Advance Styling with Responsive Design

# 4th Course in the Web Design for Everybody (Basics of Web Development and Coding) Specialization Sep

# About Course

This course will expand upon the basic knowledge of CSS3 to include topics such as wireframes, fluid design, media queries, and the use of existing styling paradigms such as Bootstrap. After the course, learners will be able to:

* Explain the mobile-first paradigm and the importance of wireframes in the design phase
* Create sites that behave across a range of platforms
* Utilize existing design frameworks such as Bootstrap

# Skills You Will Gain

**JAVASCRIPT**

**CASCADING STYLE SHEETS (CSS)**

**RESPONSIVE WEB DESIGN**

**BOOTSTRAP (FRONT-END FRAMEWORK)**

# Course Materials

General Links

Here is a link to a collection of code for each week of the course:

Week One: <http://codepen.io/collection/nmyRWM/>

Week Two: <http://codepen.io/collection/nYwVkZ/>

Week Three: <http://codepen.io/collection/DZQyBr/>

Week Four: <http://codepen.io/collection/XkkwBb/>

*Additional Reading*

# Fluid Measurements: Background readings

1. Responsive Web Design: Using Fonts Responsibly by Annarita Tranfici <http://www.sitepoint.com/understanding-responsive-web-design-how-to-manage-fonts/>
2. Web Design Basics: Rem vs. Em vs. PX - Sizing Elements in CSS by Matthew Davis (\*\* Material here is used in Week Two Assessment\*\*) <https://www.futurehosting.com/blog/web-design-basics-rem-vs-em-vs-px-sizing-elements-in-css/>
3. What's the Deal with Em and Rem? (\*\* Material here is used in Week Two Assessment\*\*)<https://codemyviews.com/blog/whats-the-deal-with-em-and-rem>
4. Font Size Idea: px at the Root, rem for Components, em for Text Elements by Chris Coyier <https://css-tricks.com/rems-ems/> (Please note, some people on Firefox get a warning when using this link. It is not the case for other students.)

# OPTIONAL: Good examples of responsive design

The following is a link to good examples of responsive design: <http://www.awwwards.com/50-examples-of-responsive-web-design.html>

this article did a nice job of showing Bootstrap classes along with the CSS that goes with them. <http://www.sitepoint.com/responsive-web-design-tips-bootstrap-css/>

*Beyond Scope*

[JavaScript: Best Practice](https://www.sitepoint.com/premium/books/javascript-best-practice/read)

<https://startbootstrap.com/> free and fee templates

<https://mdbootstrap.com/products/jquery-ui-kit/> free and fee based professional tools

# Week

Week 1

# Syllabus

## Week One: Style with Responsive Design

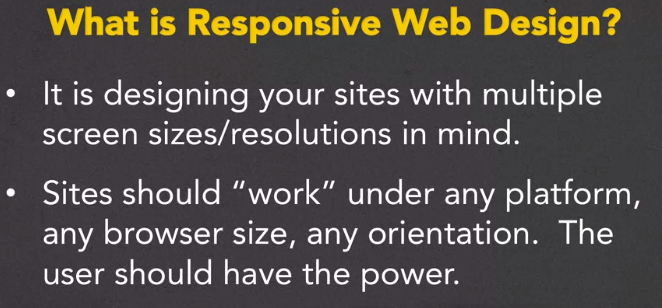
What does it mean to have responsive design for your site? How can you tell if your existing site is responsive? This week we will begin with the theories behind the "mobile-first paradigm" - the idea that your mobile site should provide everything needed, not a pared-down version of a good page. We end the week by taking the first concrete step of using fluid measurements in your CSS.

There are three common approaches to responsive design.

* The first is to create your own site using fluid measurements and media queries.
* The second is to use an existing framework (such as Bootstrap) that does all of the responsive design for you.
* The third is to use a hybrid approach - use a framework in combination with your own responsive code.

Notes

# **What is Responsive Web Design?**



#### Concepts to consider:

Media queries – detecting the viewpoint size ( meta data about the device hosting the browse )

Flexible grid-based layout for relative sizing

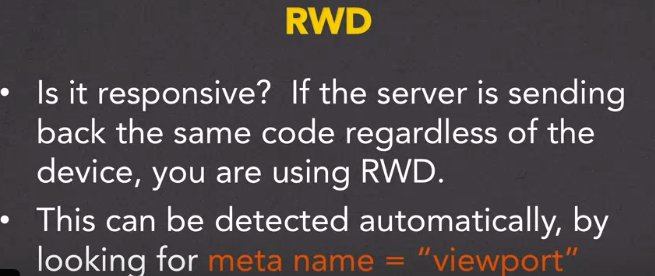
Flexible image

Use inspect element to access the browse’s built in device simulator feature.

