CS3240 Interaction Design: Lab 02

Front-end web development

with Bootstrap and jQuery

Prepared By: Wong Zhouchun, Steve Ng, Dr Bimlesh Wadhwa, Satyam Agarwala, Low Guan Hong & Tay Yang Shun

CONTENTS

OBJECTIVE

In this lab, you would learn the following:

1. BootStrap – A UI framework for rapid development of user interface elements on your html pages.
2. jQuery – A JavaScript library that ease and simplify client-side scripting of HTML.

Note: For this lab, it would be advisable to use either Google Chrome or Firefox as you might experience some inconsistency using Internet Explorer.

Note2: For editing the html files in this guide, you are encouraged to use Brackets. However, you can use Notepad++ on Windows and TextMate or Sublime Text 2 on Mac.

INTRODUCING TWITTER'S BOOTSTRAP

Twitter's Bootstrap is an excellent set of carefully crafted user interface elements, layouts, and JavaScript tools, freely available to use in your next web design project. Together, these interface elements provide all the trappings of a standards compliant, user-friendly, professionally built HTML5 website, right out of the box.
Here are a few examples of website that has utilized Bootstrap for their user interface design:

1. http://demo.yoarts.com/start/
2. http://demo.thedevelovers.com/resume/cvilized/
3. http://demo.juhanda.net/civi/
4. https://www.quickvi.com

After this lab, you can consult with your tutor on how to view the source code of the page to have an understanding of how to do such site.

GETTING STARTED

DOWNLOAD TWITTER BOOTSTRAP

1. Go to Twitter Bootstrap's Homepage at http://getbootstrap.com/
2. Click on *Download Bootstrap* at the home page

Note: After downloading, you would see a main folder *dist* with 3 sub-folders css, fonts and js inside. We will be placing our .html files into /dist folder.

LAYOUT OF HTML PAGE

1. BOOTSTRAP GRID SYSTEM

Bootstrap grid system utilizes 12 columns (or spans). Using the editor of your choice, create a file call **layout.html.** Save the file inside /dist folder. Inside layout.html, copy the following code and run the html file in your browser.

```
<html>
    <head>
        <title>Layout</title>
        <link href="css/bootstrap.css" rel="stylesheet">
    </head>
    <body>
        <div class="container row">

            <div style="background-color:red" class="col-md-2 col-lg-2">
                Content of first column
            </div>

            <div style="background-color:green" class="col-md-4 col-lg-4">
                Content of second column
            </div>

            <div style="background-color:blue" class =" col-md-6 col-lg-6">
                Content of third column
            </div>

        </div>
    </body>
</html>
```

Note: It is always a **must** to include bootstrap.css inside your .html files in order to use Bootstrap UI elements or layout. The line to include bootstrap.css is line number 4.

Note 2: The above example embedded css inside html page. This is for demo purpose and it is not a good practice, you should always create .css for the style.

2. CENTERING YOUR HTML PAGE

In order to center your html page, replace **layout.html** line 7 with the following:

```
<div class="container row center-block">
```

3. Responsive Web Layout

Responsive web design is an approach to web design in which a site is crafted to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from desktop computer monitors to mobile phones).
Examples of site that has responsive web layout are as follows:

1. www.fork-cms.com
2. http://www.whitelotusaromatics.com/

Try viewing them through your mobile phone browser or you can try changing the size of your browser and watch how the UI element changes accordingly in the layout. Typically if you are using purely HTML/CSS, you would have to write CSS for different views which are very troublesome to do. However, if you are using Bootstrap, this is automated for you when you have set the rows and columns. As shown below is a reference for Bootstrap's grid system.

