



# HTML5 Mobile Development Cookbook

Shi Chuan



## Chapter No. 1 "HTML5 and the Mobile Web"

## In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.1 "HTML5 and the Mobile Web"

A synopsis of the book's content

Information on where to buy this book

## About the Author

**Shi Chuan** has over five years experience in web development. He is a member of the HTML5 Boilerplate project, lead developer of Mobile Boilerplate (<http://h5bp.com/mobile>), owner of the JavaScript Patterns project (<http://shichuan.github.com/javascript-patterns>). He is now an independent developer living in the UK and China. You can find out more about him on his personal website at: <http://www.blog.highb.com>. He loves reading, travelling, great food, and eclectic and indie music.

---

I would like to thank my parents, and the whole family who have been my positive and unconditional supporters. I would also like to thank Jiang Xue, who taught me so many things about life, in ways she does not even know.

I would also like to thank my friends from the Boilerplate Team - Paul Irish, Divya Manian, Mathias Bynens, and Nicolas Gallagher. Former CTO of the company I worked for - Chi Tran. They have been and will always be my inspiration and aspiration.

---

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

# HTML5 Mobile Development Cookbook

How do I create fast and responsive mobile websites that work across a range of platforms? For developers dealing with the proliferation of mobile devices each with unique screen sizes and performance limitations, it is an important question. This cookbook provides the answer. You will learn how to apply the latest HTML5 mobile web features effectively across a range of mobile devices.

*HTML5 Mobile Development Cookbook* will show you how to plan, build, debug, and optimize mobile websites. Apply the latest HTML5 features that are best for mobile, while discovering emerging mobile web features to integrate in your mobile sites.

Build a rock-solid default mobile HTML template and understand mobile user interaction. Make your site fast and responsive, leveraging the uniqueness of location-based mobile features and mobile rich media. Make your mobile website perfect using debugging, performance optimization, and server-side tuning. The book finishes with a sneak preview of future mobile web technologies.

## What This Book Covers

*Chapter 1, HTML5 and the Mobile Web*, introduces HTML5 and the mobile web, along with some emulators and simulators.

*Chapter 2, Mobile Setup and Optimization*, discusses various mobile setups and optimization, such as preventing text resize and optimizing viewport width.

*Chapter 3, Interactive Media with Mobile Events*, discusses mobile interactions such as gesture events.

*Chapter 4, Building Fast and Responsive Websites*, talks about the various ways to make mobile websites fast and responsive.

*Chapter 5, Mobile Device Access*, discusses location-based mobile web and other HTML5 device-specific features.

*Chapter 6, Mobile Rich Media*, talks about the HTML5 rich media elements that can be used on mobile browsers.

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

*Chapter 7, Mobile Debugging*, teaches you ways to work around mobile screen-side constraints and debug mobile websites and web apps effectively.

*Chapter 8, Server-Side Tuning*, focuses on the server-side tuning for mobile websites.

*Chapter 9, Mobile Performance Testing*, teaches you various Tools and techniques that can be used to boost mobile performance.

*Chapter 10, Emerging Mobile Web Features*, talks about ECMAScript 5 as well as mobile-specific features that were added to allow more functionalities with mobile and boost the performance.

**For More Information:**

**[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)**

# 1

## HTML5 and the Mobile Web

In this chapter, we will cover:

- ▶ Getting your mobile device ready
- ▶ Emulators and simulators
- ▶ Setting up the mobile development environment
- ▶ Using HTML5 on mobile web
- ▶ Making HTML5 render cross browser
- ▶ Designing for mobile
- ▶ Determining your target mobile devices
- ▶ Defining a content adaptation strategy

### Introduction

Both HTML5 and mobile web are promising technologies. Both have relatively short histories. In this chapter, most topics we will be covering are relatively basic. This is to help you get started with mobile development quickly and with minimum effort.

Both mobile and HTML5 are still evolving in nature and there could be many questions in your mind. We will clear those doubts and set your mind focused on things that matter.

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

The mobile web is growing fast. We now have mobile Safari which is one of the most used apps on the iPhone, allowing developers to build high performance web applications and enhancing users' browsing experience. You do not need a developer account to host and run a mobile site, you don't need to get approval from any app market to launch a mobile website and you can make updates any time you like without the hassle of waiting for approval. All these are benefits of mobile web development, but at the same time, there are challenges such as inconsistencies among browsers, the lack of certain features compared to native apps, and security. We can't tackle them all, but we sure can solve some of them. And we will see, when developing a mobile site, how we can separate the common practices from the best practices.

There are literally thousands of smartphones out there; you don't need every single one of them to test your application on. In fact, you may need fewer than 10. If that's still out of your budget, then two devices are good enough. For the rest, you can use simulators/emulators to do the job. This book focuses on six A-grade mobile devices, with focus specifically on iPhone, Android, and Windows Phone:

- ▶ iOS
- ▶ Android
- ▶ Windows Mobile
- ▶ Blackberry v6.0 and above
- ▶ Symbian 60
- ▶ Palm webOS

There are two browsers that are device-independent that will also be covered in this book. They are:

- ▶ Opera Mobile
- ▶ Firefox Mobile

Just because other browsers are not in the list does not mean they won't be covered by the issues and techniques we discuss in this book.

## Identifying your target mobile devices

Target browser: all

You can't possibly make a mobile site for every single mobile device. No one has the time or energy to do so.

Cross-browser mobile web development can be crazy. It's hard to define the scope of the work, as *John Resig* (creator of jQuery Mobile) pointed out in his presentation slide describing his experience building jQuery Mobile (<http://www.slideshare.net/jeresig/testing-mobile-javascript>), he asked three questions:

- ▶ Which platforms and browsers are popular?
- ▶ Which browsers are capable of supporting modern scripting?
- ▶ What devices and simulators do I acquire to test with?

When you are building a mobile site, you have to ask yourself similar questions, but not the exact same questions, because remember your site should be specifically tailored to your target audience. So your questions should be:

- ▶ Which platforms and browsers are most commonly used by visitors on my website?
- ▶ How many people access my website from a mobile device that is capable of supporting modern scripting?
- ▶ Which devices and simulators do I need for testing?

### **Which platforms and browsers are most commonly used by visitors on my website?**

Now let's answer the first question. Before building a mobile website, you must first find out who are your target audience, and what mobile devices they use when visiting your site. There are many analytics tools that can help you answer these questions. One of those is Google Analytics. You can sign up for a free Google Analytics account at: <http://www.google.com/analytics/>

The way to do it is fairly straightforward: most developers are no strangers to Google Analytics. All you have to do is to include the JavaScript snippet from the the Google Analytics site and embed it in your web pages.