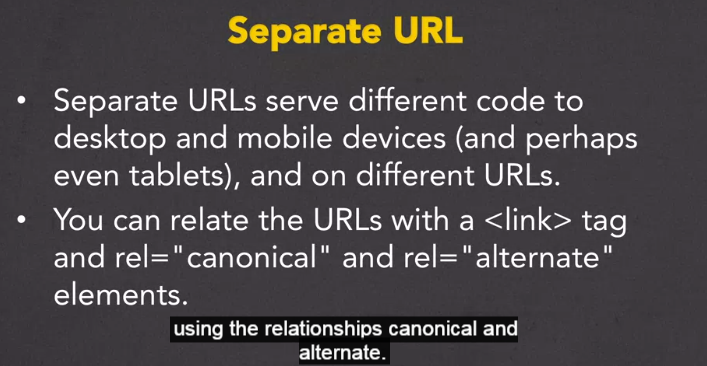
### Benefits of Responsive Design:

Responsive options

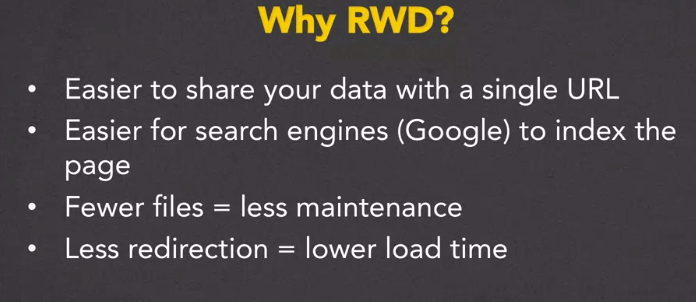
* Responsive Web Design (RWD) – fluid measurements, flexible grids, and varying CSS rules
  + This can be detected automatically, by looking for meta name = “viewport”
  + RWD assist with SEO
    - Easier for search engines (google) to index the page, fewer files and less redirections(performance
  + Accessibility
  + Easier for search engines (google) to index the page, fewer files and less redirections(performance)
* Adaptive Design ( dynamic serving ) – returns one of multiple versions of a page based on the type of device
  + Server detects device type and selects the concurrent html and css
* Separate Mobile Site (.m) – a separate page URL for the mobile site





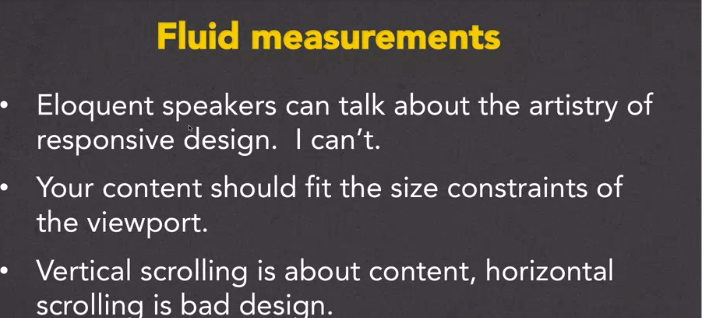


### **So why Responsive Web Design**



With Adaptive Design the server could return the wrong source code if the wrong device type is detected.

#### Fluid Measurements:



Typography is one of the most important aspects of responsive web design, and optimizing your fonts for mobile devices is an absolute necessity if you want your content to be palatable across all screen sizes. We may use different metrics for this purpose, including pixels, ems, rems, or percentages. Choosing the right metric is crucial.

{ font-size:100%; }

This effectively overrides any browser defaults or other unwanted percentage declarations, and leaves the font sizing up to other methods and metrics. In this case, the content could be sized using ems.

