Getting Ready for Responsive Web Design

Over the past four decades, PCs have evolved from being heavy commercial equipment to slick, high-power machines. Mobile phones that once used to be large and bulky have gradually shrunk to lighter weights and more manageable sizes and shapes. Today screen resolution across devices range from 320px (iPhone) to 2560px (large monitor) or even higher. Web designers often face the difficult task of fitting their designs into such a wide variety of devices and screen resolutions. Is there a common solution that can cater to the needs of the diversified user-and-device-ecosystem in the web world?

In the physical world, of course, there are experiments underway for art installations and wall structures that would bend, flex and expand as crowds approach them. Climate control systems along with motion sensors are being used to adjust a room’s temperature and ambient lighting, as it fills with people. But what about the web world? Are we future-ready? Rather than having disconnected designs for each kind of delivery channel, we should devise a single design for an optimal viewing experience, irrespective of the media on which it is delivered.

Responsive Web Design (RWD) could be a solution to this problem. With RWD, we can create customized solutions for a wide range of users, on an equally wide range of devices. This paper addresses how RWD could be a solution for similar challenges, and delves into some methods and techniques for using CSS3 media queries along with HTML5, in order to achieve this objective.
About the Author

Mukti Sendha

Mukti is a Solution Architect with 12 years of experience at Tata Consultancy Services (TCS). He has been involved in architecting solutions on a diverse set of technology platforms in the areas of financial services, insurance, manufacturing and retail. Currently, he leads the User Experience Centre of Excellence (CoE) and is a Certified Usability Analyst (CUA) from Human Factors International.
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation/ Acronym</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCS</td>
<td>Tata Consultancy Services</td>
</tr>
<tr>
<td>CoE</td>
<td>Centre of Excellence</td>
</tr>
<tr>
<td>CUA</td>
<td>Certified Usability Analyst</td>
</tr>
<tr>
<td>AWD</td>
<td>Adaptive Web Design</td>
</tr>
<tr>
<td>RWD</td>
<td>Responsive Web Design</td>
</tr>
</tbody>
</table>
Table of Contents

1. Introduction 5
   1.1. Current challenges 5
   1.2. Possible solutions 5
2. Responsive Web Design 6
3. HTML5 7
4. CSS3 & Media Query 8
   4.1. Sample CSS3 media query code 8
   4.2. Multiple style sheets using media queries 9
   4.3. JavaScript and jQuery support for CSS3 10
   4.4. Media queries and screen resolution 10
   4.5. Browser support for media queries 11
5. Creating RWD Using HTML5 and CSS3 11
   5.1. Creating a responsive website 14
   5.2. Testing responsive websites: 16
6. Conclusion 16
1. Introduction

Statistics from internet usage patterns reveal a very interesting phenomenon. More than 50% of smartphone users throughout the world use the internet through their mobile devices to get relevant information quickly, especially when they are away from their computers. Around 40% of users browse the internet while travelling.

Any business providing services through internet-based channels would not want to miss out on this growing user base and would look forward to avail themselves of all possible delivery channels for providing information and services.

1.1. Current challenges

With over four billion mobile devices in use around the world, mobile browsing is on the rise. We can no longer assume that our sites will be viewed only on a desktop monitor, with an average screen resolution. However, given the innumerable number of devices, players, operating systems, screen dimensions, and contexts to design for, User Experience (UX) designers face the challenging task of figuring out how to deliver the perfect website across desktops, tablets and smartphones.

1.2. Possible solutions

There are two recommended strategies for UX designers – Responsive Web Design (RWD) and Adaptive Web Design (AWD or mobile websites).

Responsive Web Design (RWD) is an approach in which a site is designed to provide an optimal viewing experience across a wide range of devices, from desktop computer monitors to mobile phones. RWD is widely adopted by organisations across the globe; but it is not the only solution for designing websites that need to work across various media.
Adaptive Web Design (AWD), on the other hand, is an approach for device-specific design. In other words, it comprises specific designs suited for specific media such as the web, mobile devices, desktops and so on.

However, no single design approach might be suitable all the time. It depends on the context and the real business situation.

This paper provides information on how to make a web site available for all channels, by employing the RWD technique.

2. Responsive Web Design

Responsive Web Design (RWD) is an approach in which a site is designed to provide an optimal viewing experience across a wide range of devices, from desktop computer monitors to mobile phones. Optimal viewing experience includes easy reading and navigation with minimum resizing, panning or scrolling. In other words, it is an approach that enables design and development to respond to the user's behaviour and environment, based on screen size, platform and orientation.

