

# Frameworks

# Frameworks

- The term “web framework” can mean many things, it depends upon who you ask.
- Agreement, a framework should make your job easier by providing code and structure.

# What does “framework” mean here?

- Front-end developers
  - CSS, JavaScript, jQuery
- Back-end developers
  - Routing, resources, security



# Popular front-end frameworks

- Bootstrap (2011)
  - Its popularity makes it... popular

# What it means for you

- Many people build their sites directly from templates.
- Others use their own code and add functionality.
- Even “pure” coders should know the basics of one popular framework

© Colleen van Lent  
University of Michigan  
School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.



# Introduction to Bootstrap

# What is it?

- Framework for creating web sites
- Focuses on responsive, mobile-first approach
- Consists of
  - CSS and HTML templates
  - JavaScript extensions
- Emphasis on interfaces and layouts



# Who uses it?

- Many popular sites are (or were) built using Bootstrap
  - [nba.com](https://www.nba.com)
  - [target.com](https://www.target.com)
  - [walmart.com](https://www.walmart.com)
  - [codeacademy.com](https://www.codecademy.com)
  - [bloomberg.com](https://www.bloomberg.com)
  - [airbnb](https://www.airbnb.com)

# Bootstrap benefits

- 12-column grid system
  - Helps with spacing issues
  - Built-in responsive design
- Common jQuery functionalities
  - Accordion, Drop-down menus, Carousel
- Familiar “look and feel”
  - Many sites use Bootstrap
  - Makes your forms look “legitimate”

# Why you should use it

- Fast development
- Platform Independent
- Responsive by default
- Customizable



# Why you shouldn't use it....

- Doesn't follow best practices
  - Content and layout are intertwined
  - CSS frameworks and semantics
- Can be resource-heavy
- The look is somewhat generic
  - This can be good or bad.

# What we will cover

- There are two ways to use Bootstrap
  - As a supplement to your style
  - As a theme that you expand upon
- We will talk about some of the basics, but there are still many parts left untouched.
- It is important to test often when using code that isn't yours.

© Colleen van Lent  
University of Michigan  
School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.





# Getting Started With Bootstrap 3

# 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



# Saving your own copy of Bootstrap

- You can download a copy of Bootstrap at <http://getbootstrap.com>
- Use the default values or select <http://getbootstrap.com/customize/>
- Make sure you know where you saved your files!

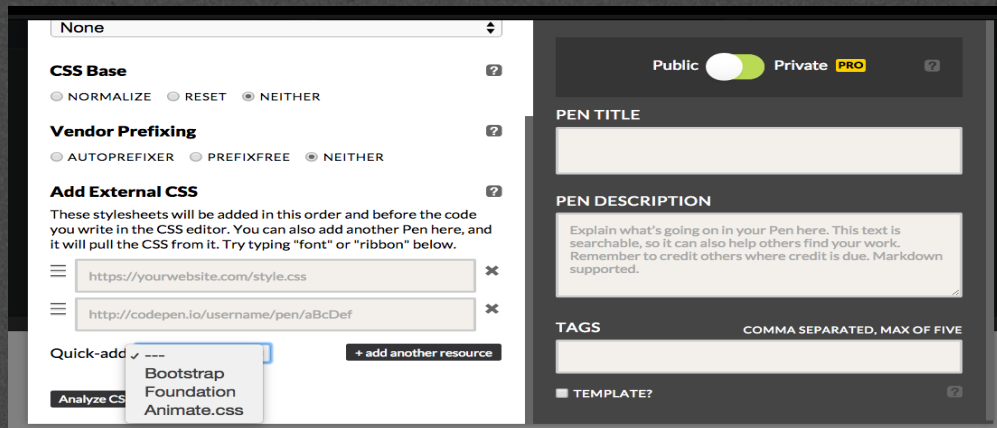


## Using a CDN

- A CDN is a content delivery network
- Provides a way to connect to the Bootstrap code using an absolute reference
  - <http://getbootstrap.com/getting-started/>

# Using Bootstrap with CodePen

- You can add Bootstrap to CodePen in the Settings.
- Select Settings → CSS → Add external CSS





# Template

- It is common to start with a template file.

<http://getbootstrap.com/getting-started/#examples>

- A modified version from getbootstrap.com can be found here:

[RD:Bootstrap Template](#)



# How do you know it is working?

- For such a simple file, it may be hard at first to see if it is working.
- To test:
  - Comment out link to css
  - Use Inspect Element
    - This method is definitely preferred, particularly as you start to add your own styles.