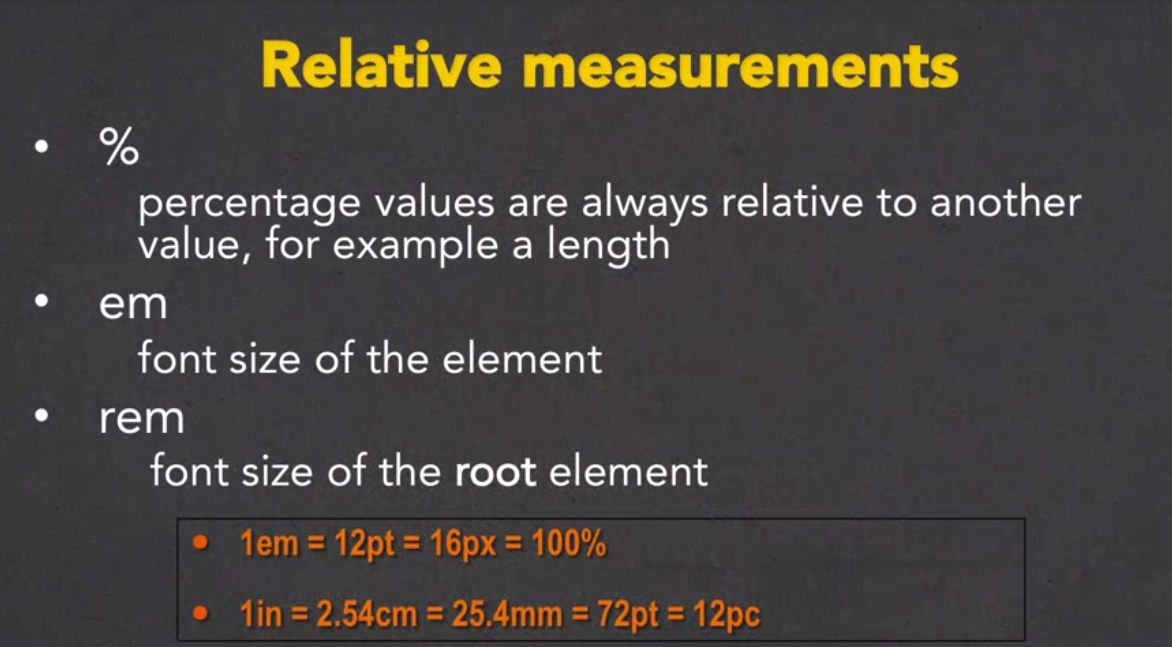
Continue…

#### Absolute measurements

* px – one device pixel (dot) of the display.(1px is equal to 1/96th of 1in)
* mm, cm, in (helpful when you need to conform to printable measurements)
* pt – one point (1/72 of 1in)
* pc – one pica(12 points –pt)

#### Relative measurements

* % values – these values are always relative to another value or elements value
* Em – font size of the element(e.g., if font of a tag set to .5em and font of a nested tag is also .5em then the first tag font size will be half of root font and the nested tag font size will be half again, half of the root tag font size )
* Rem – font size of the root element
* Vw – viewport’s width, 1/100th of the width of the viewport
* Vh – viewport’s height, 1/100th of the height of the viewport



Links

[CodePen examples for week 1](http://codepen.io/collection/nmyRWM/)

[Visit media queries for examples of well implement responsive design](http://mediaqueri.es)

[Visit ami responsive design website to test if your site implements responsive design](http://ami.responsivedesign.is/)

[This CodePen by Chris Coyier will let you use sliders to dynamically change the font size on a page](http://codepen.io/chriscoyier/pen/tvheK)

This demo goes with one of the readings on Fluid Measurements, but I am including it here again just in case you missed the code. You can read it here if you want to know how to use the tool. <https://css-tricks.com/rems-ems/>

Closing Thoughts

Week 2

## Week Two: Basic Concepts

This week you will get a chance to put the theories into practice using media queries in your CSS. These queries can automatically detect the size of the browser being used to view the page so that you can decide what type of look you want to achieve. There will also be a discussion on common practices for designing your different views at various screen sizes ("viewports").

Notes

## **Media Queries**

Media queries allow the style to depend upon the media properties

E.G., <link rel=”stylesheet” href=”style.css” media=”screen”/>

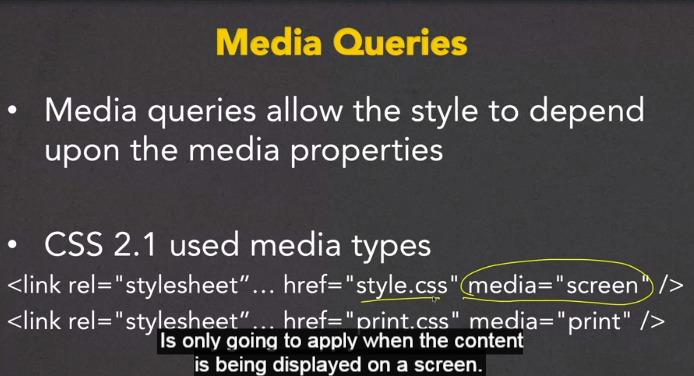
<link rel=”stylesheet” href=”style.css” media=”print”/>