RWD mainly consists of a mix of flexible grids, layouts and images, and makes intelligent use of CSS media queries to adjust screen resolutions and automatically resize images.

The following picture displays how the same site appears on different devices and screen resolutions:

![Fig 2: Responsive Display (One site for every screen)]
AWD, on the other hand, is an approach for device specific design. In other words, there will be different designs for different screen resolutions and devices.

A comparison between RWD and AWD is as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Responsive Web Design</th>
<th>Adaptive Web Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout</td>
<td>Different layouts for different devices</td>
<td>A dedicated layout for each type of device</td>
</tr>
<tr>
<td>Content</td>
<td>Content stays same for all devices</td>
<td>Different content for different devices</td>
</tr>
<tr>
<td>Speed</td>
<td>Websites with lots of HTML code and multimedia take far longer to load</td>
<td>Loading and processing time is faster as the HTML and CSS code is condensed</td>
</tr>
<tr>
<td>Viewing Experience</td>
<td>No accounting for users’ behavioural differences with different media</td>
<td>Optimum viewing experience on the device used</td>
</tr>
<tr>
<td>Cost and Time of</td>
<td>Less time and cost involved as the solution is generic</td>
<td>More cost and time involved for device-specific development</td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. HTML5

HTML5 is the most recent version of Hyper Text Markup Language (HTML) and is an outcome of the cooperation between the World Wide Web Consortium (W3C) focusing on XHTML 2.0, and the Web Hypertext Application Technology Working Group (WHATWG) focusing on web forms and applications.

With the increase in internet and web usage, more and more users began going online and using multiple devices to access information and content from the internet. Existing languages were unable to provide features that matched user expectations and, in this sense, were deficient. Keeping in mind these trends, in the year 2006, W3C and WHATWG decided to cooperate and create a new version of the HTML.

Some of the new features expected from the new version were as follows:

- Improved new features based on HTML, CSS, DOM and JavaScript
- Device-independent features
- Reduced need for external plugins (like Flash)
- Better error handling
- More markup to replace scripting
- A visible development process for the public

HTML5 covers the features of HTML4, XHTML1 and DOM2HTML. It is a development framework with CSS3 and JavaScript, with a number of new elements, attributes, 2D and 3D graphics, video, audio elements, local storage facility, and local SQL database support.
It supports both desktop deployment and mobile deployment. Smartphones like the Apple iPhone, Google Android, and phones running Palm WebOS, have gained huge popularity with HTML5-based rich web applications.

All major browsers (Safari, Chrome, Firefox, Opera and Internet Explorer) continue to add new HTML5 features to their latest versions.

**Minimum HTML5 document:**

The following is a simple HTML5 document, with the mandatory tags:

```html
<!DOCTYPE html>
<html>
<head>
<title>My First HTML5 Page</title>
</head>

<body>
Hello, The content of the document appears here......
</body>
</html>
```

### 4. CSS3 & Media Query

Along with HTML5, the use of related technologies such as CSS3 media queries, are also becoming increasingly important. Users expect a site to maintain a high level of user-experience satisfaction, no matter how or where they see the content – desktop, laptop, tablet or a handheld device.

CSS 2.1 supports several media types that are also supported by CSS3: screen, print and handheld. CSS3 goes further and includes max-width, device-width, orientation (portrait or landscape) and color features.

The display attributes are controlled in smart devices using CSS3 media queries and sensor technologies. Let us take a detailed look at some samples of CSS3 media query code.

#### 4.1. Sample CSS3 media query code

```html
<link rel="stylesheet" type="text/css"
media="screen and (max-device-width: 480px)"
href="four8zero.css" />
```

This media query triggers the ‘four8zerostyle’ cascading style sheet if the visitor’s browser is 480 pixels or less. In this manner, different style sheets can be created to accommodate different orientations as well as screen sizes.
4.2. Multiple style sheets using media queries

New CSS3 features also include **orientation** (portrait vs. landscape), **device-width, min-device-width** and more.

Multiple style sheets can be created, or basic layout alterations defined to fit a range of widths, including landscape versus portrait orientations. However, the most efficient option is the multiple media query, which can also be used in a single style sheet:

An example of multiple style sheets:

```html
<link rel="stylesheet" media="screen and (max-width: 600px)" href="small.css" />
<link rel="stylesheet" media="screen and (min-width: 600px)" href="large.css" />
<link rel="stylesheet" media="print" href="print.css" />
```

An example of multiple media queries in a single style sheet:

```css
@media screen and (min-width: 600px) {
 .sixhundredminwidth { width: 30%;float: right; }
}

@media screen and (max-width: 600px) {
 .sixhundredmaxwidth { clear: both;font-size: 1.3em;}
}
```

In the above examples, **min-width** and **max-width** are specific to both the screen size as well as the browser size. The **min-width** property sets a minimum browser or screen width that a certain set of styles (or a separate style sheet) would use. If anything is below this limit, the style sheet link or styles will be ignored. The max-width property does just the opposite. Anything greater than the specified maximum browser or screen width would not apply to the respective media query. In contrast, **max-device-width** and **min-device-width** apply specifically to the device, and work best with handheld devices.

For iPads, there is a specific media query property called orientation. The value can be either landscape (horizontal orientation) or portrait (vertical orientation).

```css
/* iPadLandscape Orientation----------- */
@media screen and (orientation: landscape) {
 .iPadLandscape { width: 30%;float: right; }
}

/* iPadPortrait Orientation----------- */
@media screen and (orientation: portrait) {
 .iPadPortrait { clear: both; }
}
```
4.3. JavaScript and jQuery support for CSS3

For browser versions and devices, which do not support CSS3 media queries, there is a JavaScript named css3-mediaqueries.js. This pre-made JavaScript library allows older browsers (IE 5+, Firefox 1+, Safari 2) to support CSS3 media queries. A copy of the file can be downloaded from http://code.google.com/p/css3-mediaqueries-js/. It can then be included in the markup css3-mediaqueries.js.

The following is a sample jQuery snippet that detects browser width and changes the style sheet accordingly:

```html
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
 $(window).bind("resize", resizeWindow);
 function resizeWindow(e)
 {
 var newWindowWidth = $(window).width();
 if(newWindowWidth< 600)
 {
 $("link[rel=stylesheet]").attr({href : "mobile.css"});
 }
 {$("link[rel=stylesheet]").attr({href : "style.css"});}
 }
});
</script>
```

4.4. Media queries and screen resolution

A media query allows the targeting of certain device classes, as well as the inspection of the physical characteristics of the device rendering the work. Let us examine the following piece of code:

```html
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px)" href="lessthan480.css" />
```

The query contains two components:

- A media type (screen), and the actual query enclosed within parentheses, containing a particular media feature (max-device-width) to be inspected.
- The target value (480px).

In this case, we are checking if the device’s horizontal resolution (max-device-width) is equal to or less than 480px. If the test is cleared (that is, if we’re viewing our work on a small-screen device like the iPhone), the device will load the ‘lessthan480.css’. Otherwise, the link is ignored altogether.
4.5. Browser support for media queries

All modern browsers support media queries. Desktop browsers such as Safari 3+, Chrome, Firefox 3.5+, Opera 7+ and more recent mobile browsers such as Opera Mobile and mobile Webkit etc., all parse media queries, but are not supported by older versions of desktop browsers. For browser versions which do not support native parsing of media queries, there is a jQuery plugin that provides limited media query support. It implements only the min-width and max-width media properties, when attached to separate link elements.

Recently a javascript library named css3-mediaqueries.js has been released which also enables browsers like IE 5+, Firefox 1+ and Safari 2 transparently parse, test, and apply CSS3 Media Queries’ when included via @media blocks.

5. Creating RWD using HTML5 and CSS3

5.1. Creating a responsive website

Let us create a responsive website called ‘MyFirst RWD’ using HTML5 and CSS3:

1. Create a Dynamic Web project. This example uses ‘Eclipse HELIOS’ as the web editor.
2. Create the following:
   a. One html file called ‘Default.html’
   b. Two style sheets called ‘media-queries.css’, which will contain all css3 media query attributes and ‘style.css’ for formatting the html file
   c. Some images to be kept in ‘images’ folder.
3. Use the sample code provided below to create the html and css files. You can reuse the code snippets provided below to create your files and customize them.