	Extra small devices <768px	Small devices ≥768px	Medium devices ≥992px	Large devices ≥1200px
Grid behavior	Horizontal always	Collapsed to start, horizontal above breakpoints		
Max container width	None (auto)	750px	970px	1170px
Class prefix	col-xs	col-sm	col-md	col-lg
# of columns	12			
Max column width	Auto	60px	78px	95px

Form Elements

1. Text Field & Text Area

A template called **forms.html** has been created, copy the .html file inside your /dist folder.
Right after *body*'s *<form role="form">*, copy the following code segment into the file and run it. You should be able to see 2 text input field and 1 text area.

```
<!-- Text field -->
<div class="form-group">
      <label for="inputName">Name</label>
      <input class="form-control" id=" inputName " placeholder="Name">
</div>

<div class="form-group">
      <label for="inputEmail">Email</label>
      <input type="email" class="form-control" id="inputEmail"
placeholder="Email Address">
</div>

<!-- Text Area -->
<div class="form-group">
      <textarea class="form-control" rows="3"></textarea>
</div>
```

Explanation
As you noticed, each element is wrapped inside * * and everything is wrapped *<form role="form">.* This is required as Bootstrap has added CSS markup to style the form for you.

Label – create a label for form elements like input field
Input – create an input field.

Note: There are still quite a number of other text field layouts that you could explore. Refer to http://getbootstrap.com/css/#forms

2. RADIO BUTTON

Continuing from **forms.html,** add in the following code after *textarea* tag.

```
<!--Radio Btn -->
<div class="form-group">
      <div class="radio">
            <label>
            <input type="radio" name="optionsRadios" id="optionsRadios1"
value="option1" checked>
            Option one
            </label>
      </div>

      <div class="radio">
            <label>
            <input type="radio" name="optionsRadios" id="optionsRadios2"
value="option2">
                  Option two
            </label>
      </div>
</div>
```

Explanation
Label – try removing the label tags and you will notice that the radio button is only selectable when you click the button but not the text.

Note: In order to be able to recognize the radio button as a group, they must have the same *name.*

3. CHECK BOX

Likewise, append the following code directly.

```
<div class="form-group">

      <div class="checkbox">
            <label>
                  <input type="checkbox" value="1">
                  Checbox 1
            </label>
      </div>

      <div class="checkbox">
            <label>
                  <input type="checkbox" value="2">
                  Checbox 2
            </label>
      </div>
</div>
```

Note: Replace checkbox or radio with checkbox-inline or radio-inline class to the checkbox or radio button for them to appear on the same line.

```
<div class="checkbox-inline">
      <label>
```

```
                <input type="checkbox" value="2">
                Checbox 2
            </label>
</div>
```

4. BUTTON

Append the following code directly as well.