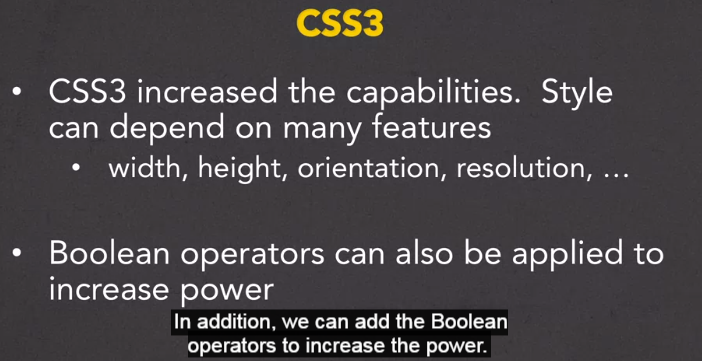
a specific stylesheet would apply to a unique characteristic denoted by the media attribute( width, height, orientation, resolution etc)

#### Boolean operators can also be applied to increase control

Every Media has two components: media type and query

1. A media type ( sceen, print, all, etc )
2. The actual query of a media feature and “trigger” size





### Wire Frames:

Provide a visual representation of you layout(outline – similar to pseudo coding )

Mobile view is the most import step in web design ( good practice to start designing for

mobile platform and scaling up thereafter

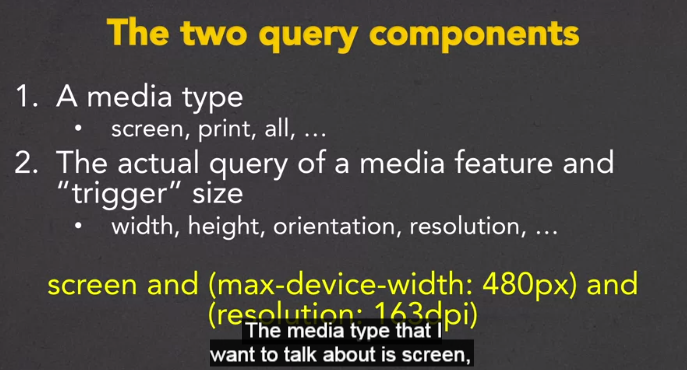
#### Breakpoints:

What are breakpoints – sizes that define a change in your site layout or content.( based of meta data about device )

Determining the Breakpoints

* Breakpoints should correspond to:
  + Devices and/or
  + Content

Design for the median or smallest possible screen size(mobile first) and scale up using the median or largest possible screen size

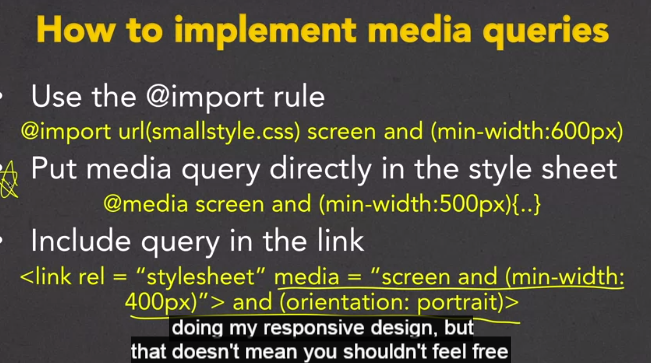


This media query conditions to initi.

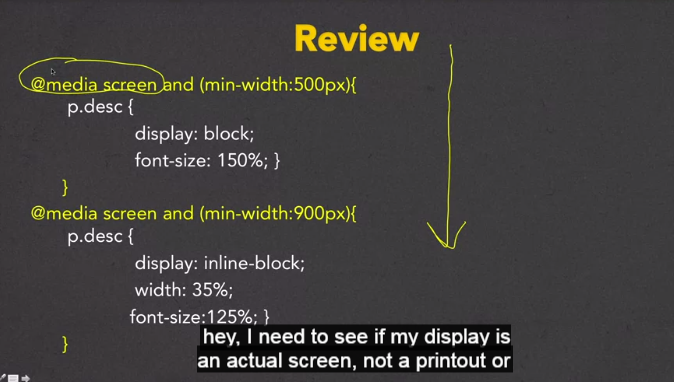
* Screen must be present (default web page viewing state)
* Device width must not exceed 480px
* Device resolution must be 163dpi

### How to implement media queries:

* Use the @import rule
  + @import url ( smallstyle.css ) screen and ( min –width:600px )
* Put media query directly in the style sheet
  + @media screen and (min-width:500px) {…}
* Include query in the link
  + <link rel =“stylesheet” media=”screen and (min-width:400px)”> and (orientation:portrait)>



