Inhalt

[Basic Code Fragments 3](#_Toc360227204)

[Hide pages from search engine 3](#_Toc360227205)

[Give your Pinned Site a tooltip 3](#_Toc360227206)

[Set a default page for your Pinned Site 3](#_Toc360227207)

[Disable link highlighting upon tap in IE10 3](#_Toc360227208)

[Suppress IE6 image toolbar 3](#_Toc360227209)

[Facebook Open Graph data 4](#_Toc360227210)

[Twitter Cards 4](#_Toc360227211)

[Pingbacks 4](#_Toc360227212)

[iOS Web Apps 5](#_Toc360227213)

[Search Engine Optimization 6](#_Toc360227214)

[Simple Basics 6](#_Toc360227215)

[Concerning CMS 6](#_Toc360227216)

[Emulatoren 7](#_Toc360227217)

[iOS SDK für iPhone und iPad 7](#_Toc360227218)

[Android SDK und Anleitung zur Einrichtung verschiedener Geräte 7](#_Toc360227219)

[Opera Mini Simulator (nutzt Proxyserver; auch für iPhone verfügbar) 7](#_Toc360227220)

[Opera Mobile (vollwertiger mobiler Browser) Emulator 7](#_Toc360227221)

[Firefox Mobile (Fennec) für Desktop 7](#_Toc360227222)

[Adobe Software Shadow 7](#_Toc360227223)

[Useful Plugins 8](#_Toc360227224)

[FTScroller for consistent experience on all browsers and devices 8](#_Toc360227225)

[Responsive Images 8](#_Toc360227226)

[Web Shim 8](#_Toc360227227)

[HTML5 9](#_Toc360227228)

[New Elements 9](#_Toc360227229)

[Form Attributes 10](#_Toc360227230)

[Input Types 10](#_Toc360227231)

[Form Validation 11](#_Toc360227232)

[CSS3 12](#_Toc360227233)

[Mediaqueries 12](#_Toc360227234)

[Breakpoints 12](#_Toc360227235)

# Basic Code Fragments

## Hide pages from search engine

According to Heather Champ, former community manager at Flickr, you should not

allow search engines to index your "Contact Us" or "Complaints" page if you

value your sanity. This is an HTML-centric way of achieving that.