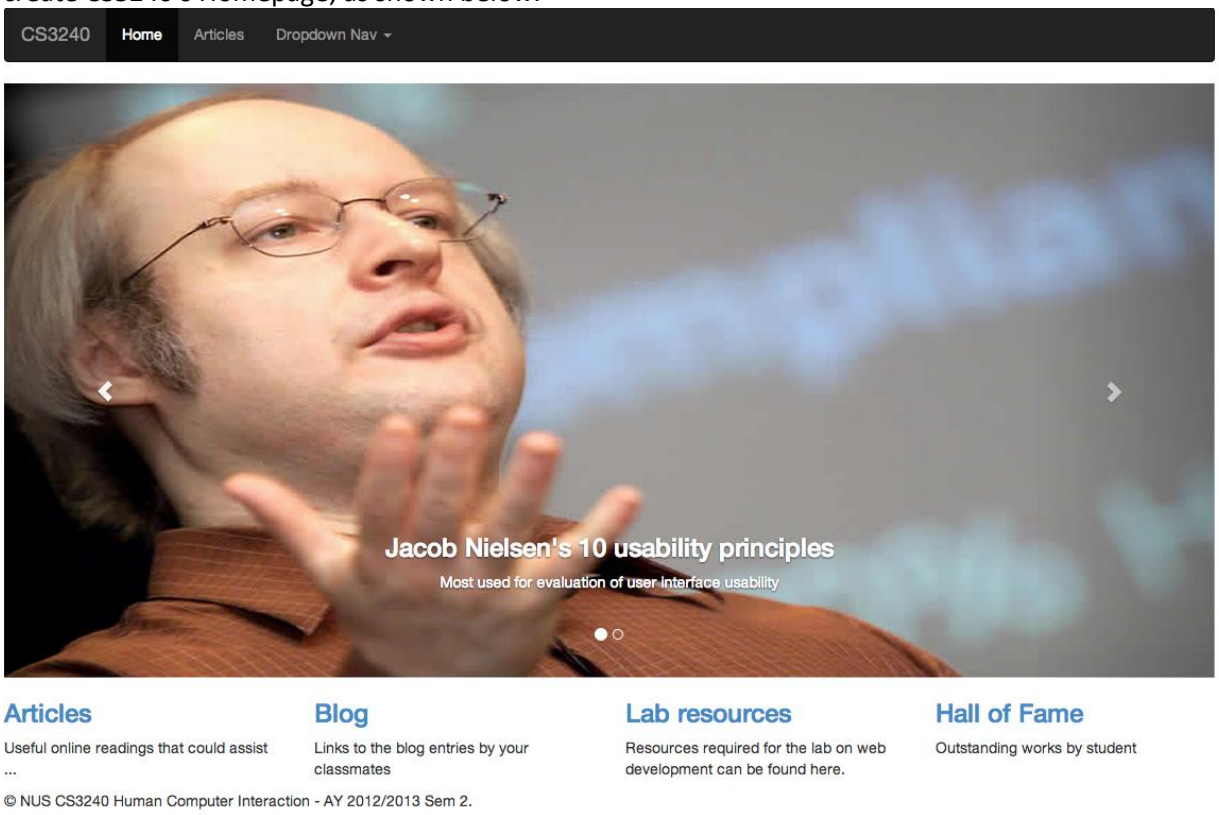
```
<!--Button -->
<button class="btn btn-primary" type="button" id="submitBtn" >Submit Form!</button>
```

Note: There are a lot of varieties of buttons - http://getbootstrap.com/css/#buttons Some of the things that can be changed is the **color** and the **size.**

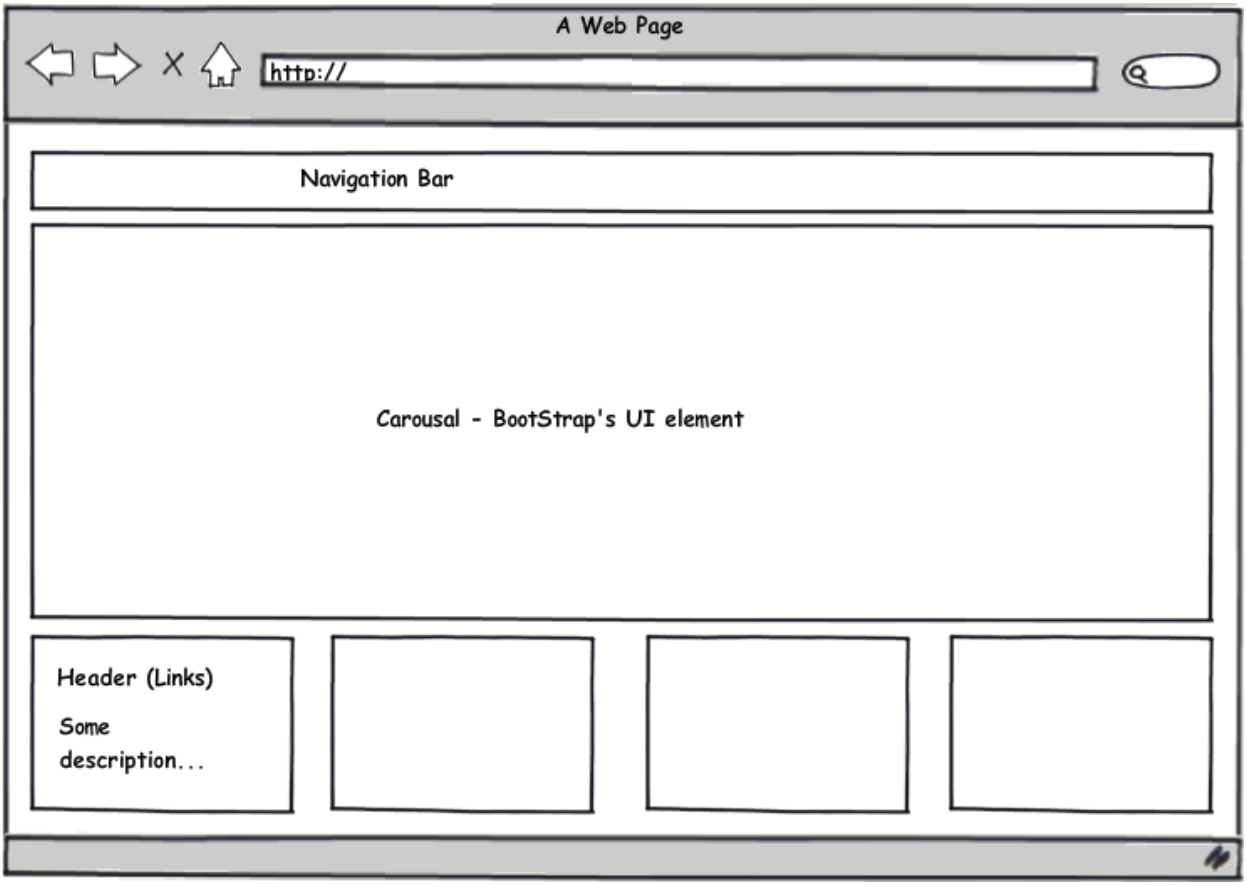
Note 2: Do not worry about the id tag for now, we will only be using it during the later part of the lab.