JavaScript can be rendered in most modern smartphones, so there is really no difference between using it on a desktop site and on a mobile site.

### **How many people access my website from a mobile device that is capable of supporting modern scripting?**

Now let's answer the second question. One thing you may want to find out is the number of people using mobile browsers to surf your site. And you also want to find out how many people use a relic mobile browser that doesn't support JavaScript at all. This is because if the percentage of people using low-end smartphones is higher than that of people using high-end smartphones, it may not be worthwhile using HTML5 in the first place (although the chance of this is very low).

So if your goal is not just to know the number of people using smartphones, but also the number of people who use older versions of mobile phones, Google Analytics for mobile comes to the rescue. You can download the script from:

[http://code.google.com/mobile/analytics/download.html#Download\\_the\\_Google\\_Analytics\\_server\\_side\\_package](http://code.google.com/mobile/analytics/download.html#Download_the_Google_Analytics_server_side_package)

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

Google Analytics for mobile server-side packages currently supports JSP, ASPX, Perl, and PHP. Let's take a look at one of the examples in PHP. All you have to do is to change the **ACCOUNT ID GOES HERE** with your GA account ID. But remember to **replace 'UA-xx' with 'MO-xx'**.

Unfortunately, when you use the server-side version, you can't use it on pages where you also use the standard JavaScript tracking code, `ga.js`. Using the server-side version means that you have to give up the JavaScript version. It can be annoying because the JavaScript version provides a lot of dynamic tracking mechanisms that are lacking in the server-side version:

```
<?php
// Copyright 2009 Google Inc. All Rights Reserved.
$GA_ACCOUNT = "ACCOUNT ID GOES HERE";
$GA_PIXEL = "ga.php";

function googleAnalyticsGetImageUrl() {
    global $GA_ACCOUNT, $GA_PIXEL;
    $url = "";
    $url .= $GA_PIXEL . "?";
    $url .= "utmac=" . $GA_ACCOUNT;
    $url .= "&utmz=" . rand(0, 0x7fffffff);

    $referer = $_SERVER["HTTP_REFERER"];
    $query = $_SERVER["QUERY_STRING"];
    $path = $_SERVER["REQUEST_URI"];

    if (empty($referer)) {
        $referer = "-";
    }
    $url .= "&utmr=" . urlencode($referer);

    if (!empty($path)) {
        $url .= "&utmp=" . urlencode($path);
    }

    $url .= "&guid=ON";

    return $url;
}
?>
```



## Alternatives to Google Analytics

Google Analytics is not the only mobile analytics service in the market. There are other services providing more specialized services. For example, **PercentMobile** is a hosted mobile analytics service that makes your mobile audience and opportunities clear. You can find out more about this service at:

<http://percentmobile.com/>

### Accuracy of Google Analytics

The location reported by mobile devices may not always be accurate; Google Analytics uses IP addresses to determine user location for Map Overlay reports. They are subject to possible inaccuracy, as mobile IPs originate from the wireless carrier gateway which is not necessarily co-located with mobile users.

### Server loading speed concern

Due to the server-side processing, some additional server load may be incurred. Google recommends you first test the snippet on a few of your pages to ensure all is well before rolling out to an entire site.

## Setting up mobile development tools

Target browser: all

Now one question still remains unanswered from the previous recipe: *which devices and simulators do I need for testing?* We will find this out in this chapter.

If you have figured out what major mobile devices you are going to support, now is the time to see how to set them up. Mobile development can be costly if you test on various mobile devices. Although we have all these mobile simulators and emulators available for testing, it's not as good as testing on a real device. Now let's see how we can maximize the testing coverage and minimize the cost.

### Getting ready

We are going to make some assumptions here. Each case is different, but the idea is the same. Let's assume you have a Windows operating system on your desktop, but the top visitors to your site surf using iOS, Android, and Blackberry.

## How to do it...

Your goal is to maximize the coverage and minimize the cost. All three devices have emulators, but not all support different platforms.

Name	Compatibility
iOS simulator	Mac
Android emulator	Windows, Mac, Linux
Blackberry simulator	Windows

As you can see, since iOS simulator only works for Mac, and if you are running a Windows OS, the best and only choice is to purchase an iPhone for testing. For Android and Blackberry, because they both have emulators for Windows, to save budget, you can download the emulators.

## How it works...

1. List the top mobile devices people use to surf your site.
2. Know the machine OS you are using for the development.
3. Find out the compatibility of each of these device emulators to your development environment.

## There's more...

If you have the budget for more than one mobile device having a different OS, you can think further about screen sizes and the DPI of the mobile device. You may not need to buy two high-end devices. For instance, it's not necessary to own an iPhone4 and an Android Thunderbolt. You can buy a lower-end of Android just to test out how your site would look on a lower-end device. So the idea is to combine your OS, mobile devices, and emulators to maximize the scenarios to cover.

## Device simulator/emulator download lookup table

The following table shows a list of popular mobile device emulators for mobile web design and development testing:

Name	Type	Compatibility	URL
iOS	Simulator	Mac	<a href="https://developer.apple.com/devcenter/ios/index.action#downloads">https://developer.apple.com/devcenter/ios/index.action#downloads</a>
Android	Emulator	Mac, Win, Linux	<a href="http://developer.android.com/sdk/index.html">http://developer.android.com/sdk/index.html</a>

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

Name	Type	Compatibility	URL
HP webOS	Virtual Machine	Mac, Win, Linux	<a href="http://developer.palm.com/index.php?option=com_content&amp;view=article&amp;id=1788&amp;Itemid=55">http://developer.palm.com/index.php?option=com_content&amp;view=article&amp;id=1788&amp;Itemid=55</a>
Nokia Symbian	Emulator	Win	<a href="http://www.forum.nokia.com/info/sw.nokia.com/id/ec866fab-4b76-49f6-b5a5-af0631419e9c/S60_All_in_One_SDKs.html">http://www.forum.nokia.com/info/sw.nokia.com/id/ec866fab-4b76-49f6-b5a5-af0631419e9c/S60_All_in_One_SDKs.html</a>
Blackberry	Emulator	Win	<a href="http://us.blackberry.com/developers/resources/simulators.jsp">http://us.blackberry.com/developers/resources/simulators.jsp</a>
Windows Mobile 7	Emulator	Win	<a href="http://www.microsoft.com/downloads/en/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce">http://www.microsoft.com/downloads/en/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce</a>

