Lecture Materials - Week 3

Frameworks

- framework = a tool that provides code & structure to make your job easier

- (for front-end development) frameworks for: CSS
  JavaScript
  jQuery

- (for back-end development) frameworks for: routing
  resources
  security

- popular front-end frameworks (context: in 2015): Bootstrap (2011)
  Foundation by Zurb (2011)
  Semantic UI (2013)
  Pure by Yahoo! (2013)
  UIkit by YOOtheme (2013)

  approaches: build site directly from templates
  use own code & some frameworks (eg. to add functionality)

- regardless if you use it or not → important to know the basics of 1 popular framework

Introduction to Twitter Bootstrap 3

- Bootstrap = framework for creating websites
  consists of CSS & HTML templates & JavaScript extensions
  focuses on responsive, mobile-first approach
  emphasis on interfaces & layout

- Bootstrap 3 features: 12-column grid system
  → helps spacing issues
  → built-in responsive design
  common jQuery functions
  - accordion
  - drop-down menus
  - carousel

  ] most common

- Bootstrap pros: fast development
  platform independent
  responsive by default
  customizable
  familiarity & legitimacy

- Bootstrap cons: doesn't follow best practices
  - content & layout are intertwined
  can be resource-heavy

can look generic

- ways to use Bootstrap: as a supplement to your code
  as a theme that you expand on

- important to test code that isn't yours often

Bootstrap Breakpoints

- Bootstrap hardcodes default breakpoints for different viewports
- set own breakpoints/change default values: customize before downloading
  modify css file
  write own media queries

- mobile-first → plan for smallest screen & make changes if it's bigger

- Bootstrap sizes: custom iPhone retina — smallest
  xs-
  sm-
  md-
  lg- — biggest

- custom, iPhone retina = width = 0-320 px
  problematic screen size → browsers don't return a size < 300 even when screen size is small

  - don't use! lol

- xs- = extra small devices
  min. width = 480 px

  → (eg. most phones): in portrait mode (sometimes even in landscape position)

- sm- = small devices
  min. width = 768 px

  → (eg. most tablets)

  → (eg. desktop browsers opened in a small window size)

- md- = medium devices
  min. width = 992 px

  → (eg. desktops)

  → (eg. tablet or phone if screen is magnified)

- lg- = large devices
  min. width = 1200 px

→ (eg. wide screens)

→ (eg. TVs)

→ (eg. smaller devices if magnified)

- remember abbreviations > exact breakpoint values

Getting Started with Bootstrap 3

- access Bootstrap CSS & JS code: download a copy
  use absolute reference
  link on CodePen

- download & save own copy of Bootstrap: http://getbootstrap.com - normal
  http://getbootstrap.com/customize - customize default values

- connect to code via absolute reference: use a CDN = content delivery network
  always have most up to date version
  include Bootstrap CDN code in HTML file
  - css in header
  - js at the end/bottom, right before </body>
  http://getbootstrap.com/getting-started

- link in CodePen: 1) settings →
  2) css tab — 2) js tab
  3) add external stylesheets/pens — 3) add external scripts/pens
  4) search & select twitter-bootstrap ↵

- common to start w/ a template file
- test if Bootstrap is working: comment out link to css
  use inspect element - preferred method

- don't forget: can customize bootstrap defaults
  can override " "

Bootstrap 3 Grid System

- Bootstrap layout based on a 12 column grid

  - 3 columns = 25% width
    6 columns = 50% width
    9 columns = 75% width
    12 columns = 100% width

- every grid consists of: container class
  row class
  col class(es)

- format:

- setting width

- format: col-xx-yy ⟶ col-md-4

- xx = viewport size : xs
sm
md
lg

- yy = # of columns : 0-12

⟶ (eg. <img src = "pic.jpg" class = "col-xs-12 col-sm-6 col-md-3 col-lg-2">): on a xs device, pic takes up whole screen
on a sm device, 50% of screen
on a md device, 25% of screen
on a lg device, 17% of screen

- side-by-side elements only on larger screens

⟶ (eg. left element
 right element) : 8+3=11<12 → okay to not use whole screen

- col-xs-12 unnecessary if element is block → by default it will take up all columns & Bootstrap is mobile first

- positioning classes (applicable on md+ viewports): push
pull

- push = move yy columns to the right

- format: col-xx-push-yy

- pull = move yy columns to the left

- format: col-xx-pull-yy

- responsive utility classes: hidden
visible
sr-only

- hidden = content will be hidden only on given screen size

- format: hidden-xx

- visible = content will only be visible on the given screen size
  - format: visible-xx
- sr-only = content hidden on all devices except screen readers
- larger screens inherit values from smaller screens
  → (eg. xs affects xs, sm, md, lg
  md affects md, lg)
- review: https://getbootstrap.com/examples/grid/

Bootstrap Navigation

- combine nav class w/ other classes to style
- types of links: nav-tabs
  nav-pills
- layout options: nav-stacked
  nav-justified
- for dropdown menus include: Bootstrap js files
  link to jQuery
- navbar class = navigation header for site/app
- navbar positioning classes: navbar-static-top
  navbar-fixed-top
  navbar-fixed-bottom
- nav class doesn't convey semantics
- for accessibility use: <nav> tag
  role = "navigation" ARIA attribute