<meta name="robots" content="noindex">

```

**WARNING:** DO NOT INCLUDE ON PAGES THAT SHOULD APPEAR IN S

## Give your Pinned Site a tooltip

You know — a tooltip. A little textbox that appears when the user holds their

mouse over your Pinned Site's icon. (Favicon.ico on taskbar)

<meta name="msapplication-tooltip" content="A description of what this site does.">

## Set a default page for your Pinned Site

If the site should go to a specific URL when it is pinned (such as the

homepage), enter it here. One idea is to send it to a special URL so you can

track the number of pinned users, like so:

`http://www.example.com/index.html?pinned=true`

<meta name="msapplication-starturl" content="http://www.example.com/index.html?pinned=true">

## Disable link highlighting upon tap in IE10

Similar to [-webkit-tap-highlight-color](http://davidwalsh.name/mobile-highlight-color)

in iOS Safari. Unlike that CSS property, this is an HTML meta element, and it's

value is boolean rather than a color. It's all or nothing.

<meta name="msapplication-tap-highlight" content="no" />

## Suppress IE6 image toolbar

Kill IE6's pop-up-on-mouseover toolbar for images that can interfere with

certain designs and be pretty distracting in general.

<meta http-equiv="imagetoolbar" content="false">

## Facebook Open Graph data

You can control the information that Facebook and others display when users

share your site. Below are just the most basic data points you might need. For

specific content types (including "website"), see [Facebook's built-in Open

Graph content

templates](https://developers.facebook.com/docs/opengraph/objects/builtin/).

Take full advantage of Facebook's support for complex data and activity by

following the [Open Graph

tutorial](https://developers.facebook.com/docs/opengraph/tutorial/).

<meta property="og:title" content="">

<meta property="og:description" content="">

<meta property="og:image" content="">

## Twitter Cards

Twitter provides a snippet specification that serves a similar purpose to Open

Graph. In fact, Twitter will use Open Graph when Cards is not available. Note

that, as of this writing, Twitter requires that app developers activate Cards

on a per-domain basis. You can read more about the various snippet formats

and application process in the [official Twitter Cards

documentation](https://dev.twitter.com/docs/cards).

<meta name="twitter:card" content="summary">

<meta name="twitter:site" content="@site\_account">

<meta name="twitter:creator" content="@individual\_account">

<meta name="twitter:url" content="http://www.example.com/path/to/page.html">

<meta name="twitter:title" content="">

<meta name="twitter:description" content="">

<meta name="twitter:image" content="http://www.example.com/path/to/image.jpg">

## Pingbacks

Your server may be notified when another site links to yours. The href

attribute should contain the location of your pingback service.

<link rel="pingback" href="">

## iOS Web Apps

There are a couple of meta tags that provide information about a web app when

added to the Home Screen on iOS.

Adding `apple-mobile-web-app-capable` will make your web app chrome-less and

provide the default iOS app view. You can control the color scheme of the

default view by adding `apple-mobile-web-app-status-bar-style`.

<meta name="apple-mobile-web-app-capable" content="yes">

<meta name="apple-mobile-web-app-status-bar-style" content="black">

You can use `apple-mobile-web-app-title` to add a specific sites name for the

Home Screen icon. This works since iOS 6.

<meta name="apple-mobile-web-app-title" content="">

For further information please read the [official documentation](http://developer.apple.com/library/safari/#documentation/AppleApplications/Reference/SafariHTMLRef/Articles/MetaTags.html)

on Apple's site.

Apart from that it is possible to add start-up screens for web apps on iOS. This

basically works by defining `apple-touch-startup-image` with an according link

to the image. Since iOS devices have different screen resolutions it is

necessary to add media queries to detect which image to load. Here is an

example for a retina iPhone:

<link rel="apple-touch-startup-image" media="(max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2)" href="img/startup-retina.png">

However, it is possible to detect which start-up image to use with JavaScript.

The Mobile Boilerplate provides a useful function for this. Please see

[helpers.js](https://github.com/h5bp/mobile-boilerplate/blob/master/js/helper.js#L354)

for the implementation.

# Search Engine Optimization

## Simple Basics

* Remember: Search Engines can only see content at the initial page load 🡪 no dynamic content :/
* Use keywords in your URL’s, separated with a hyphen “www.keyword1-keyword2.com/keyword-keyword-keyword“