### Browser simulator/emulator download lookup table

Apart from device testing tools, we also have tools for platform-independent browsers, notably Opera and Firefox. These are shown in the table below:

Name	Type	Compatibility	URL
Opera Mobile	Emulator	Mac, Win, Linux	<a href="http://www.opera.com/developer/tools/">http://www.opera.com/developer/tools/</a>
Opera Mini	Simulator	Mac, Win, Linux	<a href="http://www.opera.com/developer/tools/http://www.opera.com/mobile/demo/">http://www.opera.com/developer/tools/http://www.opera.com/mobile/demo/</a>
Firefox for Mobile	Simulator	Mac, Win, Linux	<a href="http://www.mozilla.com/en-US/mobile/download/">http://www.mozilla.com/en-US/mobile/download/</a>

### Remote testing

Apart from emulators and simulators, there are also test frameworks that give you remote access to REAL devices. One of those tools is **DeviceAnywhere**; one problem is that it is not free.

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

## BlackBerry simulator

Target Browser: BlackBerry

Most mobile device simulators are easy to install and configure if you follow the instructions on their sites, but BlackBerry simulators work differently from other mobile device simulators. For Blackberry device simulators, to connect to Internet, besides downloading the simulators, you also need to download and install **BlackBerry Email and MDS Services Simulator**.

For More Information:

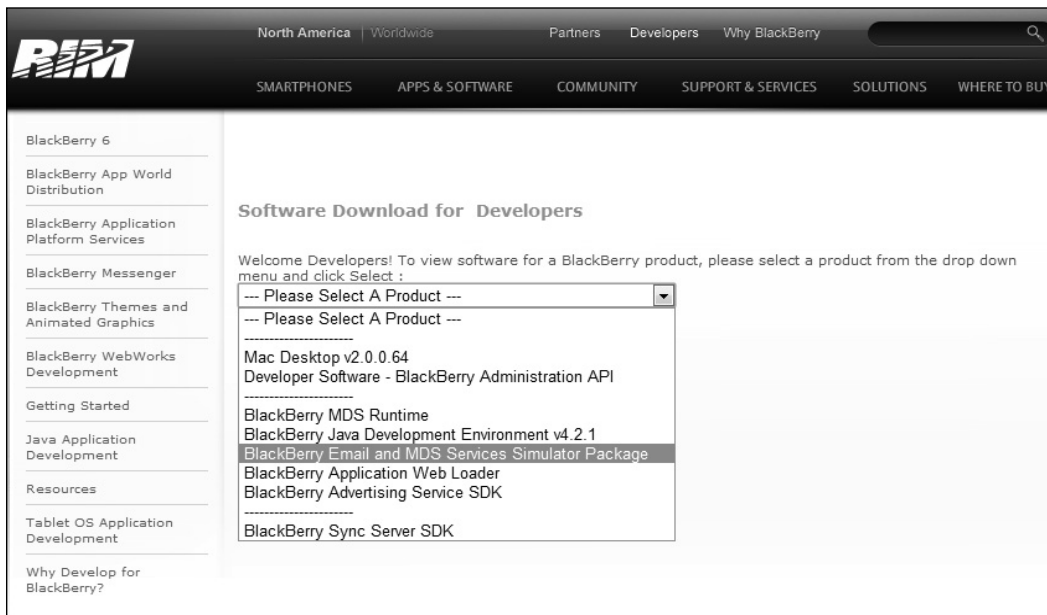
[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

## Getting ready

Make sure you have chosen a simulator to download from: <http://us.blackberry.com/developers/resources/simulators.jsp>

## How to do it...

First, go to the page: <https://swdownloads.blackberry.com/Downloads/entry.do?code=A8BAA56554F96369AB93E4F3BB068C22&CPID=OTC-SOFTWAREDOWNLOADED&cp=OTC-SOFTWAREDOWNLOADED>. There you will see a list of products similar to the following screenshot:



Now select **BlackBerry Email and MDS Services Simulator Package** and then click on **Next**.

After downloading and installing the software, you must first launch the service simulator before the BlackBerry simulator, in order to allow it to connect to the Internet.

The following is a screenshot of a Blackberry simulator:



## Setting up the mobile development environment

Target browser: all

Before we start mobile web development, we have to first set up a development environment.

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

## Getting ready

1. Set up localhost on your machine. For Windows, Mac, or Linux, the easiest way to set it up is to use the popular and free XAMPP software: (<http://www.apachefriends.org/en/index.html>).
2. Make sure you have a wireless connection.
3. Also you should have a mobile device with you. Otherwise, use a mobile simulator/emulator.
4. Ensure your mobile device and your desktop are on the same wireless network.

## How to do it...

1. Create an HTML file and name it `ch01e1.html` at the root directory of your localhost:

Inside `ch01r01.html`, type in the following:

```
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    <header>
      Main Navigation here
    </header>
    body content here
    <footer>
      Footer links here
    </footer>
  </body>
</html>
```

2. Now get your IP address. If you are using Windows, you can type the following line in your command prompt:

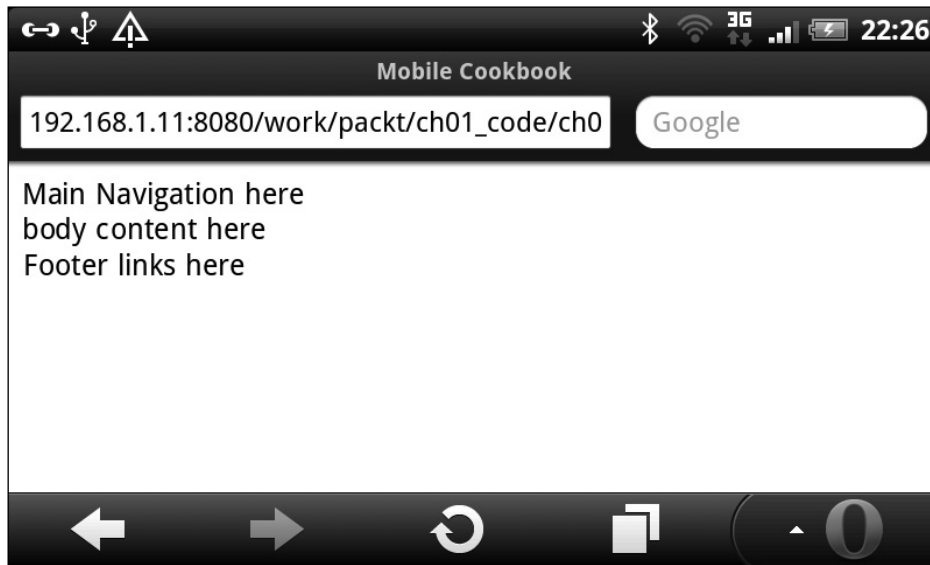
```
ipconfig
```

### Downloading the example code for this book



You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you

3. Once you have got your IP address, (for example, 192.168.1.16.), enter it in your mobile browser URL address bar. Now you should see the page load with the text displayed:



### How it works...

Within the same network, your mobile device can access your desktop host through your desktop IP address.

