

## Lab 4a. Styling for Mobile: Viewport Sizing Units

### Learning Objectives:

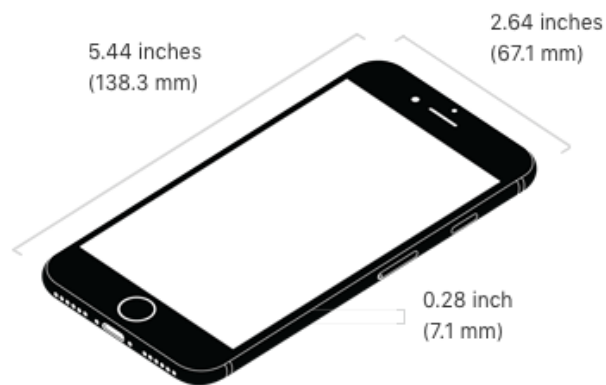
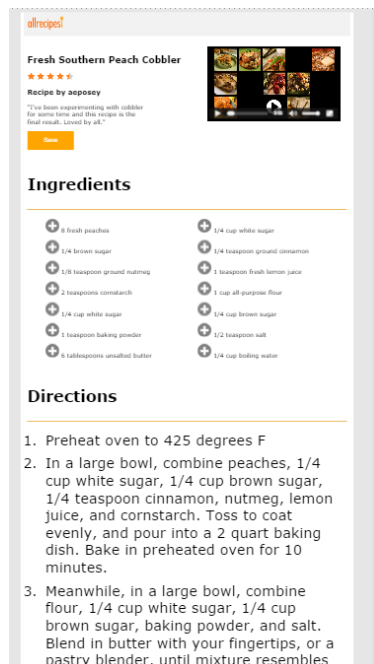
- Understand viewport sizing and how to use it to create responsive websites.
- Understand the important of responsive design.
- Further practice CSS styling attributes.

### Introduction:

By now, we have created a beautiful website that teaches our users how to make a delicious peach cobbler. But, what if our website's visitor is not looking at our site from a desktop computer? Because some of our HTML elements are still "statically" sized-- that is their sizes will remain the same regardless of device size-- our webpage is likely to be unpleasant to read on such a small screen!

For reference, an iPhone7 screen's has a height of just 5 inches and a width of 2 inches with a resolution of 750 x 1334! Many desktop monitors these days are 27 inches across with a resolution of 1920 x 1080. Without scaling your site content for each device, properly viewing your website would be nearly impossible and likely frustrating!

Without proper scaling, your website may look something like (left hand photo). To read any of the text you would have to zoom in many times and some of the text sizing doesn't make sense (i.e the list items are larger than the h1 tags).

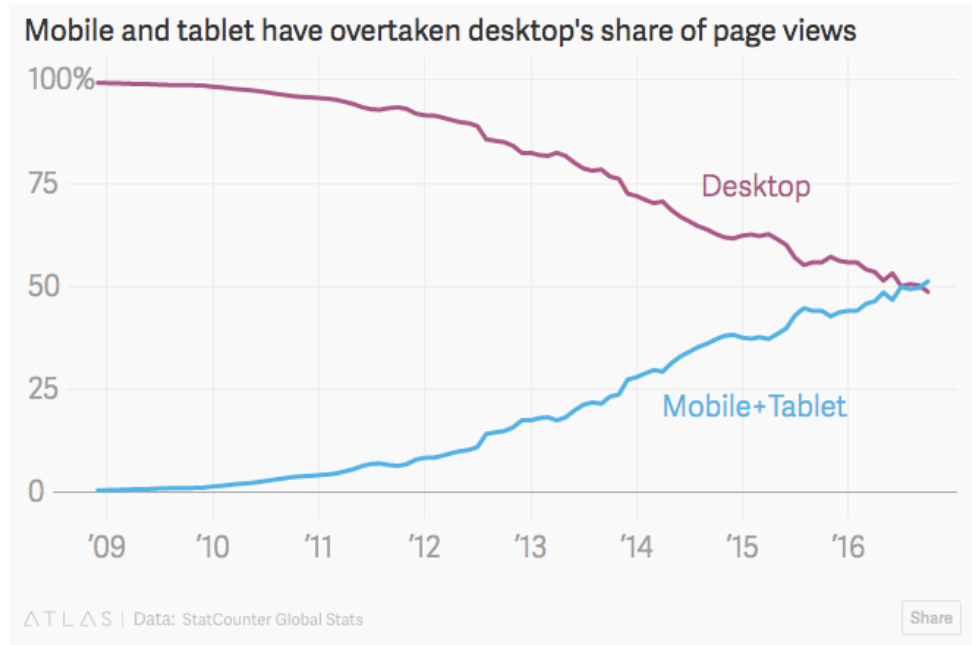


Weight: 4.87 ounces (138 grams)

Above: iPhone7 dimensions.

