**Highlights**

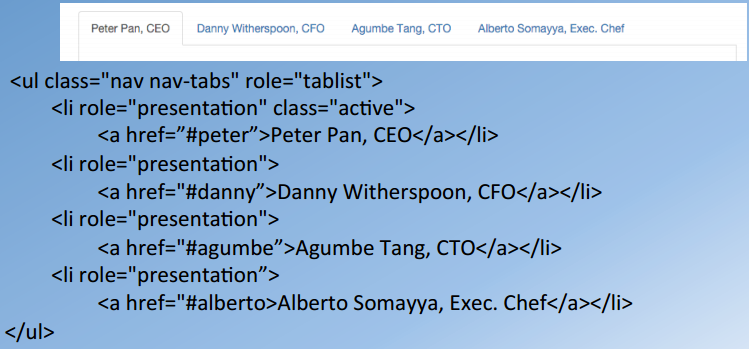
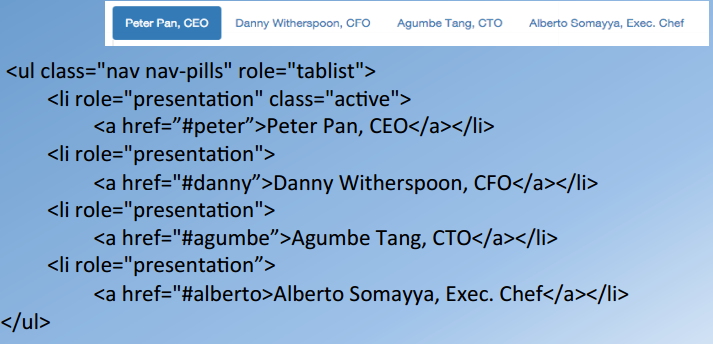
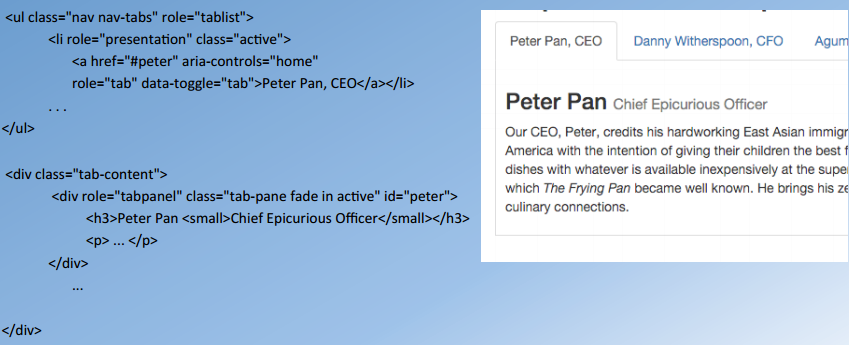
* JavaScript Components
  + Tabs and Tabbed Navigation Page 2
  + Collapse JavaScript Plugin
  + Lesson 3
    - Accordion with use of collapsev
    - Scrollspy Page 4
    - Affix
  + Lesson 4 Page 5
    - Tooltips
    - Popovers
    - Modals
  + Lesson 5 Page 7
    - Carousel

**Bootstrap JavaScript Component**

***Bootstrap’s JavaScript*** support is through JS plugins which is written based on jQuery. Hence, when attempting to include bootstrap’s js library, first include full jQuery library via CDN or including the library.

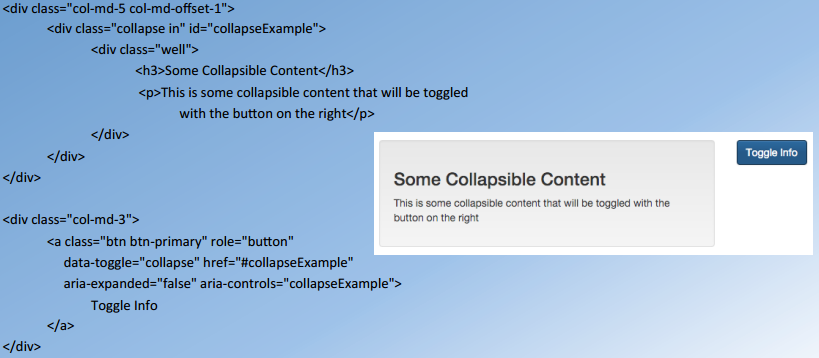
* Using Bootstrap JS Components
  + These JS components can all be used without writing single line of JavaScript
    - Use data-\* attributes
      * \* can be anything. Ex: data-toggle, data-spy
  + We need to know jQuery syntax and have knowledge of JavaScript to be able to use JS API