### There's more...

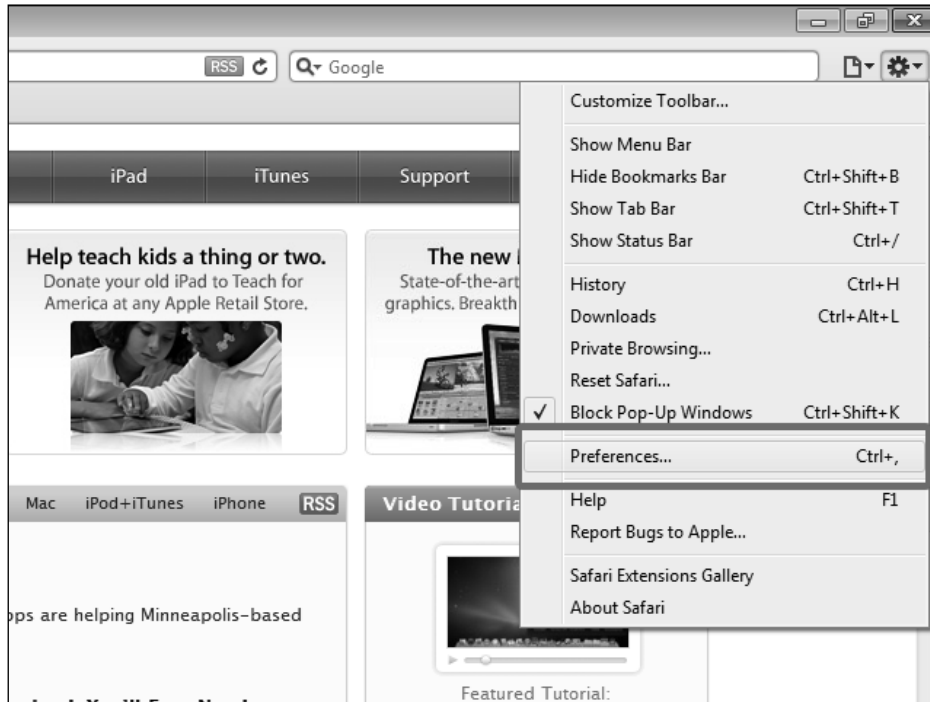
If you don't have a mobile device, you can use one of the simulators for testing. But it's recommended to have at least one or two real mobile devices for testing. A simulator could test most things, but not everything, accurately.

#### For More Information:

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

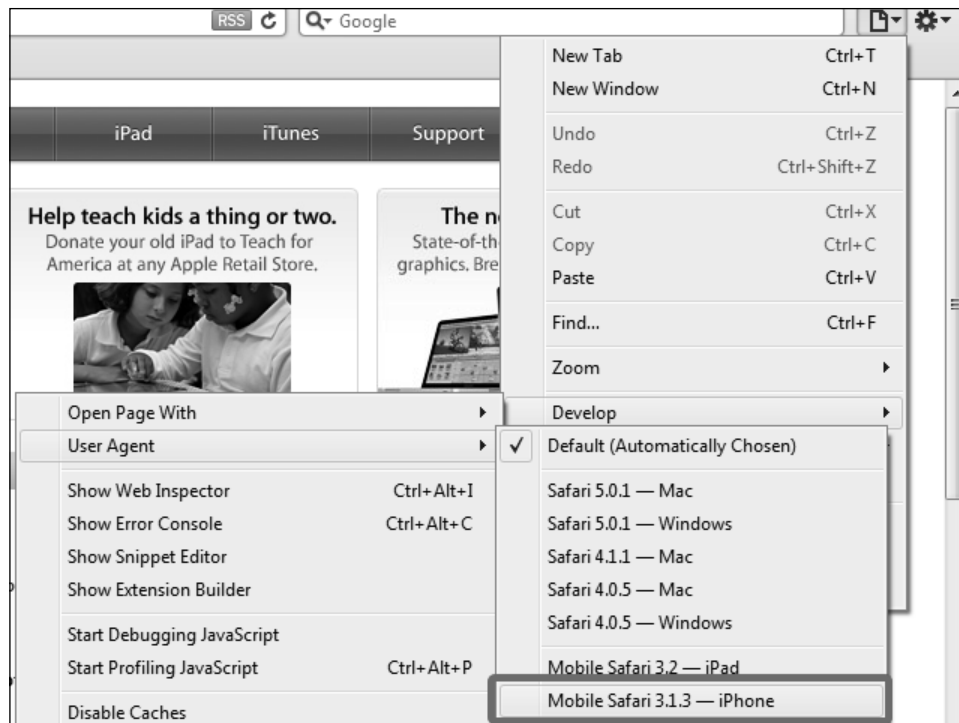
## Testing on a Safari desktop

If your main target audience is iPhone mobile Safari users, you can also do testing on a desktop to save time. To do so, open up Safari, go to **Preferences**, click on the **Advanced** tab, check **Show Develop menu bar** as shown next:



Now click on the display menu for the current page, choose **Develop** | **User Agent** | **Mobile Safari 3.1.3 – iPhone**:





### List of community-based collection of emulators/simulators

There is a list of emulators and simulators available if you really don't have a Smartphone at hand. You can find the list on a wiki on the Mobile Boilerplate project:

<https://github.com/h5bp/mobile-boilerplate/wiki/Mobile-Emulators-&-Simulators>

### List of emulators/simulators collection by Firtman

*Maximiliano Firtman*, a mobile and web developer, also an author, maintains a list of emulators on his site at:

<http://www.mobilexweb.com/emulators>

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

## Using HTML5 on the mobile web

Target browser: all

Now we are going to create a simple HTML5 page for your mobile device. If you already have experience with older versions of HTML, HTML5 should be easy to understand. And if you have made a web page for desktop before, it won't be hard for you to make one for a mobile device.

### Getting ready

Create a new file `ch01e2.html`.

