*. Beide Quellen vermitteln Ihnen gut die Grundkonzepte. Zu finden sind sie unter <http://dvcs.w3.org/hg/webcomponents/raw-file/tip/explainer/index.html> und <http://html5-demos.appspot.com/static/webcomponents/index.html> (eventuell benötigen Sie hier Google Chrome für die korrekte Anzeige).

Das Shadow DOM ist bisher der am besten implementierte Bestandteil der Web Components und verfügt als solcher über eine umfangreichere Online-Dokumentation. Sowohl *Sitepoint* als auch *HTML5 Rocks* liefern gut verständliche Erklärungen zum Thema, diese finden Sie unter <http://www.sitepoint.com/the-basics-of-the-shadow-dom/> und <http://www.html5rocks.com/en/tutorials/webcomponents/shadowdom/>.

Falls Ihr Browser keine Custom Elements unterstützt, sehen Sie sich *X-Tags* an. Diese von Mozilla entwickelte, experimentelle Bibliothek bildet das Verhalten von Custom Elements nach und bietet dabei ein großes Register vorgefertigter Komponenten: <http://x-tags.org/>.

Im Internet Explorer 10 wurden CSS Regions als Erstes implementiert, die entsprechende *Dokumentation* eignet sich daher zum Einstieg in die Grundlagen. Siehe <http://msdn.microsoft.com/en-us/library/ie/hh673537%28v=vs.85%29.aspx/>.

CSS Exclusions finden sich ebenfalls in IE10, sehen Sie also auch hier zuerst in der *Microsoft-Dokumentation* nach. Anschließend schauen Sie sich ein paar *Demos von Adobe* an. Siehe <http://msdn.microsoft.com/en-us/library/ie/hh673558%28v=vs.85%29.aspx/> und <http://adobe.github.com/web-platform/samples/css-exclusions/>.

Im MDN findet sich die beste Dokumentation von Feature Queries, obwohl die API derzeit noch undokumentiert ist. Siehe <https://developer.mozilla.org/en-US/docs/CSS/@supports/>.

Während ich dies schreibe, ist zu Cascading Variables nur etwas im *Spezifikationsentwurf* unter <http://dev.w3.org/csswg/css-variables/> zu finden.

Falls der *Vorschlag für Box Alignment* immer noch aktuell ist, während Sie dies lesen, können Sie seinen Fortschritt hier verfolgen: <http://dev.w3.org/csswg/css3-align/>. Den Vorschlag zu *Line Grid* finden Sie unter <http://dev.w3.org/csswg/css-line-grid/>.

Håkon Wium Lie und Chris Mills haben eine sehr schöne Einführung zum Seitenkonzept mit CSS geschrieben. Lesen Sie dazu ihren Artikel *Opera Reader: Paging the Web*: <http://people.opera.com/howcome/2011/reader/index.html>. Mehr zu Seitenvorlagen finden Sie auf dem *Adobe Web Platform-Blog* unter <http://blogs.adobe.com/webplatform/2012/05/31/pagination-templates-in-css/>.