Responsive Design

Harry J. Wang, Ph.D.

University of Delaware

Fall 2017

“Learn Responsive Design” Review

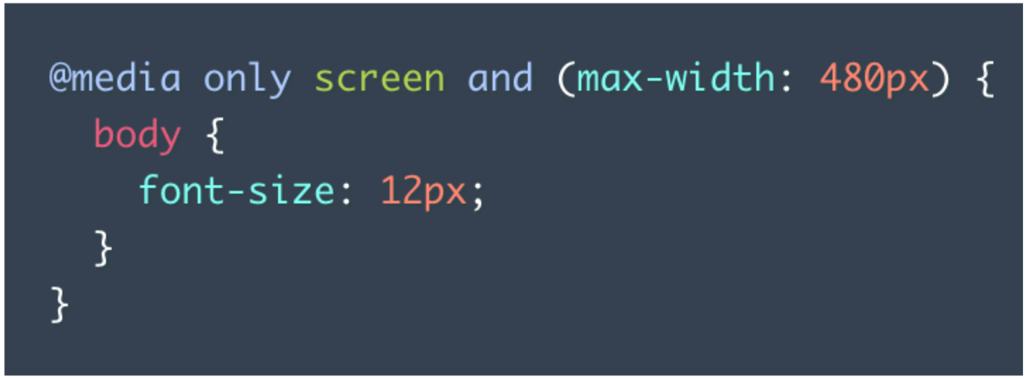
- The first part of this lecture is a review of the following course: https://www.codecademy.com/learn/learn-responsive-design

Relative Measurements

- The unit of **em** sizes font relative to the font size of a parent element.
- The unit of **rem** sizes font relative to the font size of a root element. That root element is the <html> element.
- **Percentages** are commonly used to size box-model features, like the width, height, borders, padding, or margin of an element.
  - When percentages are used to size width and height, child elements will be sized relative to the dimensions of their parent.
  - Percentages can be used to set padding and margin. Horizontal and vertical padding and margin are set relative to the width of a parent element.
- The minimum and maximum width and height of elements can be set using min-width, max-width, min-height and max-height.
- When the height of an image or video is set, then its width can be set to auto so that the media scales proportionally.
- A background image of an HTML element will scale proportionally when its background-size property is set to cover.

Media Queries

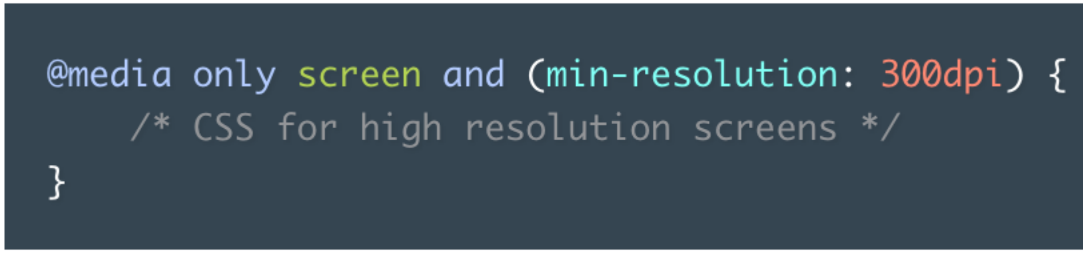
- CSS uses media queries to adapt a website's content to **different screen sizes**.
- With media queries, CSS can detect the size of the current screen and apply different CSS styles depending on the width of the screen.



```
@media only screen and (max-width: 480px) {
  body {
    font-size: 12px;
  }
}
```

Dots Per Inch (DPI)