### How to do it...

Save the following code in the file:

```
<!doctype html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  </head>
  <body>
    hello to the HTML5 world!
  </body>
</html>
```

Now render this in your mobile browser, and you should see the text render just as expected.

### How it works...

As you can see, the only difference between HTML5 and other HTML pages is the **Document Type Definition (DTD)** we used: `<!doctype html>`.

You might have seen the code `<meta name="viewport" content="width=device-width, initial-scale=1.0">` and are wondering what it does. It helps Mobile Safari to know that the page is as wide as the device. Setting `initial-scale=1` tells the browser not to zoom in or out.

## There's more...

Here's a little bit of history of HTML5: there are two versions of HTML5 draft, one created by W3C and the other by WHATWG. The W3C is run by a group that is democratic in nature, but super slow in practice. WHATWG is edited by one person, *Ian Hickson* (who is also working for Google), and a group of people who are not public. As *Ian* does most of the decision making, it makes WHATWG's version progress faster.

## HTML5 and version number

You might be wondering why HTML5 is being so ambiguous by using a declaration without even a version number. Well there are many reasons to justify this:

1. Version support of HTML doesn't matter much to browsers. What matters is the feature support. In other words, if the browser supports the HTML5 feature you are using, even if you declare the document as HTML4, it will still render the HTML5 element as expected.
2. It's easier to type!

## Mobile doctypes

One question you may ask is whether it is safe to use the HTML5 DTD `<!doctype html>`. The answer is DTDs were made for validation, not browser rendering. Your next question might be: what about quirks mode? `<!doctype html>` is the minimum information required to ensure that a browser renders in standards mode. So you are pretty safe to use `<!doctype html>`.

You may have noticed that we use `<!doctype html>` instead of `<!DOCTYPE html>`. The reason is HTML5 is not case sensitive, but for consistency with other HTML tags, we will use lowercase throughout the book.

## Free resources to learn HTML5

There are many excellent and free books, and articles about basic HTML5 tags. If you are unfamiliar with HTML5, you can check out one of the following:

- ▶ HTML5 Doctor: <http://html5doctor.com/>
- ▶ Dive Into HTML5: <http://diveintohtml5.org/>
- ▶ HTML5 Rocks: <http://www.html5rocks.com/>

If you are the kind of person who really wants to know every detail about something, you can read the official HTML5 specs.

The W3C version of the spec is at:

<http://dev.w3.org/html5/spec/Overview.html>

The WHATWG version of HTML Living Standard is at:

<http://www.whatwg.org/specs/web-apps/current-work/multipage/>

## Rendering HTML5 across different browsers

Target browser: all

There are older mobile browsers that don't recognize HTML5 elements. The problem with this is that you can't style these elements if they are not recognized. There are many shims made to tackle this issue. One of those is Modernizr.

### Getting ready

1. One of the mobile browsers that doesn't recognize HTML5 elements is Windows Mobile. If you don't have Windows Mobile, you can simply use IE7 to test this out, because they are based on the same engine.
2. Download Modernizr from the site: <http://www.modernizr.com/>. It was written by *Faruk Ateş*, *Paul Irish*, and *Alex Sexton*.

### How to do it...

1. Create an HTML file and name it `ch01e3.html`, and enter the following code in the file:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <style>
      header, footer {display:block;}
    </style>
  </head>
  <body>
    <header>
      Main Navigation here
    </header>
```

```

    body content here
  <footer>
    Footer links here
  </footer>
</body>
</html>

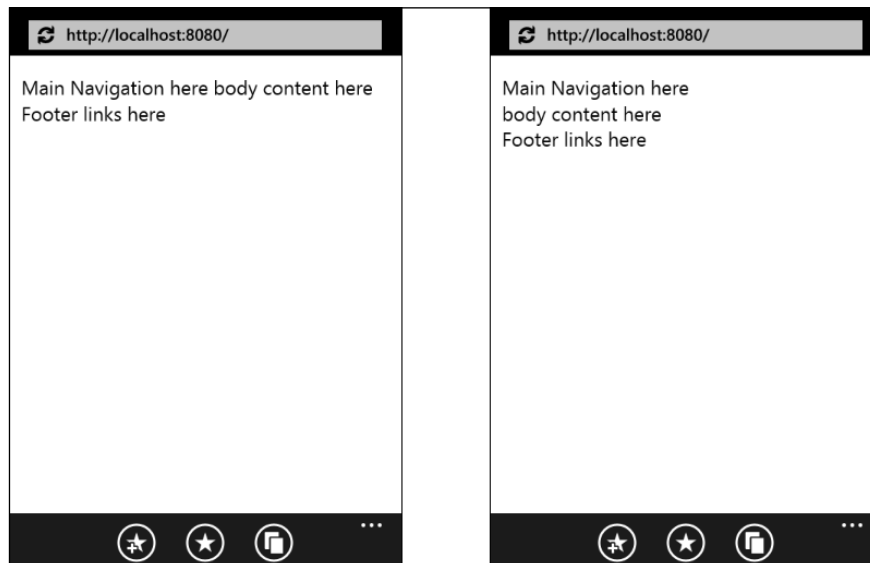
```

2. Now create another page with Modernizr included, and name it `ch01e4.html`:

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <script src="modernizr-1.7.min.js"></script>
    <style>
      header, footer {display:block;}
    </style>
  </head>
  <body>
    <header>
      Main Navigation here
    </header>
    body content here
    <footer>
      Footer links here
    </footer>
  </body>
</html>

```



## How it works...

Remember that if you use Modernizr for your project, you should always include it at the head of the file before the end of the `<head>` tag. There are other polyfills used for similar purposes, and some are listed in the following section.

## There's more...

Modernizr is not the only script library helper out there; there are two other notable ones:

- ▶ **html5shim** by *Remy Sharp, Jonathan Neal & community*, enabled for print use, as well as at:  
<http://code.google.com/p/html5shim/>
- ▶ **innerShiv** by *Joe Bartlett*, enables elements for innerHTML use at:  
<http://jdbartlett.github.com/innershiv/>

## HTML5 CSS reset

You may want to reset the set of new CSS HTML5 elements in your stylesheet:

```
article, aside, canvas, details, figcaption, figure, footer, header,
hgroup, menu, nav, section, summary, time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline;
}
```

## Enable block-level HTML5 elements in older IE

In your CSS, you might want to include the set of block-level HTML5 elements in your CSS reset. Do note that not all HTML5 elements have to be displayed as block elements.