Not only will your website be hard to read, but your "user" will likely navigate away after experiencing frustration, resulting in potential loss of revenue or advertising sales!

Designing our website so that it is viewable on a variety of digital devices is of **great importance**. In 2016, the number of individual web page views on mobile devices exceeded those of individual web page views on desktop computers! Because of this, the web development community has coined the term: “mobile first design.”



## Part 1: Viewport Sizing

So far we’ve worked with pixels and percentages. There are **many** different ways to size content on a web page, however the most “dynamic” sizing method made available in CSS3 is “viewport sizing.”

Viewport units are extremely powerful in responsive web design because they allow you to size your content according to the size of the current browser window. Viewport units are numerical values, where the number corresponds to (1% \* the number) of the width/height of the window your website is rendering on. **There will be some examples further down in this document.** 😊

### Before you continue:

1. Read the w3schools [CSS Units documentation](#). Play around with the viewport examples.
2. Read the w3schools page [RWD Design](#) and [RWD Viewport](#).
3. Watch the videos in the below folder: (there are many but they are short and informative!)
  - a. \Full Stack Web Development - Quarter 1\Videos\Udacity Videos\Responsive

vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension

In general, we will mostly be using viewport units to dynamically size our text! However, viewport units can also be used to resize images + heights of our content.

## Let's see the math...

Let's convert some font-sizes from pixels to viewport unit sizing:

Example 1: Converting font-size.

### Values:

Viewport width (based on most desktops): 1440px

Current font size (default font size): 16px

### Conversion:

Step 1:  $(1440\text{px} * 0.01) * 1\text{px};$  = 14.4px

Step 2:  $(16\text{px} / 14.4\text{px}) * 1\text{vw};$  = **1.11vw**

**Therefore, 16px on a desktop is equivalent to ~1vw.**

Example 2: Converting image sizing.

### Values:

Viewport width: 1440px;

Viewport height: 800px;

Image width: 150px;

Image height: 250px;

### Conversion:

#### Width:

Step 1:  $(1440\text{px} * .01) * 1\text{px}$  = 14.4px;

Step 2:  $(150\text{px} / 14.4\text{px}) * 1\text{vw}$  = **10.42vw**

#### Height:

Step 1:  $(800\text{px} * .01) * 1\text{px}$  = 8px;

Step 2:  $(250\text{px} / 8\text{px}) * 1\text{vh}$  = **31.25vh;**

**\*Note:** *Because viewport sizing is a (fairly) newly released feature, we will be using “fallback” sizing each time we use viewport units. The term “fallback” is used to explain to the browser what should happen if that specific styling is not available or supported on your platform. Fallback styles are important because they make sure that people using “legacy systems” or older versions of a platform can still access our content!*

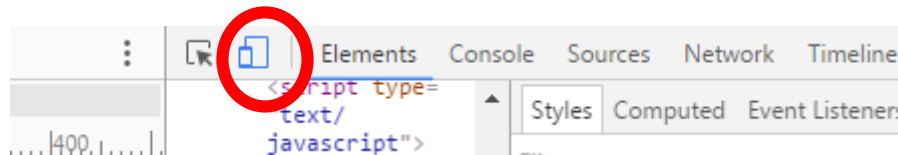
*Fallbacks will look something like:*

```
h1{  
    font-size: 32px; /* Fallback value in case the browser does not support vw */  
    font-size: 2.2vw; /* Viewport sizing unit */  
}
```

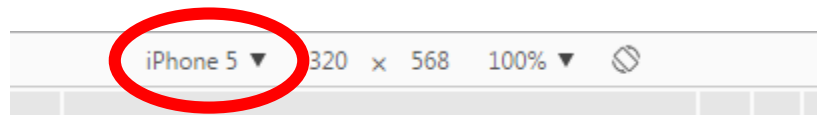
**^^^Because the CSS rule `font-size: 2.5vw;` is applied last, if viewport sizing is supported on the platform, that style will override `font-size: 36px.`**

First, let’s take a look at what our current recipes web page looks like on different devices. To do this:

1. Open up Google Developer Tools.
2. Click the “Toggle Device toolbar” button on the upper lefthand corner. It will look something like (red circle):



3. Now, you can select a variety of devices to “emulate” how your site will look on them. Notice, some of your layouts and font-sizing may be hard to read.



