44-563: Unit 02

Developing Web Applications and Services

No class on Monday!

U02: Overview

- Responsive design
- Bootstrap (front-end framework for building responsive apps)

Responsive Design

Accessing web sites

Users access web sites using **different devices**

–Notebook computers, tablets, phones

And **different browsers**

–Chrome, Firefox, Safari, IE, Opera

Challenge: Clients vary considerably

- The **size** and **resolution** of displays vary considerably.
- A site designed for width = 1000 pixels may look very bad on a display with width = 300 pixels.
- Size = width x height (in distance)
- Resolution = width x height (in pixels)
- Pixel = "picture element" (smallest square that can be turned on or off)

Solution: Responsive design

1. Specify **relative sizes** (em, %), not absolute sizes (px).
2. Use **media queries** to detect size and other attributes of the display area.
   a. This allows specification of different styles for small devices vs large devices.
3. Use industry-standard **frameworks** and libraries, e.g. Bootstrap framework

M

Techy Aside: em

- An **em** is the preferred unit, according to W3C, for indicating font size
- 1 em = current font size, so ems scale easily.
- e.g., 1.2em = 1.2 x current font size
- Use font-size to make this happen.
- e.g., h1 {font-size:1.6em;} means that h1 will be 1.6 x the current font size.
- The default size for font size in a browser is 16px.

```
h1 {
    font-size: 2.5em; /* 40px/16=2.5em */
}

h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
 }

p {
    font-size: 0.875em; /* 14px/16=0.875em */
}
```

Techy Aside: Attribute Selectors

The W3 schools demo referenced 2 slides from now uses an **attribute selector,** a CSS selector that selects those tags with a given attribute (and value). For example,

```
[lang="fr"]{ color:pink; }
```

would apply to all tags that have a language attribute with a value of fr (French, I presume.)

```
<span lang="fr">Comment voulez-vous dire rose?</span>
```

The = in the attribute selector implies equality. There are other operators, including *=, which matches when the string in the selector is a *substring* of a tag's attribute value.

E.g., [href*="cbc"]{color:blue; } selects all those tags with href values in which "cbc" is a substring

```
<a href="http://www.cbc.ca">This element would be selected. Fancy that!</a>
```

Techy Aside: Attribute Selectors

In the W3 Schools demo, the selector is [class*="col-"]{ … }, so it selects all those tags with a class whose value *contains* col-

e.g.,

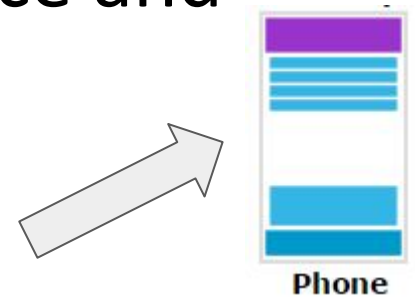
[class*="**col-**"]{color:blue;}.

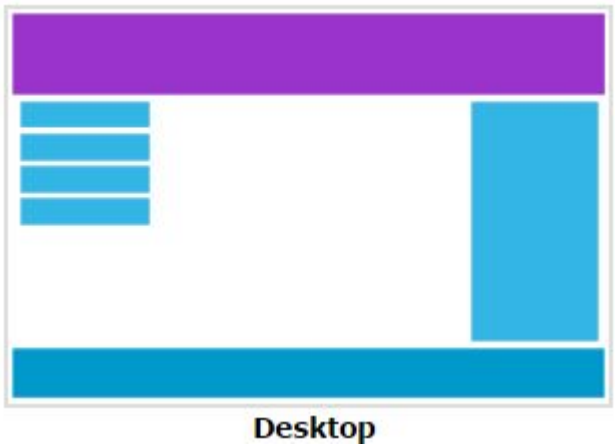
I'm feeling blue, can you cheer me up?
<input type="text" class="**col-**4">I am input, and the color:blue style applies to me</input>

CSS media queries

Ask questions about the client device and display accordingly

- If phone-sized and vertical, then
- If desktop-sized and landscape, then
- Try: W3 schools demo





CSS Media Query syntax

@media not|only *mediatype* and (*expressions*) {
    *CSS to apply when this media query is **true**;*
}

mediatype can be

- all -- all media devices
- print -- for printers
- screen -- for anything that glows in the dark
- etc.

expressions can specify the maximum/minimum width/height of the device/window, among other things, e.g., @media only screen and (min-width: 100) and (max-width: 200) would apply for screens that have a minimum display width between 100-200

See the docs for details or https://css-tricks.com/logic-in-media-queries/ for overview. Not negates. Only if browser understands MQs (not typically needed)

Use responsive design!