Here is a list of block-level HTML5 elements:

```
article, aside, details, figcaption, figure, footer, header, hgroup,
menu, nav, section {
  display: block;
}
```

## Modernizr

Modernizr does more than just making HTML5 elements stylable in CSS. It also helps to detect HTML5 feature support in the browser used for rendering. With version 2.0, you will have the option to customize the download <http://www.modernizr.com/download/>.

## Designing for mobile

Target browser: all

For desktop design, one tends to use either fixed or fluid layout. On the mobile devices, one should almost always use fluid layout. Fluid layout could make your site responsive to browser resize.

### Getting ready

Now create two empty HTML files in your text editor, name one of them `ch01r06_a.html` and the other `ch01r06_b.html`.

### How to do it...

1. In `ch01r06_a.html`, enter the following code and save the file:

```
<!doctype html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0, maximum-scale=1.0">
    <script src="modernizr-1.7.min.js"></script>
    <style>
      body, #main ul, #main li, h1 {
        margin:0; padding:0;
      }
      body {
        background:#FFFFA6;
      }
      #container {
        font-family:Arial;
        width:300px;
        margin:0 auto;
      }
      header, footer {
        display:block;
      }
      #main li{
        list-style:none;
        height:40px;
        background:#29D9C2;
        margin-bottom:0.5em;
        line-height:40px;
```

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

```
-moz-border-radius: 15px;
-webkit-border-radius: 15px;
border-radius: 15px;
}
#main li a {
  color:white;
  text-decoration:none;
  margin-left:1em;
}
</style>
</head>
<body>
  <div id="container">
    <header>
      <h1>Title here</h1>
    </header>
    <nav id="main">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Contact Us</a></li>
        <li><a href="#">Location</a></li>
        <li><a href="#">Product</a></li>
        <li><a href="#">About</a></li>
      </ul>
    </nav>
    <footer>
      Footer links here
    </footer>
  </div>
</body>
</html>
```

2. In ch01r06\_b.html, enter the following code and save the document:

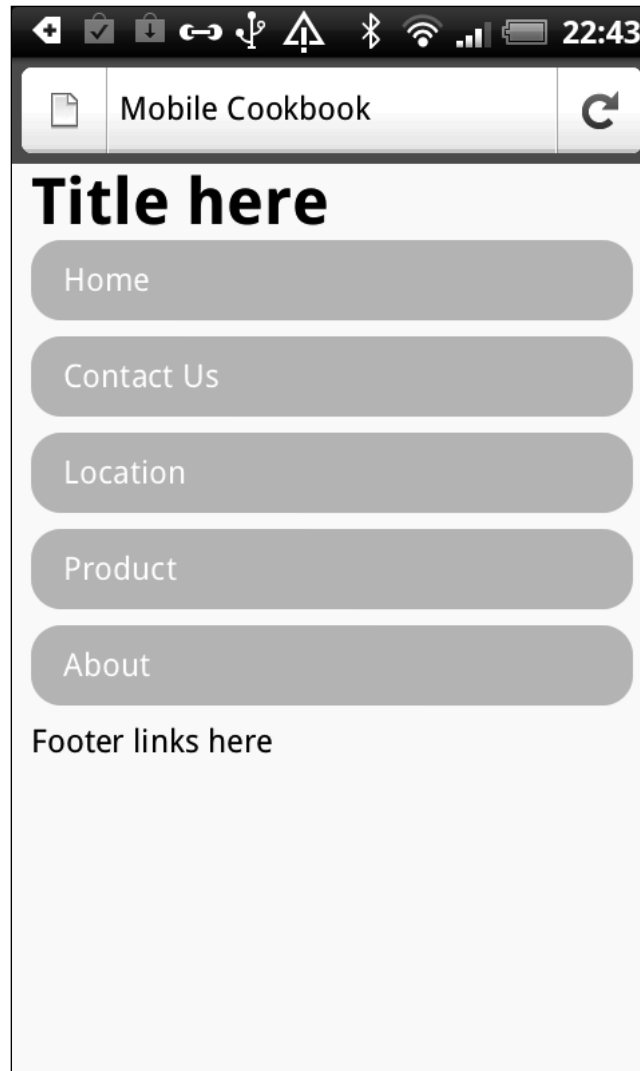
```
<!doctype html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0, maximum-scale=1.0">
    <script src="modernizr-1.7.min.js"></script>
    <style>
      body, #main ul, #main li, h1 {
        margin:0;
        padding:0;
      }
    </style>
  </head>
  <body>
    <div id="container">
      <header>
        <h1>Title here</h1>
      </header>
      <nav id="main">
        <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">Contact Us</a></li>
          <li><a href="#">Location</a></li>
          <li><a href="#">Product</a></li>
          <li><a href="#">About</a></li>
        </ul>
      </nav>
      <footer>
        Footer links here
      </footer>
    </div>
  </body>
</html>
```



```
body {
  background:#FFFA6;
}
#container {
  font-family:Arial;
  margin:0 10px;
}
header, footer {
  display:block;
}
#main li{
  list-style:none;
  height:40px;
  background:#29D9C2;
  margin-bottom:0.5em;
  line-height:40px;
  -moz-border-radius: 15px;
  -webkit-border-radius: 15px;
  border-radius: 15px;
}
#main li a {
  color:white;
  text-decoration:none;
  margin-left:1em;
}
</style>
</head>
<body>
  <div id="container">
    <header>
      <h1>Title here</h1>
    </header>
    <nav id="main">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Contact Us</a></li>
        <li><a href="#">Location</a></li>
        <li><a href="#">Product</a></li>
        <li><a href="#">About</a></li>
      </ul>
    </nav>
    <footer>
      Footer links here
    </footer>
  </div>
</body>
</html>
```

## How it works...

When you view the two sites in portrait mode, they both look pretty much the same:



Now try rotating your screen and see what happens.