* Unique Page titles for every page
* Description meta tag with about 150 to 160 characters
* Noindex meta tag for pages which should not be included in search engine results (Contact, Imprint etc.)
* Key words in <h1> <h2>
* Link anchor text 🡪 If many links point at a page with the same link anchor text, this page is seen as relevant for these words
* Use Sitemaps 🡪 SE understands structure of site and can find updated content 🡪 sitemap.xml / sitemap.txt
* Duplicated Pages such as PDFs or other document types with exactly the same content as on your page should be hidden for SE with scripted links or with rel=”nofollow” attribute
* Ensure that every page has links back to the main navigation to prevent the SE from getting stuck in a dead-end
* Avoid using scripted navigation 🡪 SE’s can’t follow the link structure of your site if the navigation is build with js 🡪 you run the risk of deep content being invisible for the SE

## Concerning CMS

* Prevent the CMS from building dynamic URLs 🡪 URL to a content should always be the same
* Prevent the CMS from building multiple URLs 🡪 each piece of content should only exist on one URL
* No infinite loops! Prevent the SE from getting caught in a potentially endless tast of following “next” links 🡪 Use scripted links which the SE can’t follow 🡪 SE should spend the time on the important content which should be indexed

# Emulatoren

## iOS SDK für iPhone und iPad

https://developer.apple.com/devcenter/ios/

## Android SDK und Anleitung zur Einrichtung verschiedener Geräte

http://developer.android.com/sdk/

http://sangatpedas.com/android-device-emulator-testing-your-mobile-site-on-anyandroid-

device/

## Opera Mini Simulator (nutzt Proxyserver; auch für iPhone verfügbar)

http://www.opera.com/developer/tools/mini/

## Opera Mobile (vollwertiger mobiler Browser) Emulator

http://de.opera.com/developer/tools/mobile/

## Firefox Mobile (Fennec) für Desktop

<http://www.mozilla.org/de/mobile/>

## Adobe Software Shadow

Eine beliebige Website auf seinem Smartphone anzeigen lassen

und mit dem *Chrome Inspector* deren Bestandteile direkt vom Rechner aus anzeigen lassen.

http://labs.adobe.com/technologies/shadow/

# Useful Plugins

## FTScroller for consistent experience on all browsers and devices

<https://github.com/ftlabs/ftscroller/commits/master>

## Responsive Images

<https://github.com/scottjehl/picturefill>

## Adaptive Images (Empfehlung aus Zilligens Buch)

<http://adaptive-images.com/>

<https://github.com/mattwilcox/Adaptive-Images>

Bei dieser von Matt Wilcox erstellten Variante handelt es sich um eine Kombination

aus server- und clientseitigen Abfragen, die dafür sorgen, dass je nach Bildschirmbreite ein Bild in der passenden Größe ausgeliefert wird.

Der große Vorteil dieser Methode besteht darin, dass der HTML-Code für die Einbettung der Bilder nicht angefasst werden muss. Das kommt Ihnen vor allem dann

zugute, wenn Sie auf einer bestehenden Website bereits zahlreiche Bilder integriert

haben oder wenn aufgrund eines CMS der Code der Bildausgabe nicht verändert

werden kann.

Zur korrekten Funktion braucht es serverseitig PHP5, die GD-Bibliothek und

ModRewrite, was heutzutage aber Standard sein sollte.

Sehen wir uns die Funktionsweise anhand eines Beispiels an. Auf der Website zum

Buch sind auf der Startseite einige Bilder, die bisher lediglich per CSS verkleinert

wurden, aber von den Bildmaßen her für mobile Geräte reichlich überdimensioniert

daher kommen.

Für Retina Displays:

<http://adaptive-images.com/details.htm#alternate-js>

Bei Adaptive-Images gibt es dafür ein alternatives JavaScript, das neben der

aktuellen Bildschirmgröße auch noch die Pixeldichte in das zum Server übertragene

Cookie speichert und das ausgelieferte Bild darauf abstimmt:

<script>document.cookie='resolution='+Math.max(screen.width,screen.

height)+("devicePixelRatio" in window ? ","+devicePixelRatio : ",1")+';

path=/';</script>

An die Grenzen stößt dieses Konzept, wenn es darum geht, individuelle Bildausschnitte für die einzelnen Kontexte anzulegen, also das Motiv gänzlich

auszutauschen Ein weiterer Nachteil besteht darin, dass das größte Bild die

Ausgangsposition ist. Funktioniert das Script nicht, ist zwar ein Fallback da, der

Nutzer leidet dennoch unter langen Ladezeiten.

## Flexslider für Bildergalerien

Is responsible (also concerning gestures) and can be used with Wordpress.

<http://www.woothemes.com/flexslider/>

## Web Shim

<http://afarkas.github.com/webshim/demos/>

* Plugin im head Bereich des HTML Dokuments platzieren!
* Form Validation:

HTML5 Input Attributes

HTML5 Input Types

* Folgende Module können ebenfalls angehackt werden:

HTML5 Canvas

HTML5 Audio

HTML5 Video

HTML5 localStorage