CREATING A WEBSITE

Now that you've familiarize with some basic elements of what Bootstrap has to offer, we will go on to create CS3240's Homepage, as shown below.



A template called index.html has been provided for you. Copy this file into your Bootstrap folder as well. Also, copy the 2 images (**Nielsen.png** and **UIParade.png**) from templates/img into your dist/img folder. These 2 images would be used later on for the **Carousal**. As shown below is the structure of the page.



1. NAVIGATION BAR

Right after the body tag, copy the following code segment.

```
<div class="container center-block">
    <nav class="navbar navbar-inverse" role="navigation">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">

        <!-- Responsive nav bar settings -->
        <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
            <a class="navbar-brand" href="#">CS3240</a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-
1">
            <!-- Nav bar content -->
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">Articles</a></li>
```

```
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle" data-
toggle="dropdown">Dropdown Nav <b class="caret"></b></a>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li><a href="#">Something else here</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </li>
            </ul>
        </div><!-- /.navbar-collapse -->
    </nav>
</div>
```

Explanation

1. We started off by wrapping everything inside *container center-block*; this will allow the navigation bar to be on the center.

2. *class = "navbar"* is the class required to invoke Bootstrap's navigation bar.

3. *Class = "navbar-header"* is meant for setting the 'brand' **CS3240** and including how the nav bar would look like when it has collapsed.

4. The code below is to create the look of how a collapsing navigation bar will look like. Try adding more lines of ** and decrease the size of the browser. Notice the top right icon will have more lines.

```
<!-- Responsive Navbar -->
<button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1">
    <span class="sr-only">Toggle navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
</button>
```

5. At *ul class="nav navbar-nav",* each <li> tag refers a link on the navigation

2. CAROUSEL

Carousel is a Bootstrap's JavaScript element that shows a slideshow of different images. An example of Carousel can be found at http://getbootstrap.com/javascript/#carousel whereby you could see that there's a slideshow that rotates between the slides