As you can see now, in landscape mode, the first example renders with spaces at the left and right side, and the second example covers most of the space of the screen:



The second example shows a different result:



This site page looks very awkward in a fixed layout, but looks normal in a fluid layout. So when you design for mobile, always remember that your site should be designed with this flexibility. The reasons are:

- ▶ Mobile has both portrait and landscape mode
- ▶ Mobile has very limited space, so you should use every pixel available on the screen

## There's more...

CSS media queries is also an essential part of responsive design. It assists you with the flexibility of mobile design.

```
<!doctype html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0, maximum-scale=1.0">
    <script src="modernizr-1.7.min.js"></script>
    <style>
      body {
        margin:0;
        padding:0;
        background:#FFFFA6;
      }

      #main section {
        display:block;
        border:5px solid #29D9C2;
        width:60%;
        height:120px;
        margin:5% auto;
      }

      @media screen and (min-width: 480px) {
        #main {
          width:90%;
          margin:0 auto;
        }

        #main > section:first-child {
          margin-right:5%;
        }

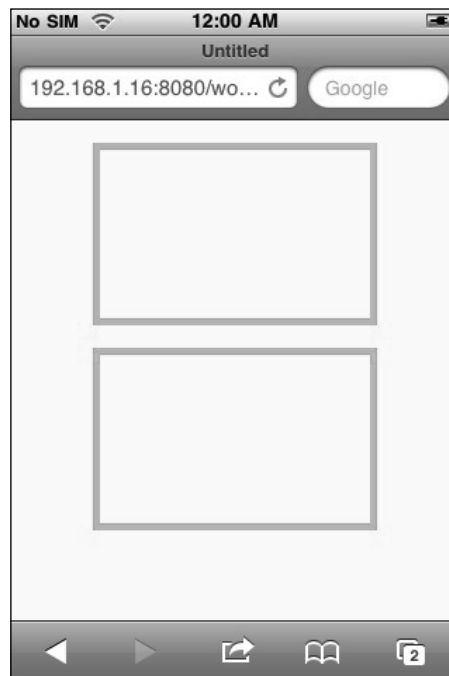
        #main section {
          float:left;
```

```
        width:45%;
    }
}
</style>
</head>
<body>
    <div id="container">

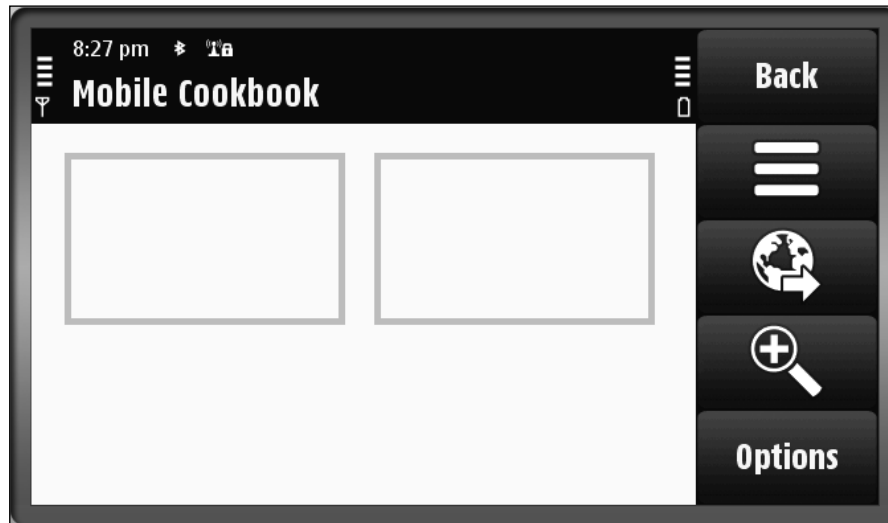
        <div id="main">
            <section id="top-news"></section>
            <section id="sports"></section>
        </div>

    </div>
</body>
</html>
```

When rendered in a narrower screen, the two sections will be laid out vertically, when rendered in a wider screen, the two sections will be laid out horizontally. The technique we used to make this happen is by using CSS media queries. As in this example, we used `@media screen and (min-width: 480px) { .. }`, so what it means is that the page is rendered on a page that has a minimum width of 480px, the styles within will be applied:



Now let's see it in orientation mode, as shown next. The two boxes are now next to each other.



### Desktop-first site

Apart from the idea of building a purely mobile or desktop site, there have also always been other ways. One of those is to build a desktop site first, and make it degrade gracefully in mobile.

### Mobile-first site

Another approach is to build it 'mobile first' but make it render gracefully on the desktop.

One such approach uses the following in the CSS:

```
@media only screen and (min-width: 320px) {  
  /* Styles */  
}  
@media only screen and (min-width: 640px) {  
  /* Styles */  
}  
@media only screen and (min-width: 800px) {  
  /* Styles */  
}  
@media only screen and (min-width: 1024px) {  
  /* Styles */  
}
```

## One web approach

A third approach is to have a 'one web' version, by which, instead of focusing on mobile or desktop, you focus on both.

## Defining a content strategy

Target browser: all

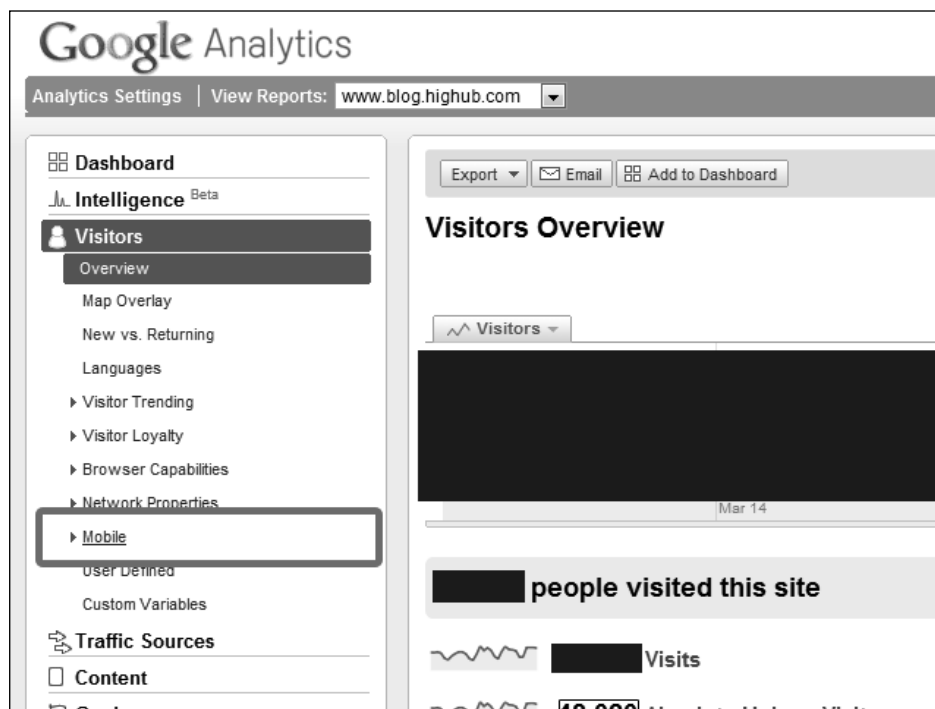
With the data you gathered from the analytics tools, you can define a strategy towards what you want to build. This is particularly useful if you already have a desktop version of the website.

