Assignment #5:   
Necessary Mobile Version Changes

CIS 373

Isaac Martinez

February 22nd, 2020

# Introduction

When considering what changes need to be made to a website to support mobile browsers, in the past (circa 2000), the predominant answer was to create an entirely separate website with separate HTML, images, JavaScript, and CSS stylesheets. This separate site would be purely focused on mobile browsers. These mobile-focused sites typically had an “m.” DNS hostname prefix. As an example, if your desktop site name was strayer.edu or www.strayer.edu, the mobile site would be m.strayer.edu.

I have personally worked on e-commerce websites that followed this paradigm in the past, and the lesson learned was maintaining two separate sites to look and behave the same way is difficult. Following this paradigm, your development and support costs are doubled, if not more.

The preferred approach today is to develop your website in a mobile-friendly manner. With this approach, you develop one site, with one set of assets, and make it responsive. I will explain some techniques that are commonly used to accomplish this. These techniques include using relative sizes, grids, and media rules.

# Mobile-Friendly Design Techniques

## Relative vs. Absolute Sizes

Whether it’s in HTML or CSS, when specifying element dimensions, you should use relative sizes vs. absolute sizes. Some examples of absolute sizes are points (pt), pixels (px), and picas (pc). Some examples are percentage (%), device font-size relative (em), screen-width relative (vw), screen-height relative (vh). To further illustrate, a 50vw size would take up half of the screen horizontally (x-axis). An 80vh would take up four-fifths of the screen vertically (y-axis). 1.5em font size would be one and a half the font-size of the device font size; keep in mind that the current device font system is a system-defined user preference.

Using relative sizes ensures that the elements on your screen will scale with the screen dimensions of various devices.

## Tables vs. Grids

Unless you need to present data in an Excel-like table fashion, you should use a grid system to layout the elements on the page. The defacto grid system today is FlexBox and supported by every commonly used browser. What differentiates a grid system vs. a table is that grid systems will wrap elements when the elements don’t fit together.

The cells (<td>) within an HTML table don’t wrap. If your user is on a mobile device, chances are they’ll have scroll horizontally to view all of the data. Horizontally scrolling is not ideal when considering the end-user experience.

Grid systems vary, but FlexBox allows for horizontal as well as vertical wrapping. It also allows you to defined various layout properties, left, center, right, justified.

## CSS Media Rules

Despite the aforementioned techniques, you will likely run into a scenario where something doesn’t look correct or “breaks.” To resolve these issues, you develop CSS media rules at predetermined “breakpoints” to handle these scenarios.

These medial rules support referencing different CSS files entirely based on a number of criteria. As an example, when the below elements are added to the <head> element of an HTML document, the browser will load different CSS files. The first two examples are based on the screen width of the browser, the third is used if the document is being printed.

<link rel="stylesheet" media="screen and (min-width: 800px)" href="desktop.css">  
<link rel="stylesheet" media="screen and (max-width: 650px)" href="mobile.css">  
<link rel="stylesheet" media="print" href="printer-friendly.css">

As another example, if an image needs to be made smaller on a mobile device, you could a technique similar to this in your CSS:

@media screen and (max-width: 650px) {  
  img {width: 50%;}  
}

# Closing

In closing, there are many techniques that can be used to make your site mobile-friendly. If you want your site to have the widest audience possible, making it mobile friendly is going to be required. I have given explanations on my three most commonly used techniques, which are relative sizes, grids, and media rules.