# Bootstrap Essential Training

Initial Test

* Apparently adding a row does not get rid of the padding that appears on eithr side of a container

Getting Starte

* Bootstrap is
  + A framework
  + Mobile First
  + 12-column Responsive Design
  + Battle Tested
  + Available in less and sass versions
* Framework Means
  + You learn how to write how the framework wants you to write
* When to use?
  + Small quick projects
  + Prototyping and scaffolding
  + Can also be great for larger projects
  + Customize your install
* CDn
  + Content delivery network
  + Browser checks cache to see if browser has been to similar site with same..
* Bootstrap REQUIRES jQuery
* Install 4 way
  + Download css and js
  + Use cdn
  + Download “source files”
    - Everything including orig sass files and documentation
  + Use Package Manager
    - Npm
    - Ruby gems
    - …
* Reboot files are special code that resets bootstrap in the browser
* Map files that map the css to the sass where it was originally created
* Map files that map to javascript are for different browser usability
* Bundle javascript contains popper
  + Can also just install that separately

Minimum Template

* Download vs CDN
* Need to also add jQuery
* May also need popper
* Basic

Basic Typography

* CSS code that comes with boostrap overwrides default browser behavior
* Special portion at beginning of bootstrap styles referred to as Reboot.css so it looks similar across browsers
* Uses rems
* Vertical margins can collapse in css
  + Confusing to calc proper spacing
  + All margins are put at bottoms of elements
  + Will have to add spacing at top of things if necessary
* Uses inherit when possible
* When writing own css, makes it a lot easier, bc we don’t have to overwrite
* Default uses border-box sizing for box-sizing
* Div 200px wide, takes into account the padding and border sizes
* Default sans-serif fonts
* Special Styles
  + Match headlines
  + Create lead paragraph

## Basic Typography

* Reboot normalizes styles
* Also makes it easy to override bootstrap styles with own css
* Container takes care of the breakpoints
* Expands body content on the break
* Lead paragraphs
  + Makes paragraph a little bigger
* Basic bootstrap css will change the standard browser styles

## Typographic Utilities

* Text-justify
  + Left or right
* Text-nowrap
  + Let run infinitely
* Text (-xx) –pos
  + - Text-left
    - Text-sm-center
      * Centers at small breakpoint
        + Which is greater at 576px
* Text-lowercase
* Text-uppercase
* Text-capitalize
* Font-weight-bold
* Font-weight-normal
* Font-italic

## Lists & Blockquotes

* Common css elements
* List-unstyles
  + No bullets
  + Realigned to the left
* List-inline
  + On ul
* List-inline-item
  + On each LI
* Ctrl + shift + l
  + To select all occurences
  + Multi cursor
* Blockquote-reverse
  + Doesn’t seem to work
* Blockquote-footer
  + Footer dash

## Using Colors

* Texts, links and backgrounds
* Text-COLOR
  + Primary
  + Secondary
  + Success
  + Dancger
  + Warning
  + Info
  + Light
  + Dark
  + White
* Bg-<color>
* Sometimes referred to contextual color names
* “faded” doesn’t seem to work

## Images

* Img-fluid
  + Responsive images
  + Add class to image
  + Adjusts to be the width of the container
  + Will respond to the containers size
* Rounded
* Rounded-DIR
  + Direction
  + Tip right bottom left
  + Only rounded at those places
* Rounded-circle
  + Oval shaped imaged
  + If image is square, gives you circle
* Rounded-0
  + Takes out the roundness
* Img-thumbnail
  + Slightly rounded image
  + 1px border
* Be default images are inline elements
* Float-left, float-right
* Text-center
  + On div or image
* If define image as block…
  + Mx-auto
  + Centers block
* Figure class on figure tag
* Figure-img on images
* Figure-caption on that
* Images by default are inline
  + Can make them block elements
  + Mx-auto to center them
* Figure
  + Slight differences but it looks nice

## CSS Variables

* Newer feature
* Store property values to use in css
* Pre written variables in bootstrap
* Not supported in IE
* Var()
  + :<variable name>
  + :root
    - Redefine the values
* Using the variables don’t change anything of existing bootstrap
* Cant override –danger, info…
  + These values even have the ! Important attribute
  + Which cannot override
  + –pink, --blue
    - Built in variables as a courtesy

# Mastering Layout with Bootstrap

* 3 main pieces
  + Containers
    - Snap to breakpoint
  + Fluid
  + Extra small
  + Small
  + Medium
  + Large
  + Extra Large