Some examples using @media applied in the css stylesheet



## Media Queries Part Two:

Step by step

1. Grab information
   1. The media viewport tag tells mobile browser’s viewport how to behave

<meta name =’viewport’ content=’width=device=width, initial-scale=1’>

1. Use fluid layout practices
2. Implement media queries
   1. Fluid layout that is triggered by certain sizes(rules)
   2. Design for mobile first and scale up

## **Wire Frames**

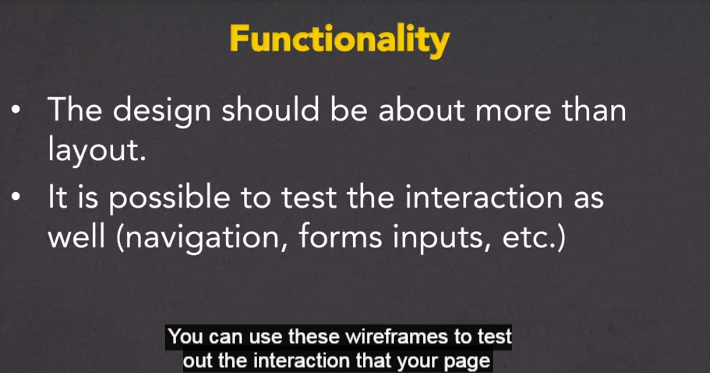
Tablets/Medium Screens

Mobile First

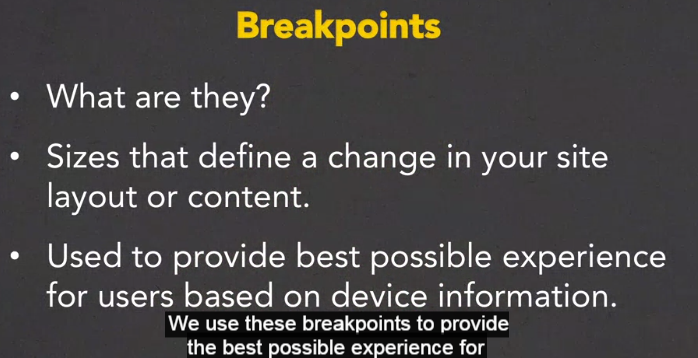
Desktop

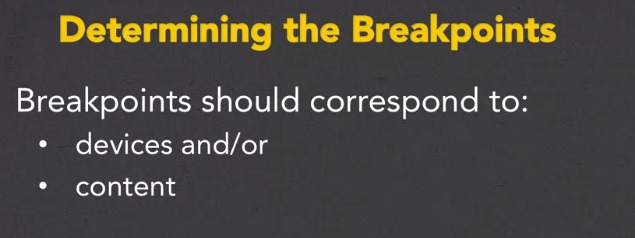




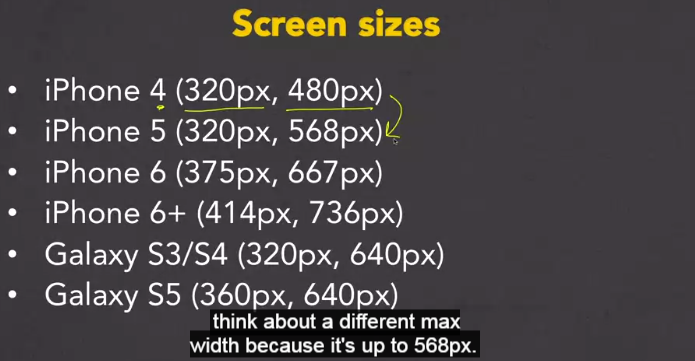


## **Breakpoints**



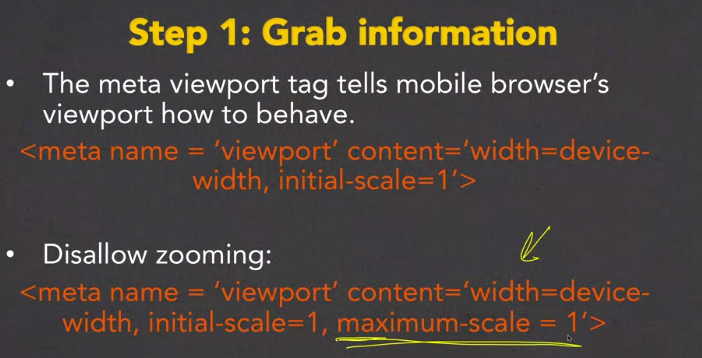


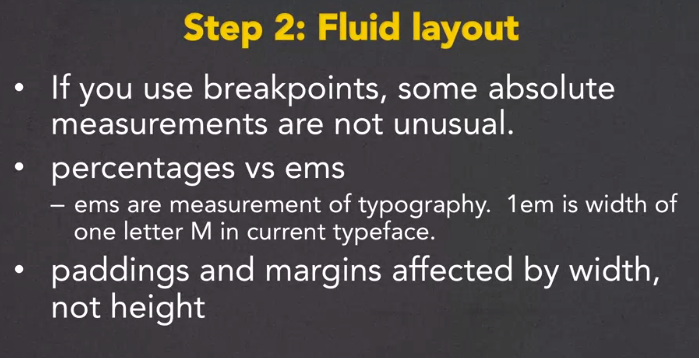
Examples of possible breakpoints to consider