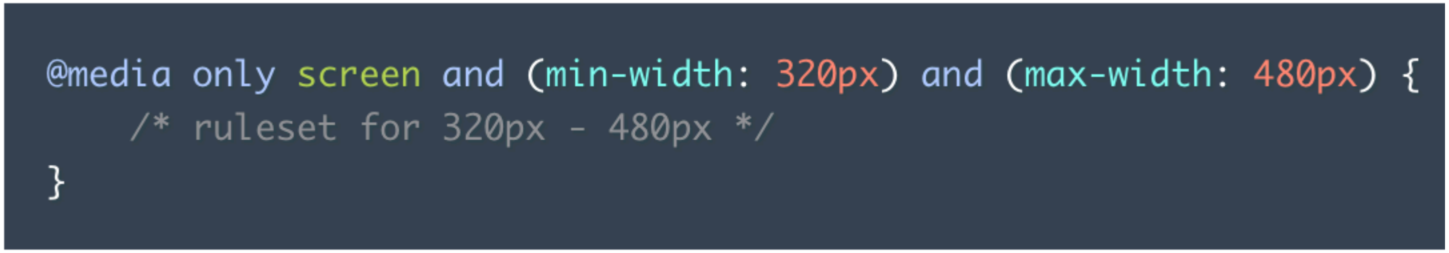
- Another media feature we can target is screen resolution. Many times we will want to supply higher quality media (images, video, etc.)



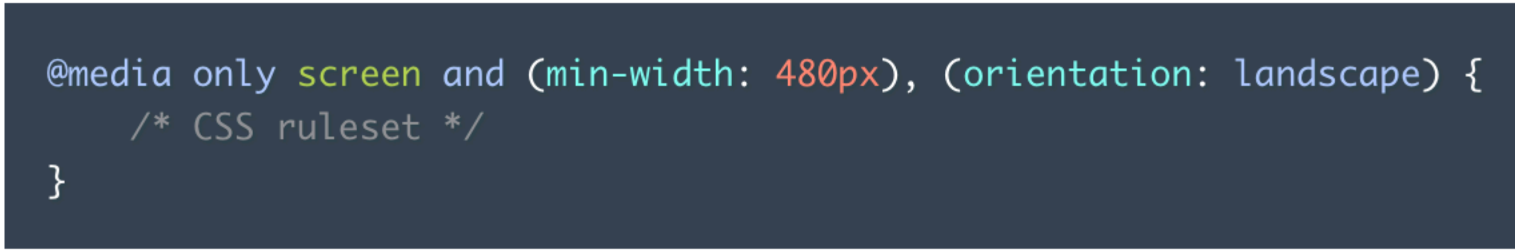
```
@media only screen and (min-resolution: 300dpi) {
    /* CSS for high resolution screens */
}
```

And and Or

- Use ”and” and “,” comma (Or) to specify multiple rules in media queries:



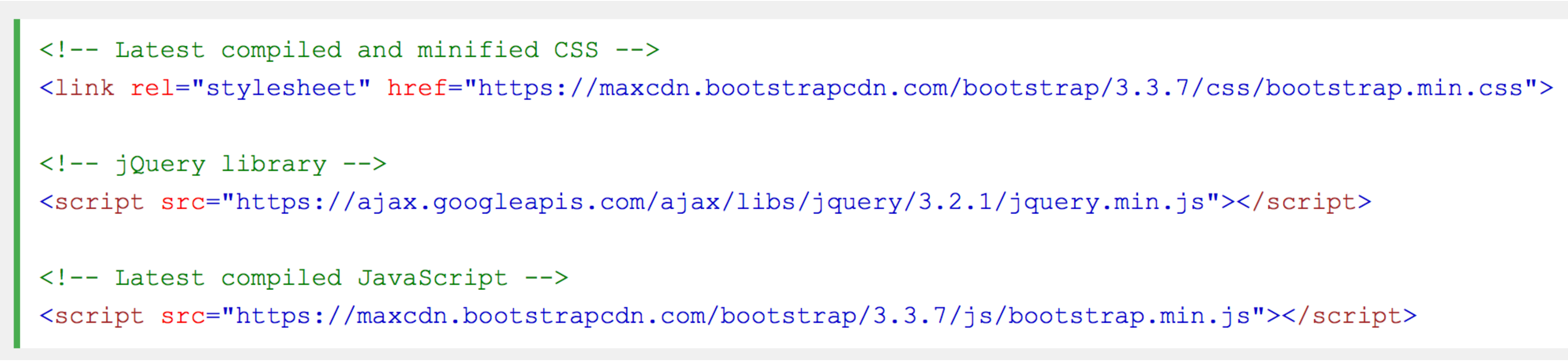
```
@media only screen and (min-width: 320px) and (max-width: 480px) {
    /* ruleset for 320px - 480px */
}
```



```
@media only screen and (min-width: 480px), (orientation: landscape) {
    /* CSS ruleset */
}
```