Append the following right after the previous line of code in the navigation bar section.

```
<div class="container center-block">

    <div id="carousel-example-generic" class="carousel slide" data-
ride="carousel">
        <!-- Indicators -->
        <ol class="carousel-indicators">
```

```
            <li data-target="#carousel-example-generic" data-slide-to="0"
class="active"></li>
            <li data-target="#carousel-example-generic" data-slide-
to="1"></li>
        </ol>

        <!-- Wrapper for slides -->
        <div class="carousel-inner">
            <!-- item 1 -->
            <div class="item active">
                <img src="img/Nielsen.png" alt="">
                <div class="carousel-caption">
                    <h4>Jacob Nielsen's 10 usability principles </h4>
                    <p>Most used for evaluation of user interface
usability</p></br>
                </div>
            </div>

            <!-- item 2 -->
            <div class="item">
                <img src="img/UIParade.png" alt="">
                <div class="carousel-caption">
                    <h4>UI Parade - User Interface Design Inspiration</h4>
                    <p>Ui Parade has good UI reference!.</p></br>
                    <a class="btn btn-primary" target="_blank"
href="http://www.uiparade.com/">Learn more</a>
                </div>
            </div>
        </div>

        <!-- Controls -->
        <a class="left carousel-control" href="#carousel-example-generic"
data-slide="prev">
            <span class="glyphicon glyphicon-chevron-left"></span>
        </a>
        <a class="right carousel-control" href="#carousel-example-generic"
data-slide="next">
            <span class="glyphicon glyphicon-chevron-right"></span>
        </a>
    </div>
</div>
```

Explanation

1. Likewise as the above, we wrapped everything inside a *container center-block* for centralizing the carousal.

2. Anything within carousel-inner would be elements for the carousel.

```
<div class=”carousel slide”>
      <div class=”carousel-inner”>
      </div>
</div>
```

3. Controls for the left and right button to view previous or next slide show

```
<!-- Controls -->
<a class="left carousel-control" href="#carousel-example-generic" data-
slide="prev">
      <span class="glyphicon glyphicon-chevron-left"></span>
</a>

<a class="right carousel-control" href="#carousel-example-generic" data-
slide="next">
      <span class="glyphicon glyphicon-chevron-right"></span>
</a>
```

4. Creates **one** slideshow element

```
<!-- item 1 -->
<div class="item active">
      <img src="img/Nielsen.png" alt="">
      <div class="carousel-caption">
            <h3>Jacob Nielsen's 10 usability principles </h4>
            <p>Most used for evaluation of user interface usability</p></br>
      </div>
</div>
```

3. LINKS & FOOTER

Add the following code after the last div tag.

```
<div class="container center-block">
    <!-- 4 columns of text below the carousel -->
    <div class="row">

        <div class="col-md-3 col-lg-3">
            <a href="http://www.uiparade.com/"><h3>Articles</h3></a>
            <p> Useful online readings that could assist ...</p>
        </div>

        <div class="col-md-3 col-lg-3">
            <a href="http://www.uiparade.com/"><h3>Blog</h3></a>
            <p>Links to the blog entries by your classmates</p>
        </div>

        <div class="col-md-3 col-lg-3">
            <a href="http://www.uiparade.com/"><h3>Lab resources</h3></a>
            <p>Resources required for the lab on web development can be found
here.</p>
        </div>

        <div class="col-md-3 col-lg-3">
            <a href="http://www.uiparade.com/"><h3>Hall of Fame</h3></a>
            <p>Outstanding works by student</p>
        </div>
    </div>

    <!-- FOOTER -->
    <footer>
        <p>&copy; NUS CS3240 Interaction Design - AY 2014/2015 Sem 2.</p>
    </footer>
</div>
```

Explanation

1. - informs Bootstrap that you are about to use the grid layout
2. - informs Bootstrap of the size you are occupying
3. <footer> - creates a footer element (This is a HTML tag)

SUMMARY

From the above, you've managed to try a number of UI elements offered by Bootstrap. There are still plenty that will be left for you to explore on. Moving on, if one doesn't like the UI elements of Bootstrap such as the padding or margin, you can edit the .CSS files that comes with bootstrap.