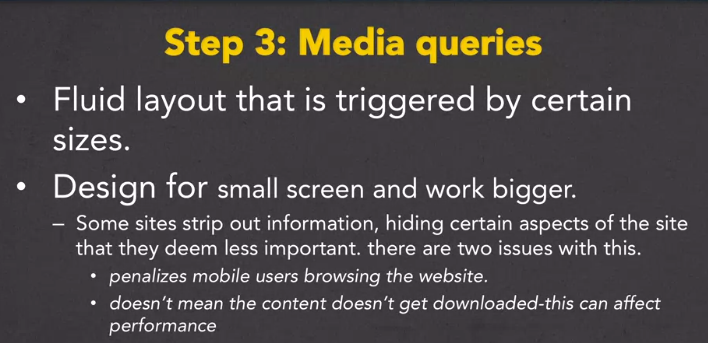




## **Media Queries Part Two**







Links

[Learning to wireframe 10 best practices](https://www.dtelepathy.com/blog/design/learning-to-wireframe-10-best-practices)

Great examples of wire framing

Closing Thoughts

Week 3

## Week Three: Use Existing Frameworks

After every good programmer has put in time creating sites from scratch, it is common to utilize existing tools out there. After all, why recreate the wheel? The work you have done up to this point will give you the knowledge needed to craft your own unique sites from these frameworks. This week we will work with Bootstrap, a framework that uses HTML5, CSS, and JavaScript (but don't worry if you have never used JavaScript yourself).

Notes

## Frameworks:

What does “framework” mean(context)?

* Front-end development
  + Tools to make planning Css, javascript, jquery easier

Popular front-end frameworks

* Bootstrap
* Foundation by ZURB
* Semantic UI
* Pure by Yahoo!
* UIkit by YOOtheme

## Introduction to Twitter Bootstrap 3:

Bootstrap focuses on a responsive mobile first approach.

### Bootstrap breakpoints:

Bootstrap has hard coded breakpoints for different viewports

Extra small Devices and Phones

* Referred to as xs-
* Any device that has a minimum width of 480px.
* Most phones fall within this category

Small Devices and Tablets

* Referred to as sm-
* Any device that has a minimum width of 768px

Medium Devices and Desktops

* Referred to as md-
* Any device that has a minimum width of 992px

Large Devices and Wide Screens

* Lg-
* 1200px

Getting started with bootstrap:

Getting started

* How do you actually use Bootstrap?
* You need access to the CSS code and JS code
* Options:
  + Download copy
  + Use absolute reference
  + Link on CodePen

Using a CDN ( to implement bootstrap )

Content Delivery Network ( CDN ) provides a way to connect to the Bootstrap code using an absolute reference

## Bootstrap Grid System:

Bootstrap layout is based on a 12 column grid system

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Grid classes

* Every grid consists of:
  + A container
  + A row- one or more column classes

<div class=”container”>

<div class=”row”>

<div class=”col-xx-yy”> where xx is viewport size and yy is number of columns to occupy ( class = “col-sm-9” where sm = small screens and 9 = 9/12 columns occupied)

Combining elements

What if you want two elements next to each other, but only on larger screen(block elements on xs screens and inline block elements on md screens )?

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

<div class=”col-xs-12 col-md-3”>Yellow Part</div>

<div class=”col-xs-12 col-md-8”>Orange Part</div>

* Because Bootstrap is mobile first, there is no need to specify the twelve columns if the element is block

### Positing classes

On viewports md and up, there is an option for push and pull class

* + –col-XX-push-YY move YY columns to the right
  + –col-XX-pull-YY move YY columns to the left

### Responsive utility classes

* Hidden-XX content will only be hidden on the XX screen size
* Visible-XX content will only be visible on the XX screen size
* Sr-only content is hidden on all devices except screen readers

Note: hidden class has been depreciated in current version of bootstrap. Replaced with the d-XX classes

## Bootstrap Navigation:

Making a navigation bar:

Decide what type of links you want: - nav-tabs or nav-pills

* RD:nav-tabs
* RD:nav-pills

Decide on layout ( horizontal, stack, vertical, justified, etc )