Bootstrap

- Bootstrap course: https://www.w3schools.com/bootstrap
- Bootstrap is a free front-end framework for faster and easier web development, that is responsive and mobile friendly.
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc.
- Bootstrap use jQuery for JavaScript plugins (like modals, tooltips, etc.).
- Include Bootstrap using CDN (Content Delivery Network) or self-host files



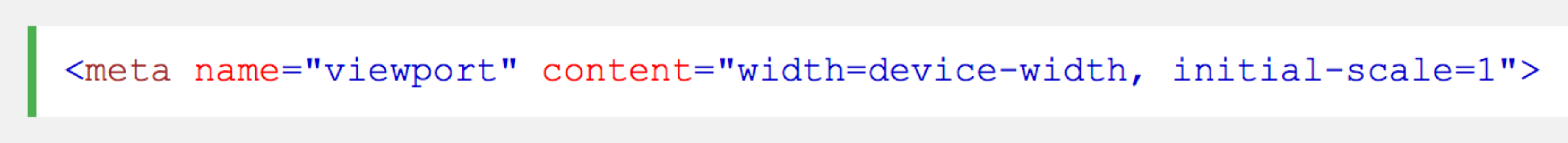
```html
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

Bootstrap Setup

- Meta Data

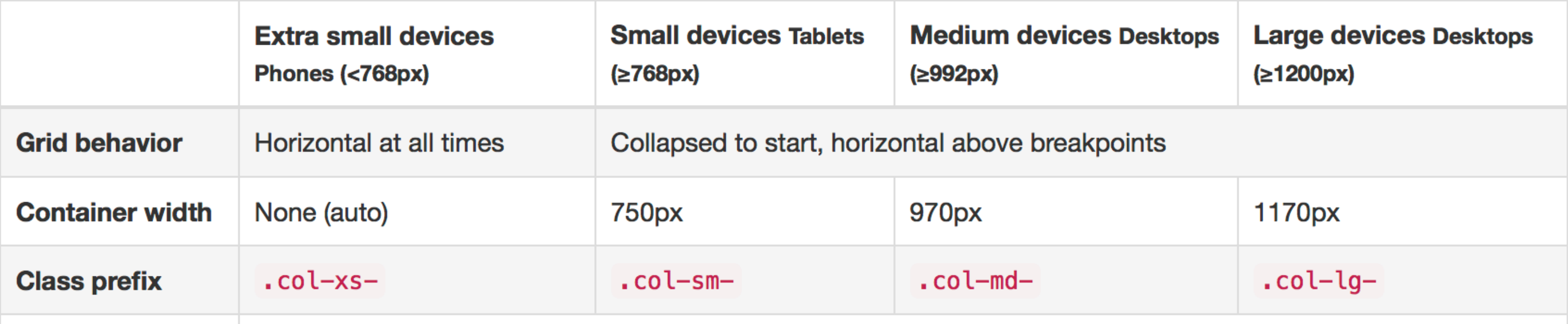


```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

  - The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
  - The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.
- Bootstrap also requires a containing element to wrap site contents.
  - The .container class provides a responsive fixed width container (the fixed size changes according to the screen size)
  - The .container-fluid class provides a full width container, spanning the entire width of the viewport
  - Note: Containers are not nestable