***Tabs and Tabbed Navigation –*** Tabs and pills allow us to organize the content.

* Typical ***Tab*** Navigation Structure
  + We use <ul> tab with “nav nav-tabs” class
  + Each list item (<li>) in the <ul> will be presented as a tab
    - Each of the tab element (<li>) carries an anchor tag <a> that has link to the tab pane with the actual content
    - In the anchor tag, include ***‘data-toggle= “tab”’*** which would be fetch data when tab is clicked
  + Example:
    - 
* Typical ***Pill*** Navigation structure
  + We use <ul> tab with “nav nav-pills” class
  + Each list item (<li>) in the <ul> will be presented as a tab
    - Each of the tab element (<li>) carries an anchor tag <a> that has link to the tab pane with the actual content
    - In the anchor tag, include ***‘data-toggle= “tab”’*** which would be fetch data when pill is clicked
  + 
* Tabs and Pills
  + Provide us the ability to navigate the content that is organized into multiple panes
  + ***“nav-justified”*** to justify tab/pill to stretch the width of parent element with each element sharing it’s proportion of the width
  + ***“disabled”*** can be used to disable the element
  + ***Pills*** can be stacked vertically using ***“nav-stacked”*** class.
* Actual content is divided into multiple panes
  + Content is enclosed in a div with ***“tab-content”*** class
    - ***“tab-pane”*** class is used to enclose each pane. All the panes are child of “tab-content” div
    - One pane is designated as ***active***
  + 

**Collapse**

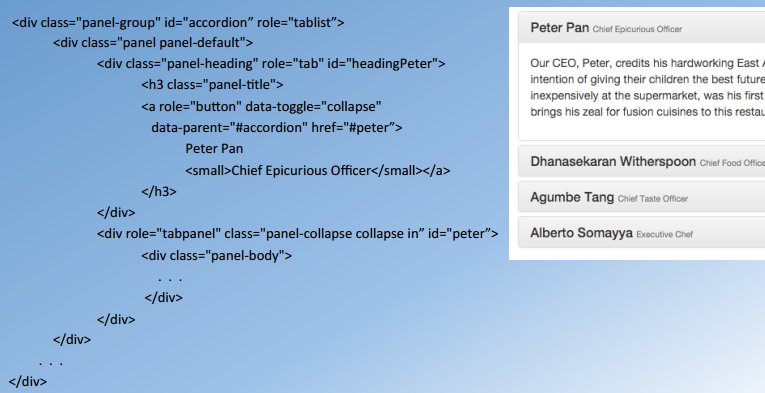
***Collapse plugin*** is used to show and hide content. It is also leveraged with the navbar so we can hide the navbar is seen on extra small screens and revealed upon clicking on a button. We’ll also make use of collapse plugin together with ***panel*** and ***panel-group*** class.

* To show and hide the content, we need a button or a link to trigger the toggle
* Collapse Example:
  + 
  + Important attributes:
    - ***collapse in*** class, ***data-toggle=“collapse”,*** and ***data-target or href***

**Accordion**

***Accordion*** is the behavior where panel and panel-group classes are combined with collapse. With accordion when panel heading is clicked it’s content is shown and previous one is get collapsed.

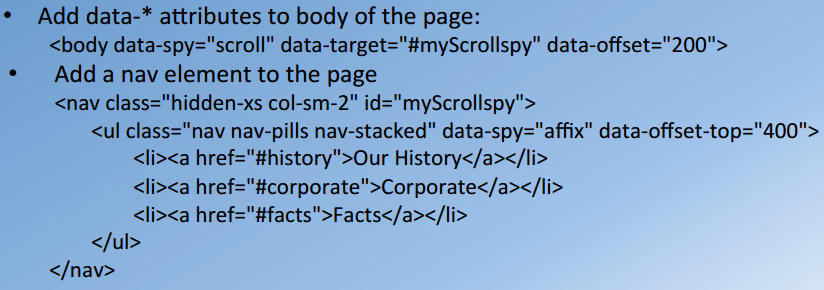
**Example:**



* Enclose the entire content inside an outer div with ***“panel-group”*** class and the ***id*** is given to the accordion (parent div)
* When <a> is created inside the panel-heading include ***data-parent=“#idofaccordion”***

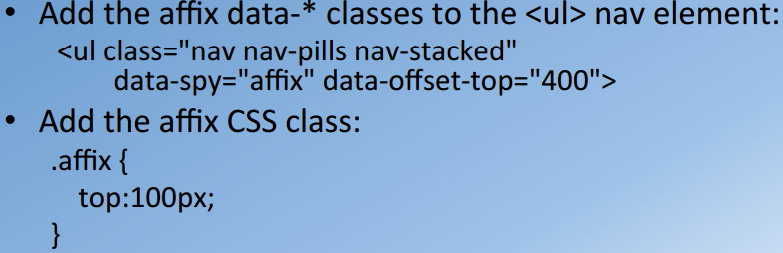
**Scrollspy and Affix plug-ins**

***Scrollspy***  is applied to nav element, so that whatever is visible in the screen will be highlighted when webpage is scrolled.



* ***data-spy=“scroll”*** enables the Scrollspy plugin. It is usually applied to body element of the webpage. ***data-target*** should be the id of the element of the navbar that we’re trying to track
* ***<nav>*** is the actual element that will be tagged. In this example we’re using nav-pills that are stacked vertically

***Affix*** is initialized when ***data-spy=“affix”*** is called as it is done in the example shown above. ***data-offset-top=“400”***means that as the scrolling happens, nav elements will scroll together with the webpage up to 400 pixels is mark from top. After that point, nav elements get fixed.

