Document Object Model (DOM)

* When you load some into browser (HTML, XML, SVG), it gets converted into DOM.
* Node – Any element tag is referred to as node.
  + <html> is a root node
  + <head> and <body> are two child node of html and siblings of each other.
* Accessing node location using Javascript
  + var node = event.target, current\_node = node.nodeName;
  + Access the parent node using node.parentNode

**Course 2**

**Front-end Web UI Frameworks:**

* What are they?
  + It provides collection of ready-to-use HTML, CSS, and JS templates for UI component.
    - Typography, forms, buttons, tables, navigation, Carousel, etc
* Why use front-end UI frameworks?
  + Responsive Web design
    - People these days use website on different devices, so we need to design a single website that is adapted all screen sizes
    - Mobile FIRST – design with mobile first design then adapt to bigger screens
  + Cross-browser compatibility
    - Dealing with quirks of browsers
    - These frameworks take care of us to deal with different browser issues
  + Increased productivity
    - Easier to get started with consistent look and feel
  + Community support
    - Larger the community that participate a framework, larger the resources are. Ex: more examples with similar issues as yours.
* List of popular front-end UI frameworks?
  + Bootstrap
  + Foundation by Zurb
  + Semantic UI
  + Pure (by Yahoo)
  + UI Kit
  + Gumby
  + Susy

**Introduction to Bootstrap**

* Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.
* Front-end framework for faster and easier web developemtn
* It has built in support for responsive design using mobile first approach
* First released in 2011 (current version 3.3.5). It’s an open source.
* It is one of the first comprehensive framework that was available to use.

**Getting Started with Bootstrap**

* 3 possible approaches
  + Download complied files from bootstrap website
  + Get hold of Bootstrap sources in SASS/Less format
  + Use precompiled bootstrap through it’s CDN (content distribution network)
* For “download complied files”
  + Your are required to have the following lines in <head> tag

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

* + Include the CSS files in the <head>

<link href=“css/bootstrap.min.css” rel=“stylesheet”>

<link href=“css/bootstrap-theme.min.css” rel=“stylesheet”>

* + - We need to put the bootstrap and bootstrap–theme css files into our project. The minified format is what we use we design our project. Reason being that minified file is much smaller in size so it’s faster to download when user is using the website
  + Bootstrap also uses JavaScript’s library (jQuery) for functionality. So include jQuery library in addition bootstrap’s own JavaScript class
    - Do this at the bottom of <body>. Reason: your webpage can be rendered even before JS is processed. It will first display all the content and reorganize using the JS. This helps page load faster.

*<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"> </script>*

*<!-- Include all compiled plugins (below), or include individual files as needed -->*

*<script src="js/bootstrap.min.js"></script>*

* Using bootstrap CDN
  + Include following in <head>

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">*

* + Include following at the bottom of <body>

*<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>*

*<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scrip>*

* Advantages of using CDN over download
  + No need to download bootstrap files to project folder
  + If user view some other webpage which also used bootstrap CDN, then CSS and JS files might already be cached in the browser which make page rendering faster.
  + Using the web page to newer version will only require the update of CDN links

**Bootstrap Container Class**

<div class= “container”>…</div>

* Way to restrict content to display in certain part of webpage. Container size will vary depending on widow size
* This is mostly used on the outermost div to wrap all the site content for **grid**  to work correctly
* .container-fluid class allow the content to displayed in full window width.

<div class= “row”>…<div>

* Divides the page into multiple rows.
* Acts as horizontal grouping
* Rows must be contained inside a container for grid to work properly

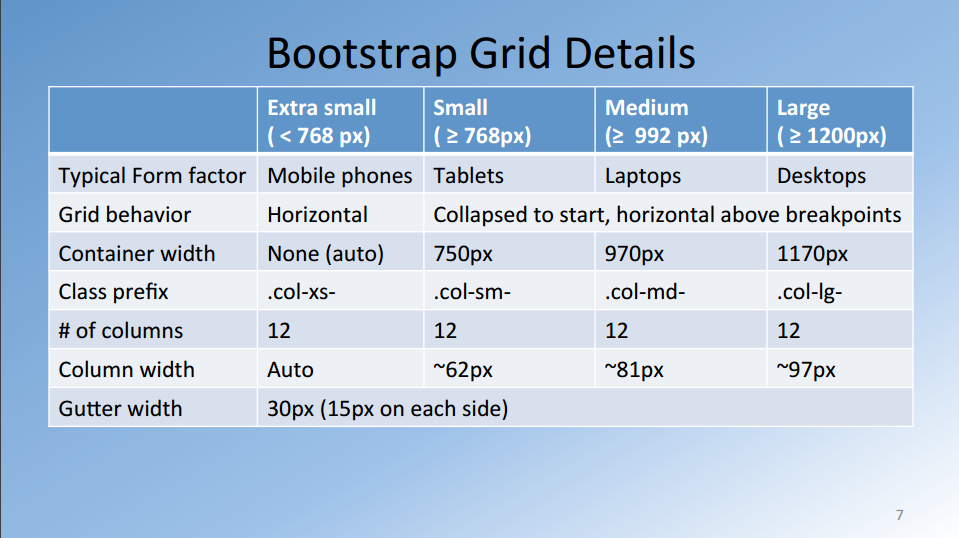
<div class=”jumbotron”>…</div>

* Lightweight (flexible) component for showcasing some key info (company name/logo)
* It can be used outside of container
  + Can be used inside a container to be displayed inside the fixed width