## Layout Grid Containers

* Responsive 12 columns
* Flexbox based
* Structure
  + Containers
  + Align content to view port or around breakpoints
  + Rows/columns
* Container
  + Center content, snap to breakpoints
  + 15px padding
  + <576
  + 576
  + 768
  + 992
  + 1200
* Container-fluid
  + Adjusts to browser
* No container will go all the way to edge with no padding
* Content always has that 15px padding from beiwport
* Sometimes u want to create container for specific elements
* If you want content to align with rest you make it the same type of container
* Rows and columns
* Prepare a row for column content
* Row should be inside container
* Right inside row you create your columns
* Content shifted over by 15px with just row
* Because you want the space to be between the columns not on the left and ride side
* Basic = col
  + Tries to fit the content is whatever rows are possible

## Columns

* Container and row prepare content for layout
* Grid is 12 columns
* Columns can span all 12 spaces
* Col(-bp)(-col)
* Can just specify col
  + And if you have more than 1 column it will auto adjust
* Column will take up full width of browser until you add a breakpoint
  + Col-sm-(num)
  + At here, column takes up how ever many columns are specified
  + But before it, column takes up full breakpoint
* Col-(num)
  + Not including breakpoint, means that column will always try to be however many of the units no matter the breakpoint size

## Multiple Columns

* Think of it like this
  + At certain view port sizes you can have more/less content inline with eachother
  + If you have 12 divs of the same content, you can have 2 side by side for a bit, and then 3 side by side when you have enough space

## Offsetting Columns

* Offset-bp-col
  + Btw 1-11
* Can be used sort of like centering the columns
* When you only have columns that add up to 8, you can offset by 2 to center them
  + Old way of centering
* Whole new way to center using “Alignment”

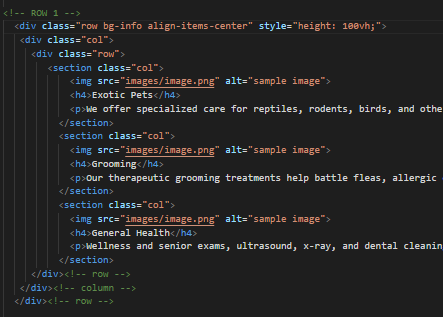
## Nesting Columns

* Columns within columns
* New row inside columns
* New 12 col grid
* You can think of “col” as the extra small column
  + The “starting” column width until other column bp’s are stated
* Insert row inside of existing column

## Customizing Order

* Order(-bp) –ord
* Can do this with columns and other elements
* If you are going to change the order, change the order of all the elements
  + This is because it works off of layers, and the layers can getmessy
* D-flex
  + Flex box
  + Default side by side
  + Flex-column
    - Makes it a column
  + Now can add order to the elements

## Alignment

* Bootstrap uses flex-box to control layout
* Vertical alignment
  + Use in rows
  + Where to align the columns content within the row
  + Align-items-<aln>
    - Start \*default
    - Center
    - End
  + Works in nested cols
    - Because we’re centering ONE column now
      * Main content is a nested row, with columns
  + 
* Individual Alignment
  + Align-self-<aln>
  + You do this inside the column itself
  + Used for the rows
* Horizontal alignment
  + Use in rows
  + Need to make sure that the contents are taking up less than the 12 columns
  + Need col width
  + Justify-content-aln
    - Start
    - Center
    - End
    - Around
      * Tries to put the spacing equidistant between the columns
    - Between
      * Pushes columns to edge and puts all the space in between

## Position/Display

* Position classes
  + Fixed-top
  + Fixed-bottom
  + Sticky-top
    - Element sticks at top of viewport as you scroll
    - Lacks support (internet explorer)
    - This can be achieved other ways too
* Basic Displays
  + Opens up bootstrap to those familiar css display properties
  + D(-bp)-typ
    - None
    - Inline
    - Inline-block
    - Block
    - Table
    - Table-row
    - Table-cell
    - Flex
    - Inline-flex
  + D-flex (allowed us to order the columns)
* Print Display
  + D-print-
  + When printing to paper
* Basic Flex Container
  + D(-bp)(-inline)-flex
  + By default flex elements are block
* D-inline-block to add width
  + Can convert any inline element to a block element =
  + Cant define width or height on inline elements
* D-flex
  + Takes that element and allows the inside elements to be effected by flex classes