## Getting ready

Make sure you already have JavaScript embedded on your site.

## How to do it...

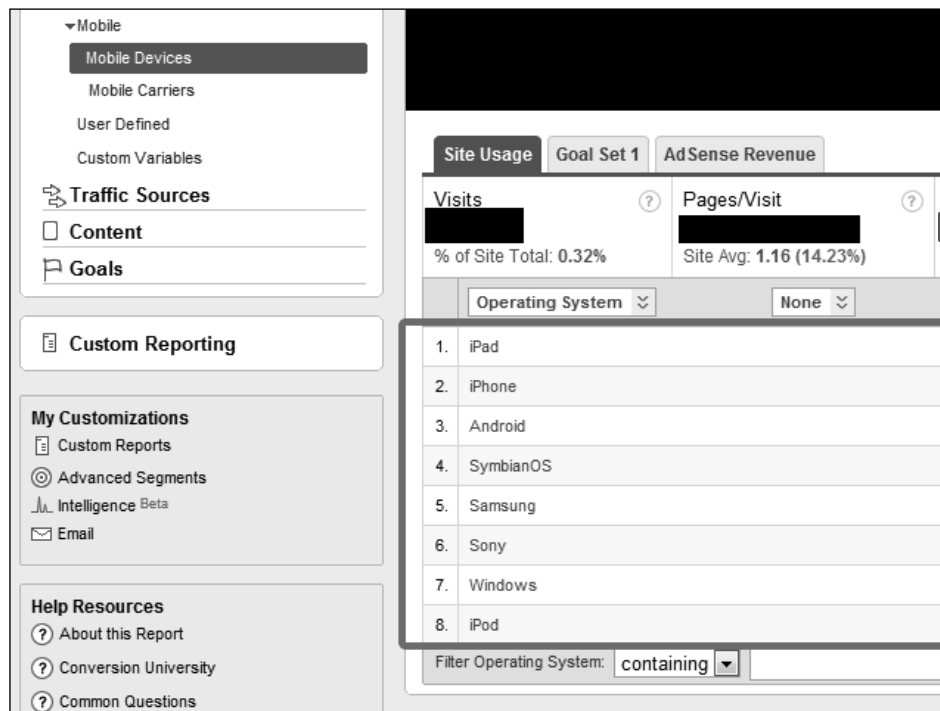
1. Go to your analytics tool and click on **Visitors** | **Mobile** under the left navigation:



For More Information:

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

- Now if you click on **Mobile Devices**, you can see the most common devices that people use to surf your site:



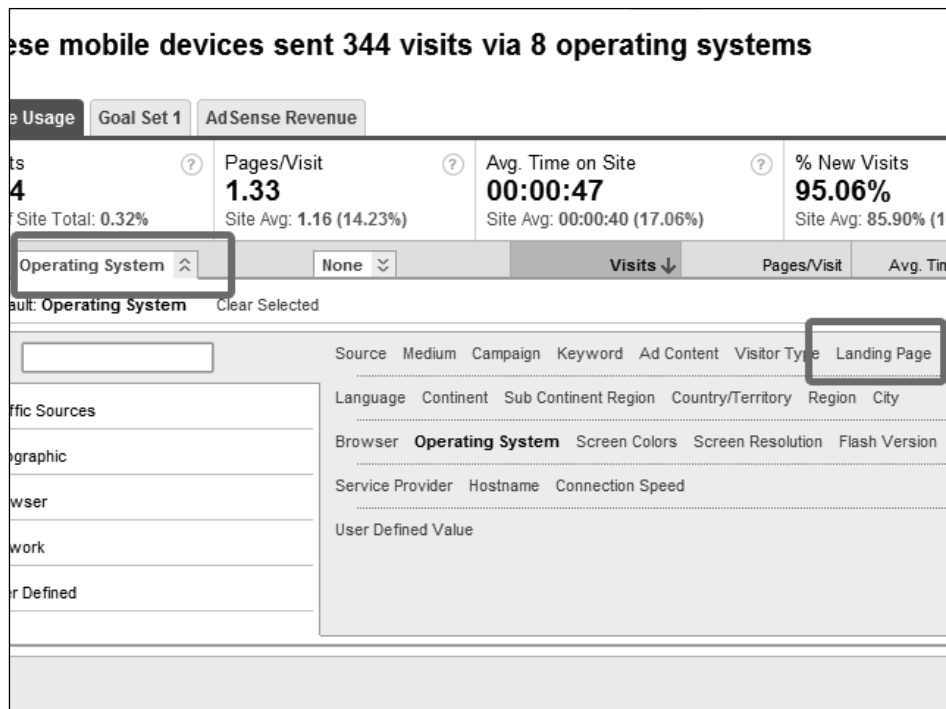
## How it works...

Google Analytics can help in finding out the most commonly used mobile devices accessing your site, and also find out the most popular sections of your website.

## There's more...

You can also determine what are the most useful pages on your mobile site. People treat mobile surfing differently from desktop surfing. For instance, if you are running a local store selling products, most people on a tend to surf pages like **Contact Us**, **Location**, and **Services** mobile device. Conversely, on a desktop, people tend to search for **Product Catalog**, **About**, and **Product Description**. Google Analytics can help you find out which are the most visited sections/pages on your site. Apart from Google Analytics, you can also use PercentMobile, as we saw earlier.





## Browser grade

Using an analytics service is one way to decide which devices you want to support. Another way is to use browser grade to know what category you should be targeting. jQuery Mobile has an awesome grid support chart at <http://jquerymobile.com/gbs/>. There is also a slide on jQuery Mobile that talks about the overall strategies regarding mobile browser web development at <http://www.slideshare.net/jeresig/testing-mobile-javascript>.

## Mobile matrices

I have been collaborating with *Jonathan Neal*, and many others, on Smartphone frontend matrices. You can have a look at:

<https://github.com/h5bp/mobile-boilerplate/wiki/Mobile-Matrices>

It contains information about most Smartphones in the market, their screen dimensions, DPI, and operating systems.

**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)

## Where to buy this book

You can buy HTML5 Mobile Development Cookbook from the Packt Publishing website: <http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book>.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our [shipping policy](#).

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.



**For More Information:**

[www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book](http://www.packtpub.com/html5-mobile-development-for-iphone5-android-blackberry-cookbook/book)