HTML5 sessionStorage

Misc Geolocation API

## Swipe for Off-Canvas Layouts

<https://github.com/torkiljohnsen/swipe>

HTML5 Videoplayer (WordPress)

Fancy aussehen // mit blauen Ladebalken?

<http://www.videojs.com/>

<http://wordpress.org/plugins/videojs-html5-video-player-for-wordpress/installation/>

Gut aussehender Player, mit der Möglichkeit, verwandte Videos (Beiträge, Sendungen) rechts // mit Kommentar oder unten //ohne Kommentar, anzeigen zu lassen:

<http://codecanyon.net/item/html5-video-player-wordpress-plugin/1613464>

# HTML5

## New Elements

Durch die neuen strukturellen Elemente in HTML5 können zum Beispiel Screenreader ein Dokument besser scannen und dem Nutzer dann den Hauptinhalt zuerst vorlesen oder den Nutzer zur Navigation lotsen.

1. header
2. footer
3. nav

* Kann auch im header Element stehen
* Mit nav-Element auszuzeichnen:
* Hauptnavigation
* Subnavigation
* Navigation innerhalb einer Seite
* Suchfunktion der Website

1. aside

* wird häufig für Seitenleisten oder andere ergänzende Inhalte verwendet, die indirekt in Bezug zum Hauptinhalt stehen, aber nicht unbedingt erforderlich sind
* Ist der Bereich zwar verwandt zum Inhalt, aber eine eher unwichtige Ergänzung, ist aside das passende Element.

1. figure

* wird für Bilder und andere erläuternde Darstellungen verwendet, wahlweise auch mit einer Bildunterschrift figcaption. Bedingung für den Einsatz von figure ist, dass das Element aus dem Inhaltsfluss entfernt werden kann, ohne dessen Bedeutung einzuschränken.
* Das Bild ergänzt den Text, ist aber nicht zwingend für dessen Verständnis erforderlich.
* Ist der Bereich eine wichtige Ergänzung zum Hauptinhalt, könnte aber auch an anderer Stelle des Dokuments positioniert werden, ist figure die richtige Wahl.

1. article

* der Inhalt des article-Elements kann für sich allein stehen und ist auch außerhalb seines Kontextes verständlich. Ist das nicht der Fall, ist das section-Element die bessere Wahl.

1. section

* Bereich eines Dokuments, der thematisch zusammengehört und üblicherweise mit einer entsprechenden Überschrift versehen ist.
* ! Geht es nur darum, einen Bereich aus Styling-Gründen zu gruppieren, ist nach wie vor das div-Element die richtige Wahl.

## Form Attributes

1. Placeholder=““

* Hier können wir nun in HTML5 einen Wert ergänzen, der im Eingabefeld erscheint und beim Anklicken ausgeblendet wird

1. Required
2. Autofocus

* Feld wird fokussiert ohne vom User angeklickt zu werden (Pro Formular nur einmal verwenden)

1. Pattern=““

* Regulären Ausdrücken Informationen zur Prüfung hinterlegen, die als Grundlage für die Validierung dienen
* Beispiele:

Für Telefonnummern:   
<input type="tel" pattern="[\+]?([0-9/.\(\) \-]{6,})+" />

Für PLZ:

<input type="text" pattern="[0-9]{5}" />

1. Title=””

Wenn wir bei der Postleitzahl Folgendes ergänzen <input type="text" pattern="[0-9]{5}" **title="Die Postleitzahl muss aus fünf Ziffern bestehen"** /> dann wird der Hinweis aus dem title entsprechend bei falscher Eingabe mit angezeigt

## Input Types

* Bieten verschiedene kontextbedingte Tastaturen
* Werden vom Browser validiert

1. <input type="email" />
2. <input type="url" />
3. <input type="tel" />
4. <input type="number" min="100" max="700" step=”50” />
5. <input type=”search”/>
6. <input type=”date”/>
7. <input type=”time”/>

Wem aber die einzelnen Minutenschritte nicht passend erscheinen, der kann mit dem step-Attribut nachhelfen. Der Wert wird hierbei in Sekunden angegeben. Möchte ich zum Beispiel 30-Minuten-Schritte einstellen, gebe ich einen Schrittwert von 1800 an (30 x 60 Sekunden)

1. Schieberegler mit Ausgabe des angesteuerten Wertes  
   (Leider funktioniert er noch nicht im Firefox und im Internet Explorer)

<input type="range" min="1000" max="10000" step="500" value="5500"

oninput="budget.value=value" />

<output name="budget">5000</output>

## Form Validation

Durch die eben vorgestellten neuen HTML5 Elemente und Attribute können schon viele Felder von den modernen Browsern validiert werden ohne hinzutun von JS oder PHP. Um dies für alle Browser zu gewährleisten hilft das [Web Shim](#_Web_Shim) Plugin aus.

# CSS3

## Mediaqueries

### Breakpoints

* 320 Pixel (iPhone Hochformat)
* 480 Pixel (iPhone Querformat)
* 649 (Kleines Tablet Querformat)
* 768 Pixel (iPad Hochformat)
* 1024 Pixel (iPad Querformat)
* 1200 Pixel (Laptop)
* 1600 Pixel (Mega-Displays und TV)

## Graphics and Patterns

<http://lea.verou.me/css3patterns/>

The patterns themselves should work on Firefox 3.6+, Chrome, Safari 5.1, Opera 11.10+ and IE10+.

# Techniques

## SVG Images for Resolution independent Graphics

<http://coding.smashingmagazine.com/2012/01/16/resolution-independence-with-svg/>

## Off-Canvas Layout

### Tutorial

<http://jasonweaver.name/lab/offcanvas/>