## Flexbox Container Options

* Flex container
* The main idea behind the flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space, or shrinks them to prevent overflow.
* Most importantly, the flexbox layout is direction-agnostic as opposed to the regular layouts (block which is vertically-based and inline which is horizontally-based). While those work well for pages, they lack flexibility (no pun intended) to support large or complex applications (especially when it comes to orientation changing, resizing, stretching, shrinking, etc.).
  + Effects all the elements inside container
  + d(-bp)(-inline)-flex
    - d-flex (makes them flexboxes) blocks that respond to width changes
    - makes element fill the parent div
    - padding of divs change depending on the size
  + flex(-bp)(-DIR)(-reverse) \*rows are default
    - row
    - column
  + order (-bp) –ord
    - #1-12
    - View Custom Order 03\_07 for reference
  + Justify-content(-BP)-ALG
    - Horizontal alightment
    - Start
    - End
    - Center
    - Around
    - Between
  + flex(-bp)-wrp(-reverse)
    - wrap
    - nowrap
  + align-content(-bp)-alg
    - vertical alignment
    - start
    - end
    - center
    - between
    - around
    - stretch
* <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
* Individual

## Individual Flex Elements

* Align induvial elements
  + Align-self(-bp)-ALG
    - Start
    - End
    - Center
    - Baseline
    - Stretch
  + Order(-bp)-num
  + Can do the order as long in flexbox

## Floating Elements

* Standard css floating with breakpoints
* Float(-bp)-dir
  + Left
  + Right
  + Clear
* Parent container loses track of elements inside
  + Need to add clearfix to parent container
  + If an element is taller than the elemnt containing it and it is floated, it will overflow outside of the container

## Margin/Padding

* Property(SID)(-bp)-siz
* P or m
  + T
  + R
  + L
  + B
  + X (left and right)
  + Y (top and bottom)
  + 0-5, auto
* Anything you can do in css, but with breakpoints

## Visibility

* Invisible
  + Still takes up room, but cannot see it
* Visible
  + With d-none this will be useful for descriptive devices
* D(-bp)-typ
  + D –none
  + Disappear

## Sizing Utilities

* SIZ(-AMT)
  + W
  + H
  + Mw (max width)
  + Mh (max-height)
  + 25
  + 50
  + 75
  + 100 percentages
* Height is reflective of the content

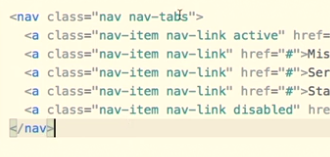
## Borders

* Border(-side)(-color)(-0)
  + Top
  + Right
  + Bottom
  + Left
  + Primary
  + Secondary
  + Success
  + Danger
  + Warning…
* Rounded(-side)(-circle)(-0)
  + 0 clears an existing border

# Nav and Navbar

* Components
  + Navs
    - Parents to all other navigation
  + Tabs
  + Pills
  + Navbars
    - Branding
    - Coor schemes
    - Dropdowns
* Completely new in bootstrap 4

## Creating Navs

* Create nav in UL
  + Or just nav tag
    - Just need anchor tag <a> with nav-item in it
    - 
  + Nav-item in link tag = li
    - If using list-item: li
    - Nav-link in anchor tag <a>
  + Navlink
    - Active
    - Disabled
  + Nav styles
    - Nav pills
      * Active link = looks like blue button with round edges
    - Nav tabs
      * Looks like tabs
  + Nav alignment
    - Justify-content-center
      * Center nav elements on page
    - Justify-content-end
      * Right align on page
    - Nav-fill
      * Have links take up entire horizontal length width of container
      * Allows some links to be bigger if content is bigger, but evrything else is same size
    - Nav-justified
      * Force space between each link the same
      * Everything is same size
    - Flex column
      * This is unique to navs, otherwise need “d-flex” for div

## Navbar

* Works with nav component
* Diff from bootstrap 3
* Navbar goes on main container
* Navbar-expand(-bp)
  + Since bootstrap is mobile first, the anchor elements will stack. Need to add this with a breakpoint for expansion
* Navbar components will stack
* Navbar colors
  + Bg-colors
  + Navbar-light
  + Navbar-dark
* Navbar-nav container
  + Main navigation goes here
  + Nav-item (if using li, this goes there)
  + Nav-link (and this always goes in anchor tag <a>
  + Active
  + Disabled
* Main nav tag
  + Navbar class, bg-class (this is background), navbar-light/dark class (this changes the text color)
* Container class to hold navbar components = navbar-nav
  + Anchor components (these will stack vertically by default)
* By default bootstrap is a mobile first framework
* Can change color by adding style property to nav tag and removing the bg color

## Navbar Brand & Text

* Navbar-brand
  + Add a logo or text
  + Link or headline
  + Can also add images
  + Navbar-text
* When using navbar brand, careful when using header tag
  + Eliminate margins with m-0
* Span navbar-text
  + Keeps the navbar style
* By default they are inline-block elements