4. Select “iPhone 6” using the gray down-facing arrow.
5. Now, we will add viewport sizing to our content so that it scales better regardless of what device we’re viewing our site from. :)

## Change the following:

1. First, we must set our viewport scaling. In your `<head>` tag in your html file, add the line:  
`<meta name="viewport" content="width=device-width, initial-scale=1">`

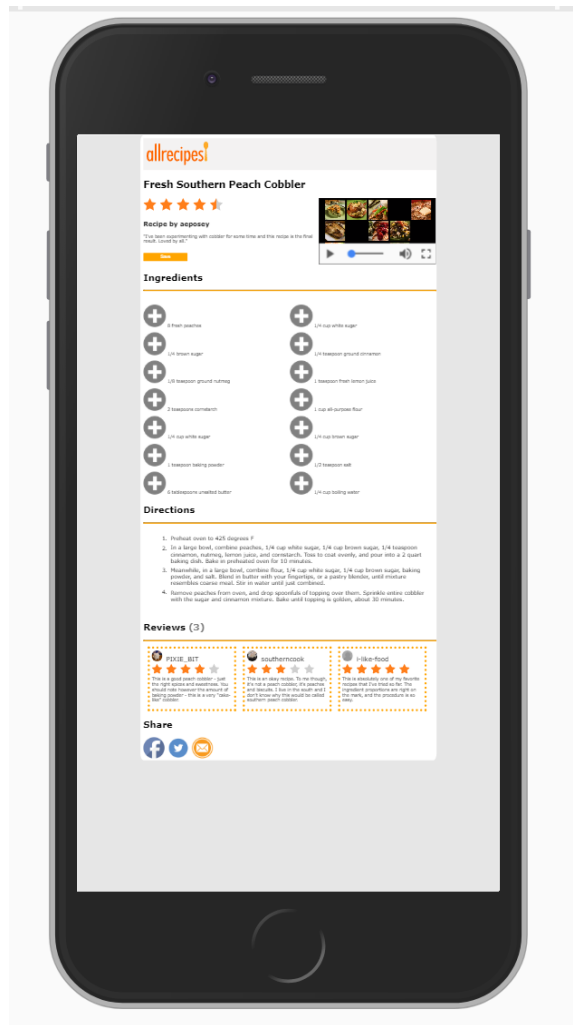
**Note:** Remember to list these after your CSS properties that use pixels. These pixel values are your fallback!

2. All paragraphs should have a font-size: 1vw;
  - a. (Fallback is 16px.)
3. h1 should have a font-size: 2.2vw;
  - a. (Fallback is 32px)
4. h2 should have a font-size: 1.66vw;
  - a. (Fallback is 24px.)
5. h3 should have a font-size: 1.3vw;
  - a. (Fallback is ~18px.)
6. Items in the un-ordered list should have a font-size: 1.1vw;
  - a. (Fallback is 16px.)
7. Let's resize the bullet images on the unordered list items. To do this properly is a bit tricky. Your ul:li:before section should contain the following CSS rules:

```
ul li:before{
    content: "";
    background-image: url("../images/plus-icon.png");
    background-size: contain;
    background-repeat: no-repeat;
    /* Calculate Fallback dimensions here */
    height: 2vh;
    width: 2vh;
    display: inline-block;
}
```

6. Items in the ordered list should have a font-size: 1.1vw;
  - a. (Fallback is 16px.)
7. Save button should have font-size: 1.1vw;
  - a. (Fallback is 16px.)
8. Usernames in the review section should have font-size: 1.66vw;
  - a. (Fallback is 24px.)

After adjusting your sizes to scale using these viewport units, your site should look \*something like\* this on an iPhone 6:



## Part 2: Media Queries

### <Instructions Pending>

After adjusting your website to scale specifically for mobile devices, your website should look something like:

