

Links & Tools

Remember to take advantage of the Responsive Design view in Firefox 15 and greater.

Responsive Resources

<http://css-tricks.com>

<http://www.smashingmagazine.com>

<http://twitter.github.com/bootstrap>

<http://pxtoem.com/>

CSS Pocket Reference - 4th Edition
(Eric A. Meyer, O'Reilly)

Responsive Themes for Plone

collective.responsivetheme

beyondskins.responsive

Examples

<http://sixfeetup.com>

<http://bostonglobe.com>

<http://nd.edu>

<http://app.ft.com>

<https://www.fundraise.com>

<http://www.hsgac.senate.gov>

Helpful Articles

Using Viewports

[http://dev.opera.com/articles/view/
an-introduction-to-meta-viewport-and-
viewport](http://dev.opera.com/articles/view/an-introduction-to-meta-viewport-and-viewport)

Designing for Touchscreens

[http://www.whatcreative.co.uk/blog/tips/
designing-for-touch-screen/](http://www.whatcreative.co.uk/blog/tips/designing-for-touch-screen/)



Six Feet Up is a woman-owned business that develops, hosts and supports sophisticated enterprise content management and collaborative intranet projects. Our clients include Harvard, UCLA, Eli Lilly & Oxfam.

Check out our other free Quick Reference cards at www.sixfeetup.com/quickref



This publication is licensed by Six Feet Up, Inc. under the Creative Commons Attribution-ShareAlike 3.0 License. Find out more at www.sixfeetup.com/quickref.

The Plone name and the Plone logo are registered trademarks of the Plone Foundation. All other trademarks and brand names used herein are acknowledged as the property of their respective owners.

Quick Reference Series

GETTING STARTED WITH RESPONSIVE DESIGN

Responsive Design is a term for websites in which the design and content adjust to match the browser width – from desktop browsers down to mobile devices. It eliminates the need of having a separate site for mobile visitors. To make a site responsive, it needs the right combination of HTML and CSS to allow the site to collapse in narrower browsers.



Break Points

A break point is set at a specific browser width when elements in the design need to change or collapse to conform to the narrower view. Instead of targeting a device width, break points should be set at widths where the site's design starts to break. The idea is you should be able to adjust your browser's width, see the collapsing of content, but never see a broken design.

Media Queries

Prior to responsive design, media queries were mainly used to separate stylesheets used for screen and print. But now these queries allow us to target specific browser widths.

The following code gives the site a fixed width if the browser is wider than 960px. Once the browser is narrowed to under 960px, the site becomes fluid. 960px is a break point.

```
#visual-portal-wrapper { width: 960px; }  
  
@media screen and (max-width: 960px) {  
  #visual-portal-wrapper { width: auto; }  
}
```

The query can also be set on the style tag:

```
<style type="text/css" media="screen and  
(max-width: 960px)">@import url(960.css);  
</style>
```

Media queries are supported by all modern browsers. Internet Explorer versions prior to 9 do not support media queries, so any styles put inside of them are ignored. Make use of a conditional style sheet for IE lt 9 to fix any unwanted behavior, or simply make the layout fixed.

Meta Tag for Viewport

In order to have the responsive design work properly in a mobile browser, you will need this tag in your `<head>`:

```
<meta name="viewport" content="width=  
device-width, initial-scale=1" />
```

This will keep the browser from zooming out to force the site to fill the browser width. With this, the browser will scale the site properly, as we expect in a desktop browser.

Fixed vs Fluid

Responsive sites need to have a fluid width in order to properly fill the browser constraints. To avoid the site becoming too wide, you can set a cap by making the site fixed width when the browser is greater than a specific size. *See the example in the Media Queries section.*

Units of Measurement

Pixels - 1px is the size of a single pixel on the viewer's screen.

Percent - Sizes measured in percents will set the width or height relative to the parent container. Percents are important for setting column widths and margins in fluid designs.

Ems - An em is a scalable unit where 1em is equal to the parent element's font size. It is a good idea to use this on fonts, to keep the text legible across various browsers and widths. Most browsers ship with 16px as the default font size. Mathematically $\text{target/context} = \text{result}$. For the font to display at 15px you will use 0.9375em, ($15 / 16 = 0.9375$)

Rem - An em based on the root HTML element, instead of the parent element. This does not work in older versions of IE.

Grid Layout

Many sites using responsive design are built on a grid layout, but it is not a requirement. A grid layout is one that uses preset column and gutter widths. There are many resources for building a grid to fit your site, that you can use for building the site. The nice thing here is that you can find a grid in which the responsiveness is already done for you.

Responsive Images

Just like page elements, images can also be given relative widths so they scale as the page scales. Though not all browsers handle the scaling well, and loading a large image can impact page load time. A possible workaround is to use background images, so a different image can be loaded within the media queries.

Collapsing Divs

With wider browsers, sites generally have room to show three to four columns. As the browser gets narrower, these columns eventually need to collapse down into a single column for mobile viewing. To determine how to order the columns in the template, first determine the order of how the content will display for mobile. Then work your way up. At full site width, columns will be floated and positioned as needed.

Tips

- Keep the touch screen in mind. Make links and buttons large enough to be easily clickable.
- Test on both iPhone & Android, if possible. These are the two most popular mobile browsers, and sometimes you will run into unexpected behavior on one of them.

Menus

As the screen size gets smaller, a menu full of links starts to take up a lot of space. This can be handled by hiding the menu, but keeping it accessible through a slide-out drawer. The menus may also need to be less complex for narrower views, such as not showing sub-navigation, but only top-level.