ADDITIONAL RESOURCES THAT COMPLEMENT BOOTSTRAP

1. http://www.bootstraphero.com/the-big-badass-list-of-twitter-bootstrap-resources
2. http://designshack.net/articles/css/20-awesome-resources-for-twitter-bootstrap-lovers/
3. http://www.webresourcesdepot.com/20-beautiful-resources-that-complement-twitter-bootstrap/
4. http://bootswatch.com/

INTRODUCING JQUERY

jQuery is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML. John Resig released it in January 2006 at BarCamp NYC. A team of developers led by Dave Methvin currently develops it. Used by over 55% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

For the rest of this lab, we will be doing the following

1. Effects using JQuery
2. Simple form validation

GETTING STARTED

1. Head over to http://jquery.com/download/
2. Download the latest development code (as of this writing, it is 1.10.2).
3. Copy the downloaded code (jquery-1.10.2.js) into your dist/js folder
4. Copy **jQuery.html** from the template folder into your dist folder and add the following code right before the closing body tag (</body>). This will allow you to call jQuery methods.

```
<script src="js/jquery-1.10.2.js"></script>
<script>
</script>
```

> **Note:** The jQuery file that you have downloaded from jQuery website might not be version 1.10.2 as a newer version might be out. Replace *js/jquery-1.10.2.js* to the file name you've downloaded accordingly.
>
> **Note:** During step (2), it would always be better to use minified version when you are done with developing. It is smaller and would allow user to load your website at a quicker speed.
>
> **Note2:** Next on, any codes we will be writing will be enclosed within the <script> tag.

QUICK TUTORIAL ON JQUERY

1. DOCUMENT OBJECT MODEL

The HTML DOM defines the objects and properties of all HTML elements, and the methods to access them. In short, you can view DOM elements as any HTML elements such as:

- <html>, <head>, <body>,

DOM elements have relationship between one another.
From the example below:

a. <Html> is the **root** node
b. <Head> is the first child of <html> element
c. <Body> is the 2nd child of <html>
d. The parent of <h1> is <body>

```
<html>
  <head>
    <title id =”document-title”>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```

2. ACCESSING DOM USING JQUERY

Using the same example above, there are various ways in which you can access the **title** element. Pass a selector into the $ function that matches the DOM element that you are interested in.

a. $(“#documentTitle”)
b. $(“title”)

Override the entire code inside **jQuery.html:**

```
<html>
    <head>
        <title id ="docTitle">DOM Tutorial</title>
    </head>
    <body>
        <h1>DOM Lesson one</h1>
        <p>Hello world!</p>
        <script src="js/jquery-1.10.2.js"></script>
        <script>
            var titleViaID = $('#docTitle').text();
            var titleViaType = $('title').text();

            console.log('Title Via ID ='+titleViaID);
            console.log('Title via Type ='+titleViaType);
        </script>
    </body>
</html>
```

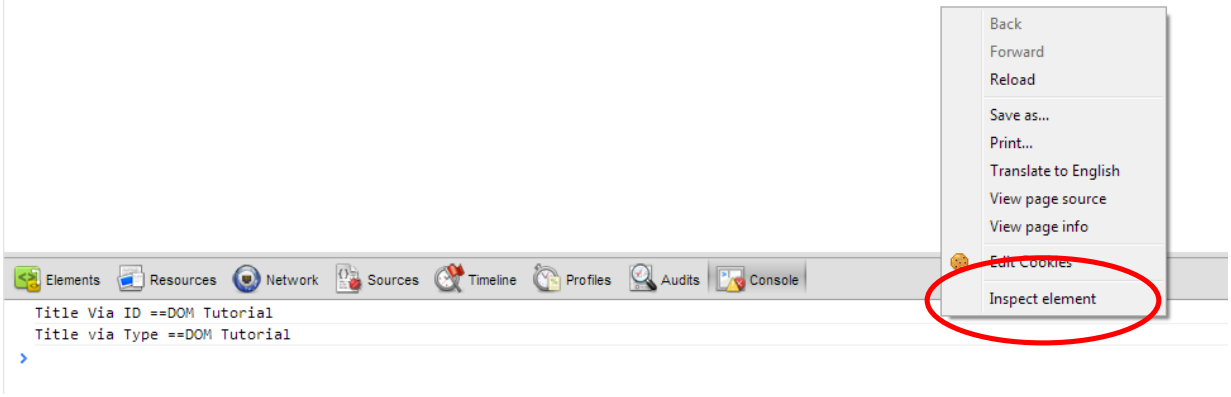
Explanation

1. .text() - Get the combined text contents of each element in the set of matched elements, including their descendants.
2. Console.log() – Prints out the string to a console

Viewing console

Google chrome

- Right click on the page and select *Inspect Element*
- Select the console option that pops out at below



Firefox

- Tools – Web Developer – Web Console
- Alternatively press CTRL+ SHIFT + K

Note: console.log() is probably going to be your next best friend in JavaScript debugging. For example whenever you are unsure of what the variable is; you can simply try console.log(variable). If you are interested in knowing more about using console.log for debugging, have a look at this article http://www.netmagazine.com/tutorials/javascript-debugging-beginners

Note2: There are still other ways to access the DOM element such as using the relationship. Refer to http://api.jquery.com/category/traversing/

FORM VALIDATION

Copy **jQueryForm.html** from the template folder into your dist folder. We will be using JavaScript and jQuery to perform simple form validation. Using JavaScript to perform form validation will save a lot of time as there is no need to submit the form to the server and back to the client if there's any error in form.

1. Include jQuery into the html file