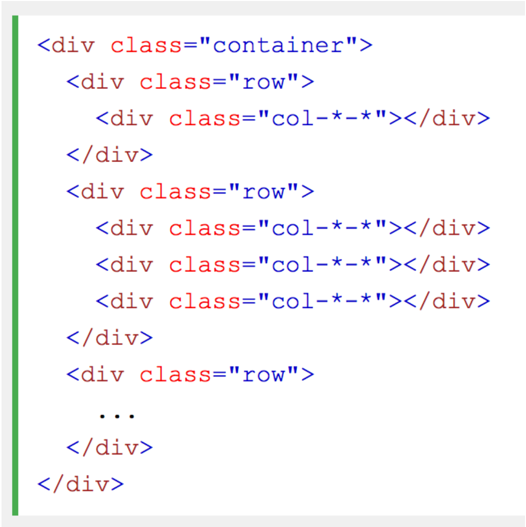
Bootstrap Grid System

- Bootstrap's grid system allows up to 12 columns across the page, which will re-arrange automatically depending on the screen size.

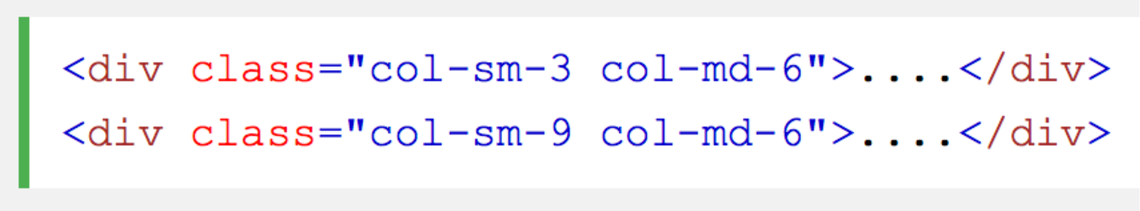


	Extra small devices Phones (<768px)	**Small devices** Tablets (≥768px)	**Medium devices** Desktops (≥992px)	**Large devices** Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`

- Basic Structure: a container, a row, and columns



```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```



```
<div class="col-sm-3 col-md-6">....</div>
<div class="col-sm-9 col-md-6">....</div>
```

"at the small size, look at classes with -sm- in them and use those. At the medium size, look at classes with -md- in them and use those" See an example

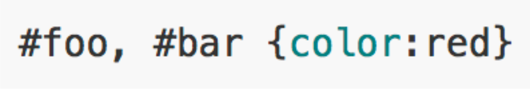
Additional CSS Knowledge

- Combinators in Selcetors
- Pseudo Classes/Elements
- FontAwesome

Source

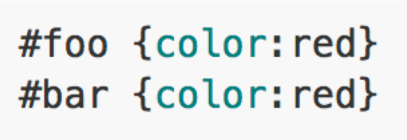
Commas and Spaces in Selector

- The comma is used for grouping when the same rule applies for several selectors.



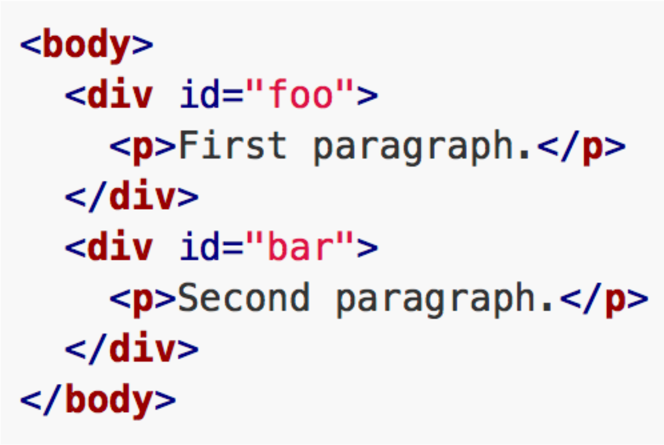
```
#foo, #bar {color:red}
```

=

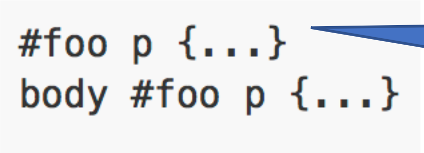