Navbar class

The navbar class serves as a navigation header for your application or site

* Positioning includes: - navbar-static-top and navbar-fixed-top or bottom

Links

[Examples of websites built with bootstrap](https://builtwithbootstrap.com/)

[Getting started with bootstrap](https://getbootstrap.com/docs/4.3/getting-started)

Closing Thoughts

Necessary for Bootstrap’s JavaScript plugins

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<!-- Include all compiled plugins (below), or include individual files as needed -->

Week 4

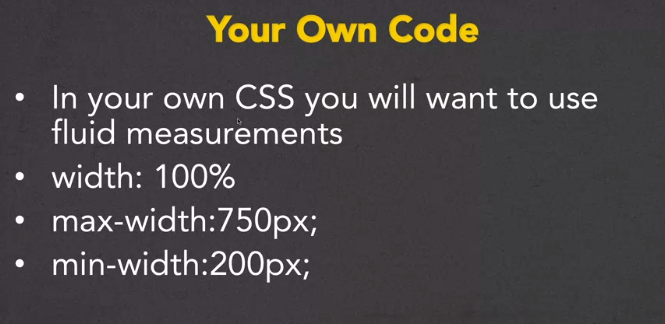
## Week Four: Experiment!

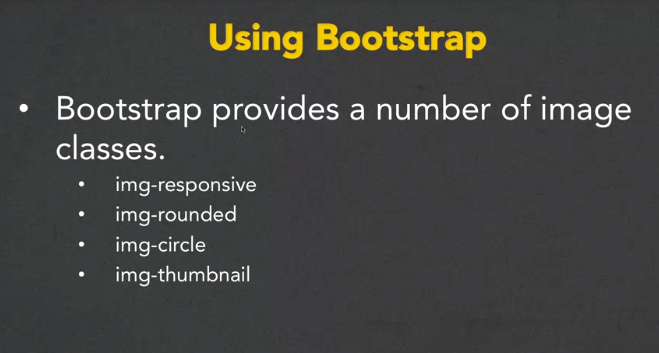
This week is dedicated to solving those little issues that pop up when you move from theory to practice. We look at more advanced framework  
options and also hear from people who are using the techniques covered in this course.

Notes

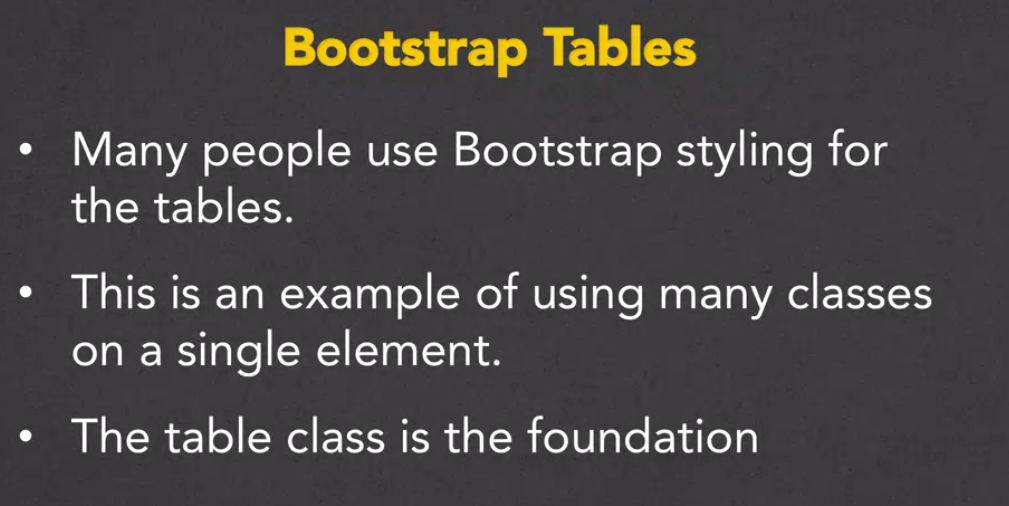
## Responsive Images:

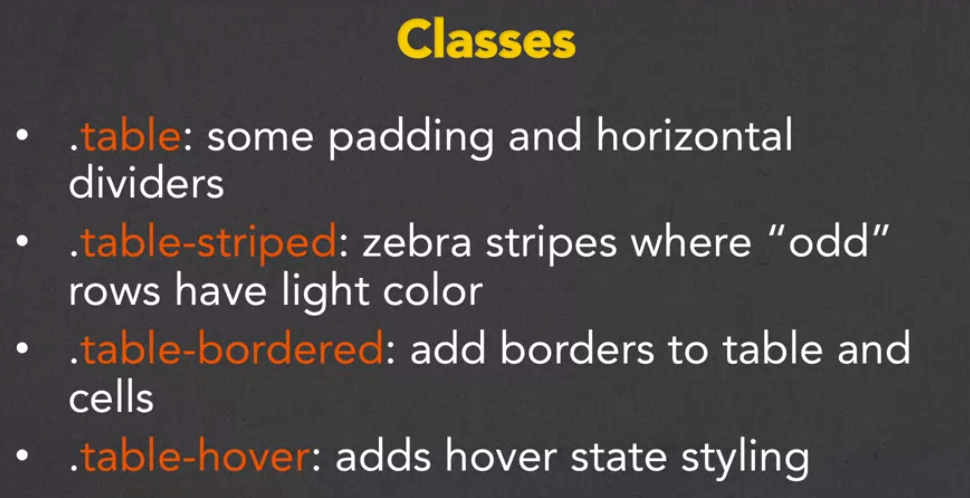
Using fluid measurements (setting width = 100%) will make your images responsive. Also setting the min-width and max-width is also suggested.

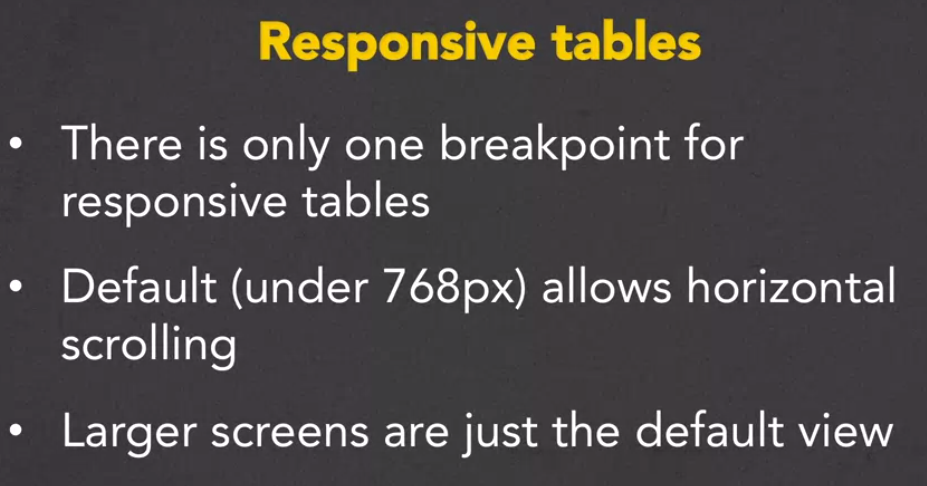


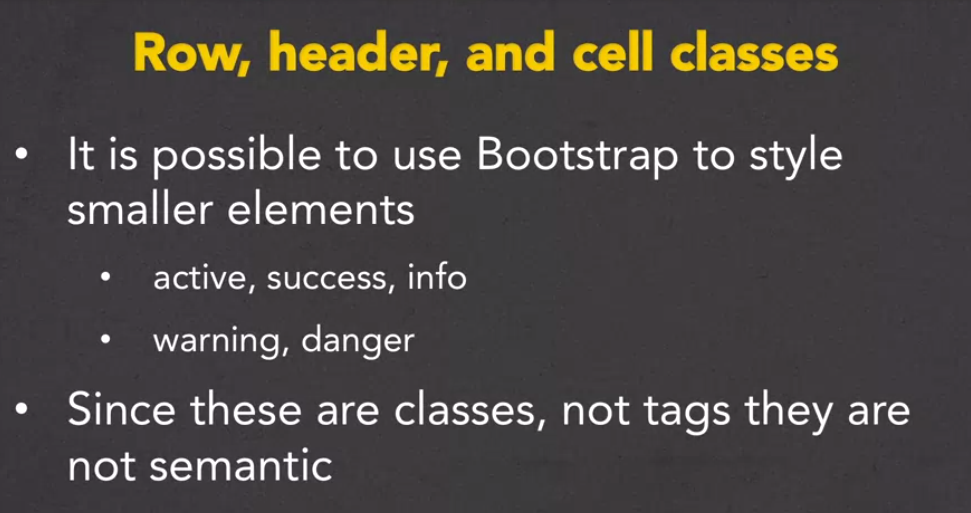


## **Bootstrap Tables**









Using Bootstrap API:

Links

<https://www.bootstrapzero.com/> free templates

<https://startbootstrap.com/> free templates

Closing Thoughts

# Closing Thoughts