- Many applications employ common features (e.g., menus, navbars, grids, jumbotrons, carousels, etc.)
- Don't recreate the wheel - focus on your core content.
- Use industry standard frameworks and libraries, e.g. **Bootstrap** framework

Bootstrap Framework

Bootstrap

- Client-side **framework** for responsive web design (more than just a library)
- http://getbootstrap.com/
- https://getbootstrap.com/docs/3.3/getting-started/ (scroll down for examples)
- Released by Twitter in 2010 as open source

Bootstrap Features

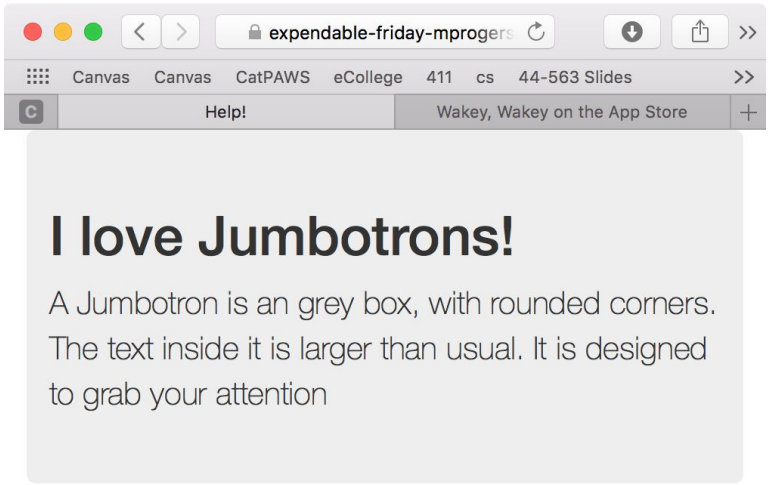
- Beautiful **typography**: HTML tags get a makeover so that they look *magnificent* (and with no effort on your part 👍).
- **Layout** is responsive and works perfectly with mobile, thanks to a sophisticated grid system
- Includes a slew of **components** (dropdowns, buttons, button groups, breadcrumbs, alerts, progress bars, etc.)

Using Bootstrap Features

Once you've got the appropriate files linked in, using Bootstrap is a 💩. Since they've CSS'd the html tag, all of your html tags will Just Look Better™. You can also use specific CSS classes using class="*bsc*", where bsc is some Bootstrap class, to produce a specific effect.

e.g.,

```
<div class="jumbotron">
<h1>I love Jumbotrons!</h1>
<p>A Jumbotron is an grey box, with rounded corners. The text inside it is larger than usual. It is designed to grab your attention</p>
</div>
```



How to add Bootstrap

- Start with your html (typically "index.html")
- Add the bootstrap **css** and **js** to your html (just as if you were adding them locally)
  - During development, just link to the CDN URL
  - For deployment (and performance), you want a local copy (with minimized, optimized versions for better performance)

CDN = Content Delivery System, a system designed to deliver all sorts of web content through a numerous, geographically disparate proxy servers and data centers

OPTION 1

Easy add with CDN

Get Bootstrap from the **content delivery network** (CDN):

https://www.bootstrapcdn.com

3 easy steps:

1. Include <meta> tags for small screens, disabling user scaling
2. **<link>** to Bootstrap css
3. **<script>** to add Bootstrap JS and jQuery (a JavaScript library it depends on that we'll learn more about soon)

OPTION 2

Or start with the template

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  </body>
</html>
```

From: https://getbootstrap.com/docs/3.3/getting-started/

HTML: adding CSS & JavaScript

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>

<body>
....
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
```

Know tags to add JS and CSS (this will be on the exam)

- *The **meta** tag should come **first** in the head*
- *There is **no closing tag** for **<meta>***
- *There is **no closing tag** for **<link>***
- *Use a separate (not self-closing />) tag for **</script>***
- *If you ignore the last point, this will be you:* 
- Placing <script> just before </body> makes faster page loads

What's this JavaScript thing?

Bootstrap uses JavaScript

Bootstrap uses JavaScript to implement responsive functionality.

JavaScript is a programming language implemented by browsers.