```
#foo {color:red}
#bar {color:red}
```

- The space is a 'descendant combinator' and means that the element matched by the sub-selector to the right of the space is a descendant (child, grandchild, etc.) of the element matched by the sub-selector on the left-hand side of the space.

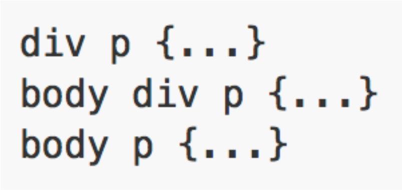


```
<body>
  <div id="foo">
    <p>First paragraph.</p>
  </div>
  <div id="bar">
    <p>Second paragraph.</p>
  </div>
</body>
```



```
#foo p {...}
body #foo p {...}
```

'a paragraph which is a descendant of an element whose ID is "foo".



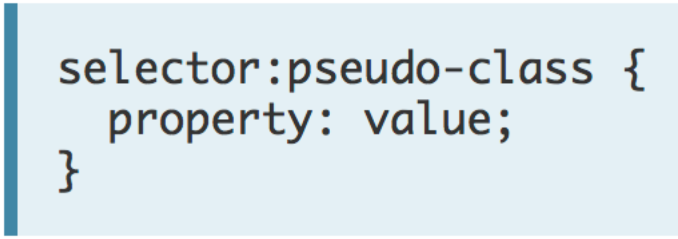
```
div p {...}
body div p {...}
body p {...}
```

Other combinators:

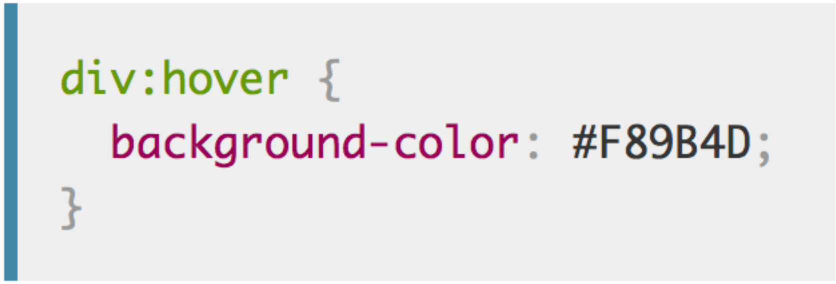
- > (child combinator)
- ~ (sibling combinators)

Pseudo Classes

- A CSS pseudo-class is a keyword added to a selector that specifies a special **state** of the selected element(s). For example, :hover can be used to change a button's color when the user hover s over it.



```
selector:pseudo-class {
  property: value;
}
```

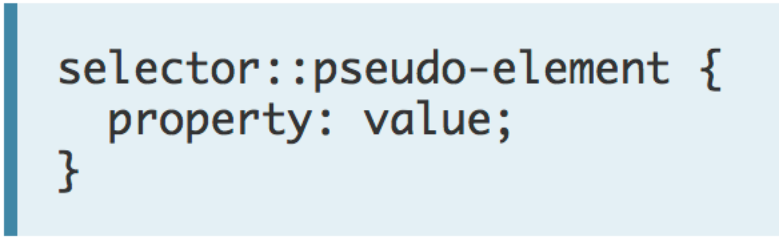