**Sample code for Default.html:**

```html
<meta name="viewport" content="width=device-width; initial-scale=1.0">
<link href="style.css" rel="stylesheet" type="text/css">
<link href="media-queries.css" rel="stylesheet" type="text/css">

For menu items use class “clearfix”
```ul" id="main-nav" class="clearfix">
<li><a href="">Home</a></li>
</ul>

```form id="searchform">
   <input type="search" id="s" placeholder="Search">
```

For content use class “post clearfix”
```article class="post clearfix">
<header><h1 class="post-title"><a href="#">Polar Bears</a></h1></header>
<figure class="post-image"><img src="images/Polar-Bears.jpg"></figure>
<p>Description goes here</p>
```

For side bar use class “widget”
```aside id="sidebar">
   <section class="widget">
      <h4 class="widgettitle">Other Arctic Animals</h4>
      <ul><li><a href="#">Blue Whale</a></li>
```
```
Sample code for media-queries.css:

```css
@media screen and (max-width: 980px)
{
    #content {width: 60%; padding: 3% 4%;}
    #sidebar {width: 30%;}
    #sidebar .widget {padding: 8% 7%; margin-bottom: 10px;}
}
@media screen and (max-width: 650px)
{
    #header {height: auto;}
    #searchform {position: absolute; top: 5px; right: 0; z-index: 100; height: 40px;}
    #searchform #s {width: 70px;}
    #searchform #s:focus {width: 150px;}
    #main-nav {position: static;}
    #site-description {margin: 0 0 15px; position: static;}
    #content {width: auto; float: none; margin: 20px 0;}
    #sidebar {width: 100%; margin: 0; float: none;}
    #sidebar .widget {padding: 3% 4%; margin: 0 0 10px;}
}
@media screen and (max-width: 480px)
{
    html { -webkit-text-size-adjust: none;}
    #main-nav a {font-size: 90%; padding: 10px 8px; }
}
```

Sample code for style.css:

```css
html, body, div, form, h1, h2, p, ol, li, a, em, i, img, fieldset
{
    margin: 0; padding: 0;
}
img, fieldset { border: 0; }
img { max-width: 100%; height: auto; width: auto; }
article, header, hgroup, menu, nav, section { display: block; }
body {
    background: #0d1424 url(images/body-bg.jpg) no-repeat center top;
    font: .81em/150% Arial, Helvetica, sans-serif; color: #666;
}
```
5.2. Testing responsive websites:

In order to test a responsive website fully, it is not practically possible to set up different systems for every screen size and device. One approach to test RWD sites is by resizing the browser window. Also, there are various third-party plugins and browser extensions that display the current browser window or viewport size in pixels. The IE Developer toolbar, Firesizer, Safari Resize are a few such examples.

Note: Viewport is the area within the browser window, where the content appears. It excludes the
toolbars, tabs etc. and is the actual area where a website appears. Screen size, on the contrary, refers to
the physical display area of a device.

The best solution to test the viewport across browsers is to create a simple html file. The following code

```html
<body>
<script src="http://code.jquery.com/jquery-latest.js"></script>

<script>
$(window).resize(function() {
    var the_width = $(window).width();
    $('#width').text(the_width);
    var the_height = $(window).height();
    $('#height').text(the_height);
});
</script>
</body>
```

The following figure illustrates how an image finally appears in different devices and screen resolutions:

**Laptop**
Viewport Size (Width * Height): 1280 * 697
Screen Resolution (Width * Height): 1280 * 800

**Galaxy Tab 750**

6. Conclusion

Understanding how to make a design responsive to the user is relatively easy to achieve, and is less taxing and more productive than designing code properly for every device available. With RWD, we can create customized solutions for a wider range of users, on a wider range of devices.

The three technical ingredients for RWD are fluid grids, flexible images and media queries. Besides these technical components, RWD also requires a unique way of thinking. Rather than confining content into disparate, device-specific experiences, media queries can be used to progressively enhance work within different viewing contexts.

However, the RWD techniques discussed above are not the final answer to the dynamic mobile world. RWD is a concept which, if implemented correctly, can improve the user experience, but cannot become a universal solution for all users, devices and platforms. We need to constantly work with new devices, resolutions and technologies to continuously improve user experience as technology evolves in the future.

References
Contact
To know more about us, contact ntdg.ux@tcs.com

Subscribe to TCS White Papers
Feedburner: http://feeds2.feedburner.com/tcswhitepapers

About Tata Consultancy Services (TCS)
Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India’s largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at www.tcs.com

IT Services
Business Solutions
Outsourcing