JavaScript is not Java nor is it a subset of Java.

JavaScript preview

Goto: http://www.tutorialspoint.com/javascript/javascript_dialog_boxes.htm

app.html

```
<html>
<head>
<script src="app.js"></script>
</head>

<body>

<p>Try this: </p>

<form>
  <input type="button"
    value="Click Here!"
    onclick="warn();" />
</form>

</body>
</html>
```

app.js

```
function warn() {
 alert ("Danger, Will Robinson!")
}


function getConfirmation(){
  let retVal = confirm("Continue?")
  if( retVal == true){ return true}
  else{ return false}
}


function getValue(){
  let retVal = prompt("Enter name: ", "Ramya")
}
```

Swap out the function name from warn() to getConfirmation() to getValue()

Tips for learning JavaScript

Much of 44-542 OO with Java applies.

All programming deals with giving **instructions**.

Sometimes you want to **repeat** an instruction.

```
while (awake) { look_for_pokemon(); }
```

Sometimes, you want to **branch**, based on conditions

```
if (rainy) {  get_umbrella();   }  else { play_cricket(); }
```

We'll cover JavaScript more next week. For now, just know that:
a) JavaScript will provide the logic in our applications
b) JavaScript is really easy

Back to Bootstrap

Bootstrap containers

A Bootstrap-based site needs a **containing element** - a div with either a **.container** or **.container-fluid** class

- *grid system specs go here*
- *grid system specs go here*

- **.container** is responsive, but **fixed-width**: the width will depend on the size of the screen and your content will be centered within it,
- **.container-fluid** fills the **entire screen**

Responsive Classes

- Adding a **-xs**, **-sm**, **-md**, or **-lg** suffix affects how an element will be rendered when displayed on a particular size of screen.
  - .btn-xs styles a button with a font-size of 12px, line width of 1.5, padding of 1px 5px
  - .btn-sm styles a button with a font-size of 12px, line width of 1.5, padding of 5px 10px
- Can make elements **hidden**, or only **visible**, based on device size
  - This will be hidden on a large device
  - <div class="visible-xs-block>This will be visible only on extra small devices
- Add **multiple classes** if you wish an element to be visible (or invisible) in **several sizes**
  - You won’t see this on medium or large devices

Rendering Images

To make an image responsive-friendly, use **img-responsive** class:

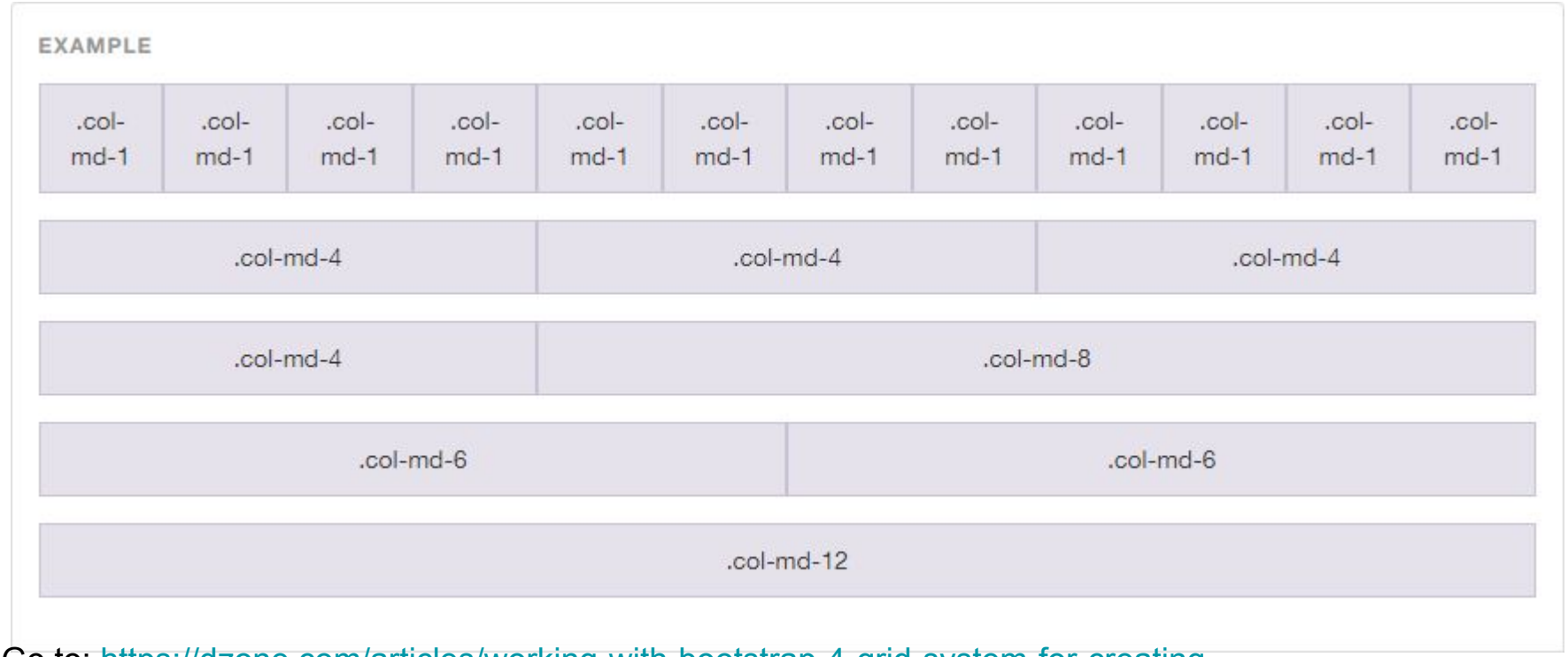
```
.img-responsive {max-width:100%;height:auto; display:block;}
```