## Don't forget...

- To use element inspector.
- To include the JavaScript at the bottom.
- That you can customize bootstrap defaults.
- That you can overwrite bootstrap defaults.





© Colleen van Lent  
University of Michigan  
School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.



# Bootstrap Grid System

## It's not a table....

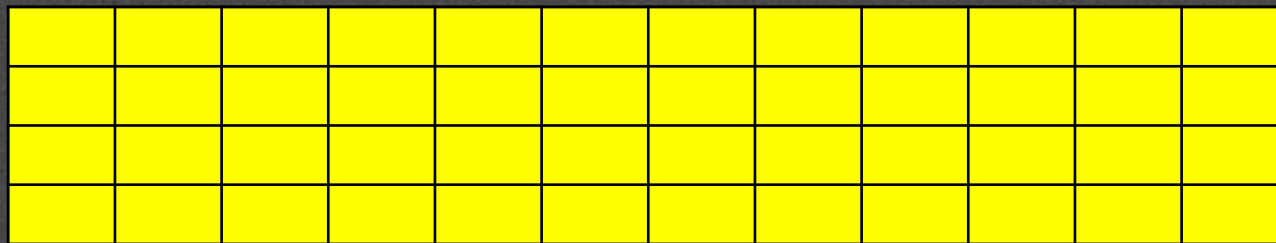
- The Bootstrap layout is based on a 12 column grid

12 columns = 100%

6 columns = 50%

9 columns = 75%

3 columns = 25%



# 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">
```



## Setting width with col-xx-yy

- The class col-xx-yy always starts with “col”
- The “xx” component is the viewport size:
  - xs, sm, md, lg
- The “yy” component is the number of columns 0..12

## col-xx-yy

- col-xs-12
- col-sm-8
- col-md-4
- col-lg-3
- These are often combined! Think photos....  

```
<img src = "pic.jpg" class="col-xs-12 col-sm-6  
col-md-3 col-lg-2">
```





# Combining elements

- What if you want two elements next to each other, but only on larger screen?

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

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



## But it is actually easier than that...

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

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

```
<div class="col-md-3">Yellow Part</div>
```

## Positioning 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

# Conclusion

- Bootstrap is mobile first, larger screens inherit values from smaller screens.
- Review <http://getbootstrap.com/examples/grid/>
- Do the math!! If you want two elements to be next to each other on a larger screen, make sure their width adds up to no more than 12.

© Colleen van Lent

University of Michigan

School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.





# Bootstrap Navigation

# Navigation Bars

- One of the components that gives Bootstrap its familiar “look and feel” is the navigation options.
- The nav class is combined with other classes to create each style



# Making a navigation bar

- Decide what type of links you want:
  - nav-tabs vs nav-pills
    - [RD:nav-tabs](#)
    - [RD: nav-pills](#)
- Decide on layout (horizontal, stack, justified, etc.)
  - nav-stacked, nav-justified



# Drop downs

- To add dropdown menus you need to include the Bootstrap js files AND a link to the jQuery.
- Bootstrap  
example: <http://getbootstrap.com/components/#pills-with-dropdowns>

[RD:dropdowns](#)

## navbar class

- The navbar class serves as a navigation header for your application or site.
- Positioning includes:
  - navbar-static-top
  - navbar-fixed-top
  - navbar-fixed-bottom

[RD:navbar](#)



# Accessibility

- Using a “nav” class does not convey semantics.
- Use the `<nav>` tag or ARIA attribute `role=“navigation”`.



© Colleen van Lent

University of Michigan

School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.

# Closing

# What Have We Covered?

- Write basic CSS rules for styling.
- Utilize tools to check our CSS (Inspect Element, Wave, etc).
- Use fluid measurements.
- Write media queries for different screen sizes.



## What Else?

- Write rules to facilitate accessibility with motion and color scheme preferences.
- Discuss the importance of the mobile-first paradigm.
- Understand the grid system of Bootstrap.

## What Next?

- More practice until you feel more comfortable.
- Look more into frontend frameworks, eg React, Angular, etc.
- Explore accessibility in more depth.
- Use your knowledge to help elevate existing team projects.

# Congratulations!

- You can also take a moment to be proud of yourself.

## Great job!



© Colleen van Lent  
University of Michigan  
School of Information

Unless otherwise noted, this work is licensed under the  
CC BY-NC 4.0 license.