* 
* In addition in the CSS file, we can our own affix class with property ***top:100px,*** it means that nav will be fixed at 100px from top.

**Lesson 3 (Tooltips, Popovers, Modals)**

These can be used to ***overlay*** the content on top of the webpage. Order of flexibility (Tooltips 🡪 Popovers 🡪 Modals)

**Tooltips**

* Simplest way to display content
  + Pops up when the user hover over an elements
* To implement we need ***3 attributes:***
  + data-toggle=“tooltip”
  + data-placement=“top|bottom|left|right”
    - This tells where the tooltip will popup
  + title=“message string”
* Tooltip require small JS code at end of body (right after where bootstrap JS class is included)
  + This tells that wherever tooltip is added it is enabled through the following JS code

*<script>*

*$(document).ready(function() {*

*$(‘[data-toggle=“tooltip”]’).tooltip(); })*

*</script>*

**Popover**

* Includes a title and content
  + Pops up when the user clicks on an element
* To implement we need ***4 attributes:***
  + data-toggle=“popover”
  + data-placement=“top|bottom|left|right”
    - This tells where the popover will popup
  + title=“message title”
  + data-content=“message string”
* Popover require small JS code at end of body (right after where bootstrap JS class is included)
  + This tells that wherever popover is added it is enabled through the following JS code

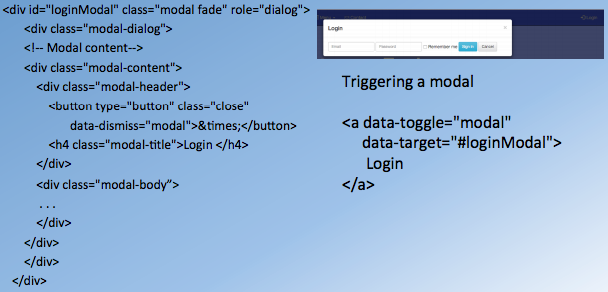
*<script>*

*$(document).ready(function() {*

*$(‘[data-toggle=“popover”]’).popover(); })*

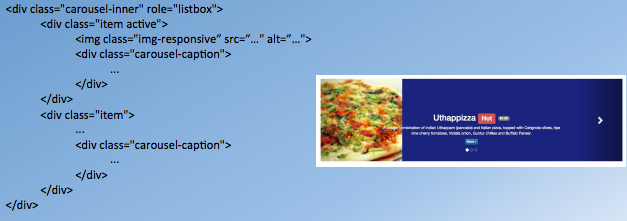
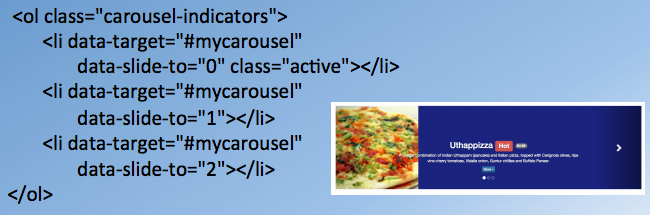
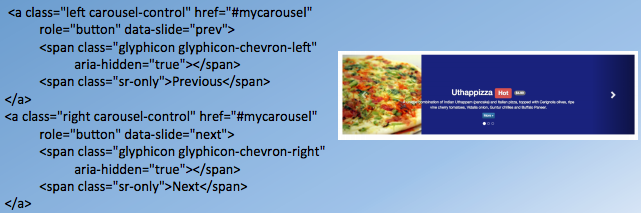
*</script>*

**Modal –** Modal allows you to define information to be represented in a lot more detail. Additionally, it allows you to format the information and specify information in the form of many HTML code.

* Modal contains header, body, and footer (if needed)
* Can use bootstrap grid layout to organize content
* 
  + parent div will have ***modal class***
  + child div will have ***modal-dialog class***
  + child div of that will have ***modal-content class***
    - Modal content element can have ***modal-header, modal-title,*** and ***modal-body***

**Carousel**

**Carousel –** it is a component that allows us to include a slideshow with caption that slides left to right. This also provides the manual control so the user can jump to any slide they want

* To add carousel to website we add a div with ***carousel class*** and ***data-ride=“carousel”*** attribute
* Adding Slides
  + 
  + element with ***class carousel-inner*** is the child outer element with ***carousel class***
    - Each slide’s element is defined ***item class***
    - Each slide can have image and caption. We can define them as shown above
    - Use ***image-responsive class*** so that image is adjusted automatically
    - ***carousel-caption class*** is to show the context
* Adding Indicator Controls
  + 
  + Above example gives manual control to go a certain slide
  + <ol> element is defined with ***carousel-indicators class***
    - <li> attribute ***data-slide-to***tells which slide to go to
    - ***data-target*** tells which id we need to refer to
* Adding Left and Right Controls
  + 
  + ***carousel-control class*** is used to show left and right arrow