Use **center-block** to center the image

.img-rounded { border-radius: 6px;}

.img-circle {border-radius:50%;}

Bootstrap Grid System



Go to: https://dzone.com/articles/working-with-bootstrap-4-grid-system-for-creating

- `.col-xs-1` spans one column and `.col-xs-8` spans eight columns.
- `xs` means colspan applies to extra small devices.
- `sm`, `md`, and `lg` for small, medium, and large respectively

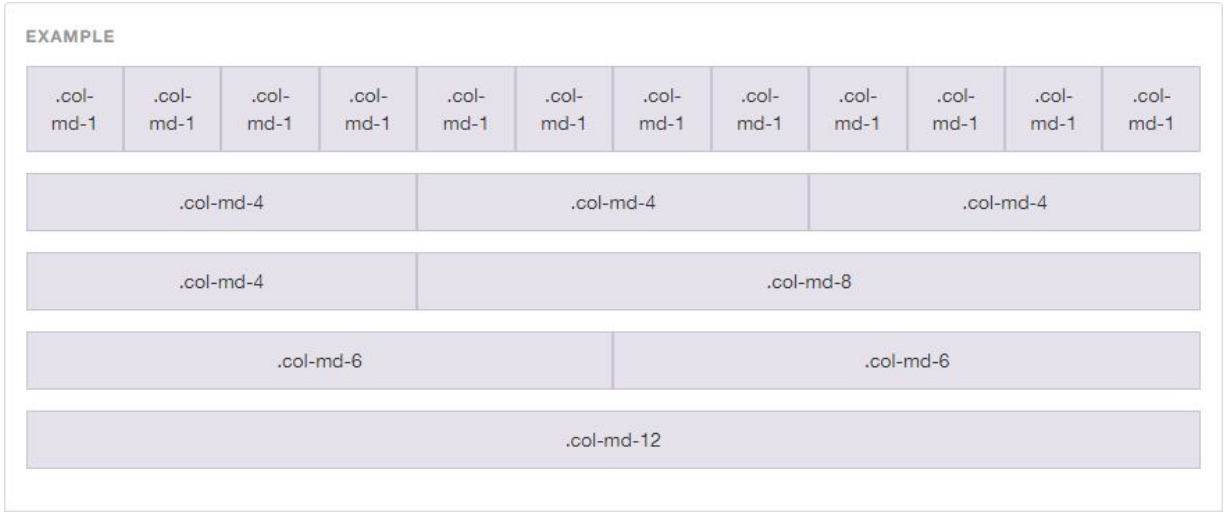
< 768px Extra small devices *Phones*

≥ 768px Small devices *Tablets*

≥ 992px Medium devices *Desktops*

≥ 1200px Large devices *Desktops*

Grid System: Sliding blocks



- Device size applies to that device size **and up**,
- e.g., col-sm-* will apply to tablets (sm), medium (md) and large (lg) desktops. On smaller devices, the columns will stack.
- col-md-* will apply just to medium and large desktops, and as before, columns will stack on smaller devices.

```
<div class="bs-example">
    <!-- Bootstrap Grid -->
    <div class="row  show-grid">
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
      <div class="col-xs-1">.col-xs-1</div>
    </div>
    <div class="row  show-grid">
      <div class="col-xs-2">.col-xs-2</div>
      <div class="col-xs-3">.col-xs-3</div>
      <div class="col-xs-7">.col-xs-7</div>
    </div>
    <div class="row show-grid">
      <div class="col-xs-4">.col-xs-4</div>
      <div class="col-xs-4">.col-xs-4</div>
      <div class="col-xs-4">.col-xs-4</div>
    </div>
    <div class="row show-grid">
      <div class="col-xs-5">.col-xs-5</div>
      <div class="col-xs-7">.col-xs-7</div>
    </div>
    <div class="row show-grid">
      <div class="col-xs-6">.col-xs-6</div>
      <div class="col-xs-6">.col-xs-6</div>
    </div>
    <div class="row show-grid">
      <div class="col-xs-12">.col-xs-12</div>
    </div>
</div>
```

Grid Example

- To create a grid, use **row** to specify a row.
- Within each row, use **col-*ss*-*n***, replacing ss with xs, sm, md, lg, and n from 1-12, to indicate how many columns should be consumed
- Total # of classes should equal 12


```
<div class="container-fluid yellow-background">
   <div class="row" style="background-color:powderblue">
     <div class="col-xs-8" style="color:white;background-color:#ffcc00">
       First 8 columns
     </div>
     <div class="col-xs-4" style="color:white;background-color:#00cc00">
