

Responsive Design

What is it?

Responsive design is a series of techniques that allow you to design consistent sites across screen sizes and formats.

Why should I learn it?

Mobile devices are becoming prevalent, and it is likely that many users will be accessing your site via a smartphone, tablet or other mobile option. You want appropriate design and functionality for your site no matter where your users are engaging with it.

What can I do with it?

You will apply a series of techniques when you design a page that will create fluid and/or custom presentations for all device formats.

How long will it take?

You can start applying Responsive Design immediately to the pages you created in the previous exercises. As you work with it, you will learn tips and techniques that will help you create interesting designs. But you should be able to pick up the basics in a few hours.

Getting Started

You are now developing Web sites to work on many different types of devices - desktops and laptops with different monitor resolutions, tablets like iPads and a variety of mobile phones. This becomes challenging when trying to provide a consistent experience for all your users. Some have developed mobile apps in languages native to the device. These tend to work well on certain devices, but have to be maintained separately from the main website. There are times when this is appropriate, when apps need to use advanced GPS or accelerometer features. But, if you just want to recreate your website experience on a variety of screen sizes, using responsive design is often the best way to go.

Responsive design combines techniques in html and css (and sometimes javascript) to allow you to detect the screen size and make adjustments to the layout. Your content remains the same.

There are four main techniques that you need to employ to make your site responsive.

1. Fluid layout. Use percentages to size your main divs, so that they change with the screen size. You can allow a max-width to prevent your site from displaying in a format that is too large.
2. Fluid images. Use techniques to size images so that they size as the container in which they are enclosed is sized.
3. Media Queries: Use media queries to provided additional, special styles at certain sizes. See the media queries section of the css page.

4. Viewport Meta Element: For iPhone viewing, you have to add this line of code to the `<head>` so that it overrides the default to simply shrink the content to fit the iPhone screen. You will see this in the html.

```
<meta name="viewport" content="initial-scale=1.0, width=device-width" />
```

We'll be working with the example in the css exercise to make it responsive.

You will need to play around with the stylesheet to become familiar with exactly how it works. Take your time to understand how the css works with the html as you add your content and images to your site.

- *Fluid Layout*

The first technique to use in creating a responsive design is to use percentages for the widths of you divs. You can determine if the div should be 100% of the page or something less.

Sometimes you don't want a div to be sized across a very large monitor. For this, you can apply a `max-width` property to limit the size to that dimension.

```
#header { width: 100%; max-width: 1200px; }
```

- *Fluid Images*

To make sure your images are fluid, you must have them contained in a div that has a percentage width associated with it. Then create a style for the `img` tag that includes `max-width: 100%`. This will size the image so that it is never more than 100% of the div that contains it. See the code sample to the right and add those styles to your stylesheet. Then contain any image you want to be responsive in a div with the class "imgbox".

Code Sample - Fluid Images

Add the style below to your css file.

Make sure you save both files. Refresh the `index.html` file in the browser. Refresh the page every time you make a change.

```
img {  
max-width: 100%;  
}
```

```
.imgbox {  
width: 30%;  
}
```

HTML

In your html page, include the image in a div with a class named "imgbox" (you can name this anything you want, it just has to correspond to the class you made above to contain the image). Because this is a class, you can use it any time you need a div that should be 30% of the page. When you open in the browser and size it, you should see the image sizing down.

```
<div class="imgbox">  
<p></p>  
</div>
```

- *Media Queries*

We want to handle as much of the responsive design with percentage widths as possible. But sometimes when you size by percent, there are other elements that need to be changed for certain sized screens or windows. For example, the text size of your navigation items may be too large at the smallest sizes or you may prefer that some divs be stacked instead of side-by-side at a smaller size. For this, we use media queries.

Media Queries are simply special styles or rules to only be implemented at a certain size. You can apply media queries for one or more sizes, and can use either the max-width or min-width properties. See the code to the right for a sample media query.

Code Sample - Media Queries

CSS

The code below applies for sizes where the minimum width is 480px. This is a small size that reflects smartphone layout. You can also include queries for things like orientation (portrait or landscape) and for different media types, like print or tv. But for computers and handheld devices, you can use the "screen" command. Include all styles you want to use at that particular size in that section of the css file.

```
@media screen and (max-width: 550px) {  
#nav a {  
font-size: 80%;  
}  
  
#footer {  
display: none;  
}  
}
```

The code above makes the font-size for the nav links to be 80% of the standard size. In the layout of our file, our divs are already stacked on top of each other, but if you used a layout with side

navigation or a sidebar, you may want to create a stacked layout for those sections at a certain size. You can even delete sections at certain sizes. See the code above that removes the footer.

Try your site at different sizes and decide what media queries you need to create to get your desired layout and design.

- Viewport

The last thing you need to do to make your site fully responsive is to include the Viewport code in the head of your html pages. The viewport code is necessary so that iPhones and other devices will recognize your media queries and responsive techniques, rather than simply shrinking down the original version of your site.

Just include this code in the head of all your html pages. That's it!

```
<meta name="viewport" content="initial-scale=1.0, width=device-width" />
```