## Dropdowns

* Dropdown requires container ->aligns the dropdown to the proper item
  + Can be a div, or list item, but if you use anchor tags you must put them in a div
    - This div needs class dropdown to align it with the element
  + Dropdown-toggle in anchor tags
    - Gives it that arrow icon
  + Data-toggle=”dropdown” attribute to a tag
  + Dropdown-menu = div within the dropdown div.
  + Each dropdown links get a dropdown-item class
    - Gives it a nice look
  + Id for accessibility and aria attributes
    - Aria for assistive devices
  + Ml-auto
    - To auto orient the navbar to the right
  + Submenus not exactly supported

## Form Elements

* To add form to menu need
  + Can for form or div elements
  + Form-inline for container
  + Form-control for input fields
  + Maybe spacing classes
    - Mr-
  + Buttons don’t need it
* Whenever new element is added to navigation it slides over to the right

## Positioning

* Fixed-top
  + Can add margin to elements so it clears
* Fixed-bottom
* Sticky-top
  + Not fully compatible

## Collapsible Content

* Thing that we want to collapse
  + Collapse class = goes on content
  + Navbar-collapse
  + Id = this connects to data-target of button
* Some element to control that content
  + Navbar-toggler
  + Create button with that class
  + Data-toggle=”collapse”
    - Usually data-toggle is javascript esc
  + Data-target = id of the content that is being collapsed
  + Navbar-toggler-icon
* Aria-controls tells screen readers what the thing is doing

# Style Element Overview

* Buttons
* Tags
* Progress bars
* List groups
  + Group list content
* Breadcrumbs

## Buttons

* Btn
* Btn-SIZ
  + sm
  + lg
* can use these classes on <a> <button> <input>
* btn-color
  + primary
  + secondary
  + success
  + danger…
* btn-outline-color
* btn-block
  + full width of container
* active
* disabled

## Button Groups

* btn-group
  + groups buttons side by side
  + roll over style
* btn-group-vertical
* btn-toolbar
* btn-group-size
  + sm
  + lg
* put these classes in a parent div

## Badges

* badge
* badge-pill
* badge-color
* they will resize according to container

## Progress Bar

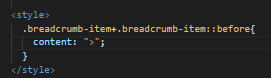
* progress class in div
* progress-bar in sub div
* width percentage to fill it, height
  + can add text
* bg-<color>
  + to color it
* progress-bar-striped
* progress-bar-animated
* copy the progress-bar sub div to add multiple colors
* role=”progressbar”
  + so that interface knows what it is

## List Groups

* iOS or Android styled button lists
* list-group container
* list-group-item
* li, button, a
* active
* disabled
* list-group-item-action style for the hover
* light-group-item-color
* can add badges
* flexbox classes

## Breadcrumbs

* indicate where you are in nav structure of side
* breadcrumb container
* breadcrumb items
* active style
* ul or ol
* li
* nav>a



# Layout Components

* certain design patterns
* jumbotron
* table styles
* card layouts

## Jumbotron

* jumbotron container
* jumbotron-fluid
  + removes the round edges
* display classes
* you use container class to align the elements to the 12 grid system
* can add image and/or background color

## Table

* table
* table-dark
* table-striped
* table-bordered
* table-hover
* head
  + thead-light
  + thead-dark
* table-color
  + active too for hover
* bg-color
  + darker
* text-color
* table-sm
* table-responsive(-bp)

## Cards

* card around main div
* card-body that contains all the info
  + card-text
  + card-title
  + card-subtitle
  + card-link
  + card-img
* bg-color
* border-color
* text-color

## Card Content

* card-header
* card-footer
* card-img
* card-img-top
* card-img-bottom
* card-img-overlay
* light-group container
* list-group-item
* list-group-flush
  + aligns the list group with the card
  + exists outside of card-body

## Card Layout

* container with cards in it
* card-group
  + side by side shared border
* card-deck
  + spacing btw cards
* card-columns
  + like pinterest
* rows & columns traditional way
  + don’t want to add column to the section

## Media Object

* images be in separate column then rest of content
* media
* div class with media-body

# Remember

* You can add multiple classes inline to an element
  + Text-center tex-sm-right
* Ctrl + shift + l
  + To select all occurences
  + Multi cursor
* Alt+click to select multiple elements
* Images by default are inline
  + Can make them block elements
  + Mx-auto to center them
* By default, bootstrap only places spaces at bottom of elements
* Using viewport height!
* Style=”height:100vh;”
  + Adjusts to the users monitor
  + Height: 50vh
  + 50% of viewheight in header tag\*\*
* Ctrl+D to select one after another
* Div.<classname>
* Div#idname
* Class=” no-gutters”
  + Removes 15 pixel space between rows
* Header when scrolling up
  + <https://www.danjford.com/blog/javascript/a-scroll-up-fixed-header/>
* When do I need “d-flex” if I can use flex-column/row without it?
* Css psudo selectors