       Last 4 columns
     </div>
     <div class="row" style="background-color:powderblue">
     <div class="col-xs-2" style="color:purple;background-color:#ffffff">
       First 2 columns are going right here
     </div>
     <div class="col-xs-8" style="color:white;background-color:#bbbbbb">
      Middle 8 will follow in the fullness of time, don't you think?
     </div>
     <div class="col-xs-2" style="color:white;background-color:#00ccaa">
       The last 2, heroically uninteresting columns.
     </div>
   </div>
 </div>
```

For different behavior based on screen size, just add more classes:

```
<div class="col-md-3 col-sm-6 col-xs-12 text-center">
```

More about Bootstrap

Go to:

https://getbootstrap.com/docs/3.3/getting-started/

Change browser size and watch how it responds.

Explore how to make this:

```
<body>
    <div class="container"><div class="row">
    <header class="col-md-12 text-center"><h2>Bootstrap's 4 Grid System</h2></header>
    <div class="col-md-3 text-center"><h3>Heading 1</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 2</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 3</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 4</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 5</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 6</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 7</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    <div class="col-md-3 text-center"><h3>Heading 8</h3>Lorem ipsum dolor sit amet..
    <a class="btn btn-default" href="#">Read More →</a></div>
    </div></div>
```

We had said that we should have 12 columns per row, but here there are more. In this case, excess columns wrap around to a new row.

https://dzone.com/articles/working-with-bootstrap-4-grid-system-for-creating

Add tablet view

Landscape view can be achieved using the **col-md-*** classes.

Portrait view can be achieved by using **col-sm**-* classes such as the col-sm-6 class.

Replace with

Add mobile view

use the **col-xs-12** class

Replace with

More styling

Get started with Initizer:

http://www.initializr.com/

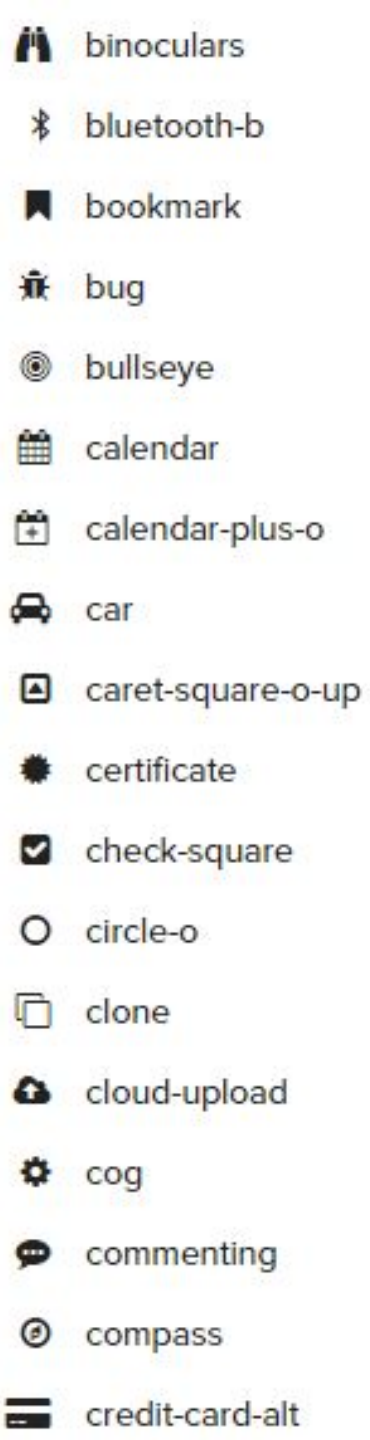
Check initializr out. This is how professional web developers start their projects, just as mobile and desktop developers do: with a template.