**Why use responsive Design?**

* User increasingly accessing websites from variety of devices of different sizes. For Example: desktop, phones, tablets, etc. It may render inappropriately on smaller size screens
* We design our application progressively that can fit all screens
* We are going to use “viewport” property of our device in order to adjust our application to the screen of our device

**Responsive Design**

* Grid system
  + <meta name="viewport" content="width=device-width, initial-scale=1”>
    - In the head we included ***viewport*** meta tag, it enables our application to identify the appropriately adapt itself to the width of the device
    - This meta allows the application width to be same as device width.
  + Bootstrap Grid is design to be:
    - Responsive
    - Mobile First
    - Fluid
  + “container” class
    - Container works responsively that fixes its’ width depending on width of the device
  + “row” class
    - It is defined in the container that takes the whole width of container
    - Bootstrap divides row into 12 equal sized columns
  + “columns”
    - We can layout one piece of content to take n number of columns and have another piece of content to take remaining columns
    - Defining columns to a piece
      * <div class= “col-sm-5”>
      * <div class= “col-sm-7”>
    - Gutter
      * Gutter is the little white space that is left between pieces that are covering all columns
  + Bootstrap makes available four classes to target different size of screens
    - xs – extra small
      * for really small screens. Ex: mobile phones
    - sm – small
      * for small screens. Bigger than mobile phones and smaller than laptop screens. Ex: tablets
    - md – medium
      * for medium screens. Ex: laptops
    - lg – large
      * for large screens. Ex: Screen bigger than 1200px in width
    - Use of classes
      * .col-xs-\*, .col-sm-\*, etc
      * \* - number of columns within a row this content is going to occupy. Cannot exceed 12 altogether within a row
    - 
  + Using Column Push and Pull
  + Column offsets
    - We can set that piece of content should push # of columns to the right
  + Nesting Columns
    - In a assigned # of columns I can insert a row which will have its own 12 columns
* Fluid Images
  + Responsive Images
* Media Queries
  + CSS technology to apply some styles based on the size of the viewport.

@media screen and (min-width: 600px) { … }

**Navigation and Navigation bar**

* Why website navigation
  + Website are rarely single pages
  + Need to provide visitors easier way to navigate around the application
  + Several commonly used navigation patterns
    - Navigation bar
    - Consider **Information Architecture** when thinking about navigation bar
      * Structure of a system with respect to the way information is:
        + Organized
        + Labeled
        + Navigation methods provided to access the info
      * <http://sixrevisions.com/usabilityaccessibility/information-architecture-101-techniques-and-best-practices/>
      * <https://webdesignfromscratch.com/website-architecture/ia-models/>
      * <http://www.steptwo.com.au/papers/kmc_whatisinfoarch/>
      * <http://www.webmonkey.com/2010/02/Information_Architecture_Tutorial/>
* Website Hierarchy
  + Home 🡪 About/Menu/Contacts 🡪 Each page’s own content
* Navigation bar is usually expected to displayed at top of application
* Dos and Don’ts
  + Dos
    - Use simple terms
    - Standardize navigation
      * No matter where the user is, they can still navigate to other pages
    - Provide indication of the location within navigation hierarchy
      * Highlight the current label that user is viewing
    - Use standard web conventions
      * Clicking logo takes you back to home page
  + Don’ts
    - Have too many items
    - Use generic labels
      * Product – what kind of product?
      * Service – what kind of service?
  + **Navigation Aids**
    - *Breadcrumbs* 
      * It is set of links indicated with pointers. Current link could be highlighted, click on link would take you back to the page where you were. It list the pages that you have visited
* Secondary navigation’
* Indicator of current page’s location within the navigational hierarchy
  + ***Path based***: set of steps
  + ***Location Based***: hierarchy
  + ***Attribute Based***: set of choices
    - Tabs
    - Pills
    - Pagination
      * When you have a lot of info that you need to display top the user so you divide it all into pages and provide some pagination support and display it either on top or bottom of the page
    - Dropdowns
    - Accordion
    - Tags
    - Scrollspy
    - Affix

**Bootstrap Navbar**

* Supports responsive behavior
  + On extra small screen you can collapse the navbar etc
  + Include button to show/collapse navbar
* class= “nav navbar-nav”
  + It will turn unordered list <ul> into unstyled and horizontal list
* class=“navbar–header” and “navbar-brand” for header of the navbar.
* Class= “navbar-right” to show links on the right side of the window
* Dropdown

**Icon Fonts**

* Set of symbols and glyphs
* Use them as regular fonts
* Can be styled anyway using CSS
* Popular lightweight replacement for simple images
  + Reduce additional requests for getting images
* Examples
  + Glyphicons – incuded as part of Bootstrap

<span class=“glyphicon glyphicon home” aria hidden=“true”></span>

* + Font Awesome

<i class=“fa fa-phon”></i>

* + Bootstrap-social
    - It’s bootstrap is a CSS class that is available to style some buttons for social media sites