```
div:hover {
  background-color: #F89B4D;
}
```

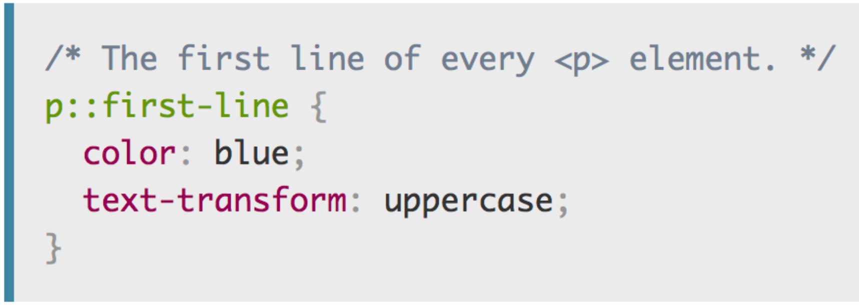
List of Pseudo classes:
https://developer.mozilla.org/en-US/docs/Web/CSS/pseudo-classes

Pseudo Elements

- A CSS pseudo-element is a keyword added to a selector that lets you style a specific **part** of the selected element(s).



```
selector::pseudo-element {
  property: value;
}
```

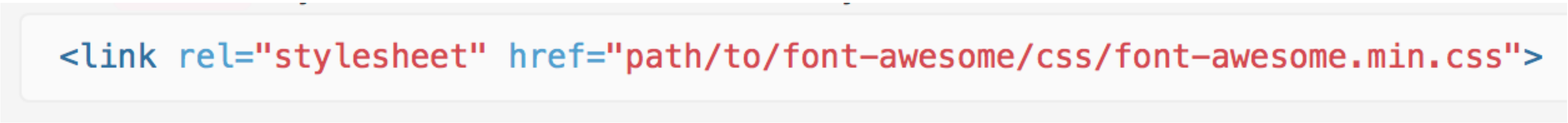


```
/* The first line of every <p> element. */
p::first-line {
  color: blue;
  text-transform: uppercase;
}
```

List of Pseudo classes:
https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements

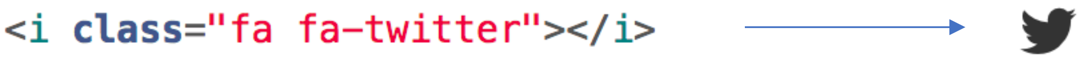
Font Awesome (http://fontawesome.io)

- Font Awesome provides scalable vector icons that can instantly be customized — size, color, drop shadow, and anything that can be done with the power of CSS.
- Include FontAwesome:
  - Copy the entire font-awesome directory into your project.
  - In the <head> of your html, reference the location to your font-awesome.min.css.



```
<link rel="stylesheet" href="path/to/font-awesome/css/font-awesome.min.css">
```

- Then, place Font Awesome icons anywhere with the <i> tag:



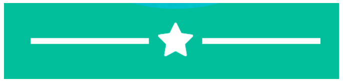
```
<i class="fa fa-twitter"></i>
```

In-class Exercise: Explore Freelancer CSS

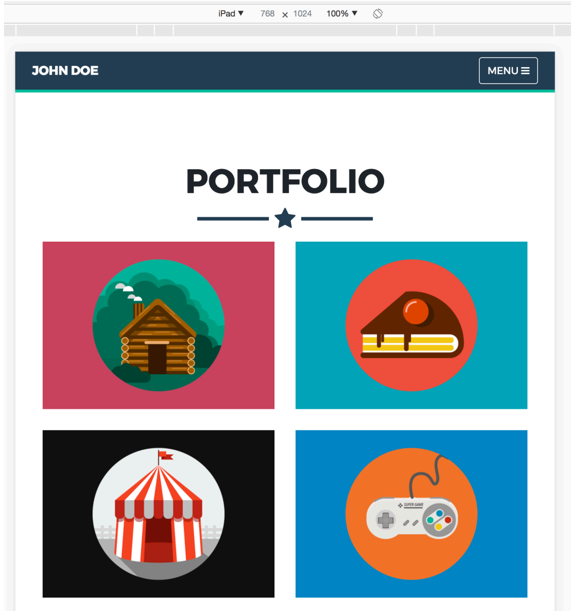
https://github.com/udmis/freelancer

Use ChromeDev Tool to find out:

1. Which CSS rule makes the avatar shown as a circle

2. How to implement: 

3. Make the portfolio
   part to show as follows on iPad:



Questions?