Font Awesome (free icons)

http://fontawesome.io/icons/

and others:

http://tagliala.github.io/vectoriconsroundup/



Bootstrap

- http://getbootstrap.com/
- http://www.initializr.com/ <-- customize your setup
- Free UI kits: http://presentation.creative-tim.com/
- getbootstrap.com's docs are great

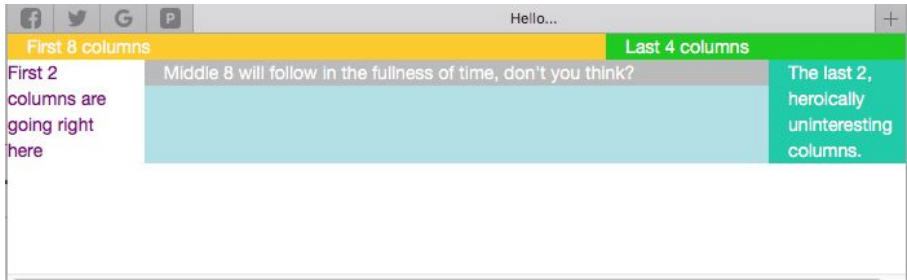
M02 Example Code

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
  <title>Bootstrap 101 Template</title>

  <!-- Bootstrap -->
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
</head>

<body>
  <h1>Hello, world!</h1>
  <div class="container-fluid yellow-background">
    <div class="row" style="background-color:powderblue">
      <div class="col-xs-8" style="color:white;background-color:#ffcc00">
        First 8 columns
      </div>
      <div class="col-xs-4" style="color:white;background-color:#00cc00">
        Last 4 columns
      </div>
    </div>
    <div class="row" style="background-color:powderblue">
      <div class="col-xs-2" style="color:purple;background-color:#ffffff">
        First 2 columns are going right here
      </div>
      <div class="col-xs-8" style="color:white;background-color:#bbbbbb">
        Middle 8 will follow in the fullness of time, don't you think?
      </div>
      <div class="col-xs-2" style="color:white;background-color:#00ccaa">
        The last 2, heroically uninteresting columns.
      </div>
    </div>
  </div>

  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>

</html>
```

M02

Copy & paste into a new bootstrap.html file to try it out.

Add a new row with at least 2 parts. Color, add text, and split it as you like



C9.io Example https://github.com/mprogers/initializr

1. Login to c9.io
2. Create a new workspace, **bootstraplives**
3. In box labeled "Clone from Git or Mercurial URL", enter the following:
   a. github.com/mprogers/initializr.git
4. Click on Create Workspace

Congratulations, you now have a playground in which you can fiddle around with Bootstrap. With a partner, do the following:

1. Resize the window to see its behavior at various sizes
2. Locate the col-md-* classes that make it possible
3. Remove everything below the Jumbotron, and replace it with 3 columns that are 4/12, 6/12, and 2/12 of the screen on devices that are sm, md or lg. [You can populate the divs with the same Latin text provided there, or generate your own.]
4. Create CSS classes .leftCoast, .midContinent and .rightCoast in the main.css file: they should change the background-color to 3 different colors (your choice)
5. Apply those classes to the 4/12, 6/12, and 2/12
6. Make the 2/12 column hidden when the size is sm, and verify that it works.
7. Visit the W3Schools tutorial, and find at least **5** interesting new things relating to Bootstrap (at the end of the class, we will share)

Friday:
Quiz / WS topic selection / A01