```
<script src="js/jquery-1.10.2.js"></script>
```

2. Bind an on mouse click event to submit button. Try clicking on the submit form button and you see a popup immediately.

```
<script>
	$(document).ready(function()
		{
		    $("#submitBtn").click(function () {
		       alert("submit button clicked!");
		    });
		});
</script>
```

Note: $(document).ready(function() { }); - this means that the code within this function only be executed when the document is fully loaded and ready. This is normally a good practice as at times your DOM elements might not be loaded fully and the code already starts executing.

Note2: alert() – will create an alert pop up.

3. Validation to ensure that the name field isn't empty

Replace alert("submit button clicked!"); with the following

```
if($("#inputName").val().length==0){
	alert("Please fill in your name");
}
```

Note: This code segment will check for the length of input name, if there is no text (length = 0) an alert prompt will occur.

EFFECTS WITH JQUERY

In this section, we will continue to work on **jQueryForm.html** .

USING BUILT-IN EFFECTS

1. Add a button. Copy the code below and place in under the Submit Form! Button

```
</br></br>
<button class="btn btn-primary" type="button" id="hideBtn">Hide Submit
Button!</button>
```

2. Hiding DOM element; inside document.ready(function() insert the following code, remove the other codes inside. Try clicking on *Hide Submit Button!*

```
$(document).ready(function() {
      $("#hideBtn").click(function () {
            $('#submitBtn').hide();
      });
});
```

3. Showing the submit button when it's hidden and hiding the submit button when its visible. Copy the code below into your document.ready function

```
$("#hideBtn").click(function () {
      if($('#submitBtn').is(":visible")){
                  $('#hideBtn').text("Show Submit Btn!");
                     $('#submitBtn').hide();
      }else{
                     $('#submitBtn').show();
                     $('#hideBtn').text("Hide Submit Btn!");
                    }
});
```

Note: The above code segment checks if the submit button is hidden, and hide() or show() accordingly. The .text() method also allows you to change the text in the button.

4. To create the *Slide toggle* effect copy the following code into your document.ready function.

```
$(document).ready(function()
            {
                $("#hideBtn").click(function () {
                    $('#submitBtn').slideToggle('slow', function() {
                       //after animation is done
                    });
                });
});
```

Note: The API documentation for slideToggle can be found at http://api.jquery.com/slideToggle/

Note2: Notice the 2nd function(); this is called a callback method. Anything within this function will only be executed after animation is completed.

Summary Of Effects

1. There are still plenty of effects that you can explore from http://api.jquery.com/category/effects/
2. What we have shown here in the examples are very basic examples. If you are interested in more about the effects that you can generate, have a look at http://vandelaydesign.com/blog/web-development/jquery-animation-tutorials/

   Note: The effects shown in those tutorials would require certain level of CSS knowledge.