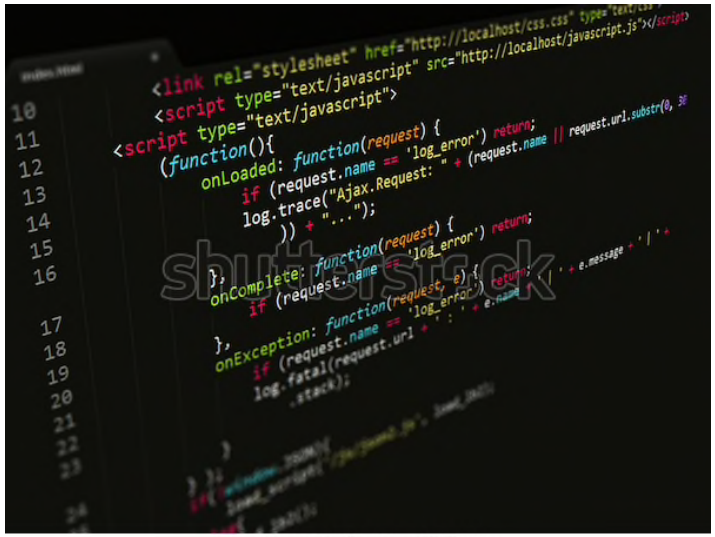
FIT5032

Internet Applications Development

Week 5: Fundamentals of Client side Javascript

ABM Russel / Murray Mount



Unit Topics

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1A	Intro to Web development and ASP.NET	Note: Studio classes commence in week 1
1B	The front end, user experience, accessibility and ASP.NET Scaffolding	
2	Introduction to C# & Version Control	
3	Entity Framework	
4	Fundamentals of Client side Javascript	Studio assessment task 1 due
5A	Validation	
5B	Security and Identity	
6	Sending Email, File Upload and Signal R	Studio assessment task 2 due
7	Web Optimisations & Evolution of ASP.NET CORE	
8A	Modern JavaScript Web Development Approaches	
8B	Testing and Deployment in Cloud	Studio assessment task 3 due
9	Review & Revision	Final Portfolio and Learning Summary due
	Examination period	
		LINK to Assessment Policy:http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-

Today

- Recap: Development Workflow
- JavaScript
- JQuery
- JQuery UI library
- Date Picker, Data Table etc.

Recap: Development Workflow

Advantages and Disadvantages of Code First

- Advantages
  - Ability to “version” control database via SVN, Git or Mercurial.
  - Reduces the amount of automatically generated code significantly.
  - A “developer” centric approach.
- Disadvantages
  - Difficult if new to the C# environment and already know SQL.
  - Heavily depends on the architecture of the systems in use, using it may be more difficult in scenarios where the database layer is important.

Advantages and Disadvantages of DB First

- Advantages
  - Assumes that the database design does not change over time.
  - Knowledge of SQL will make this task significantly easier in comparison to Code First.
- Disadvantages
  - Requires an understanding of SQL to create the Database.
  - Changes to the database can negatively impact development as auto-generated codes may not be working as intended
  - Lack the correct annotation attributes (sometimes).

JavaScript

Introduction

- JavaScript is a **lightweight**, **interpreted or JIT** compiled programming language with **first class function**.
- JavaScript designed to **run as a scripting language in a host environment,** and it is up to the host environment to provide mechanisms for communicating with the outside world. The most common host environment is the **browser**.
- JavaScript is a prototype-based, multi-paradigm, dynamic scripting language, supporting object-oriented, imperative and declarative styles.
    - **Client Side** JavaScript (JavaScript Code that is ran on the client, browser)
    - **Server Side** JavaScript (Runs on the server, for example *node.js*)

A multi paradigm language

- Programming paradigms are a way to classify programming languages based on their features. Languages can be classified into multiple paradigms.
- A multi-paradigm programming language is a programming language that **supports more than one programming paradigm.**
- JavaScript is a **multi paradigm** language.
- JavaScript supports
    - Declarative & Imperative Programming
    - Prototype based supporting object oriented

Server-side vs Client-side code

- Client-side code is code that is ran on the user's computer - when the page is viewed, the client side code is downloaded and then run and displayed by the browser. This is what we call client-side JavaScript.

- JavaScript can also be used as server-side. Recently, there is a huge surge in the popularity of using JavaScript as server side code in the Node.js environment.

JavaScript Libraries and Framework

- JavaScript library: pre-written JavaScript which allows for easier development of JavaScript applications. Examples of JavaScript libraries are
  - jQuery, jQueryUI
  - Google Maps Platform, Leaflet.js
  - d3.js
- JavaScript framework: A framework defines the entire application design. Examples of JavaScript frameworks are
  - AngularJS
  - VueJs
  - React

jQuery

jQuery

- JavaScript Library

  Make using JavaScript easier to accomplish

Default syntax:

$(*selector*).*action*()

A Selector to select the html element needed
An action that is applied to these elements

jQuery Syntax

The document ready function is used so that jQuery runs after all objects in the page are loaded.

```
$(document).ready(function()
{
  alert('Welcome to jQuery');
});
```

Alternate Syntax

Alternate syntax for document ready:

```
$(function(){

    alert('Welcome to jQuery');
});
```

What is the default jQuery syntax

What is the default jQuery syntax

A. $(action).(selector)

B. $(choice).(selector)

C. $(selector).selection()

D. $(selector).action()

E. None of the answers

The jQuery document ready function, $ (document).ready(XXXX);, has a short hand

A. <% XXXX %>

B. $(XXXX);

C. @XXXX

D. All of the answers

E. None of the answers

Using jQuery

Adding jQuery

To HTML

```
<script type='text/javascript' src='Scripts/jquery.js' />
```

To an MVC application, add to layout file:

```
@Scripts.Render("~/bundles/jquery")
```

jQuery Example

Selecting and colouring alternate lines (Match the id of the element):

```
$(document).ready(function () {
  $('#data tbody tr:even').css('background-color', 'yellow');
});
```

To select id data, e.g.:

```
<table id="data">
```

If using Style Sheets

Changing class of selected elements:

```
$(document).ready(function () {
$("#data  tbody tr:even").removeClass("data");
$('#data tbody tr:even').addClass("rowColour2");
});
```

In StyleSheet:

```
.rowColour2 { background-color: yellow; color: #666666; }
```

jQuery UI Library

Date Picker

Add the date picker scripts (after other scripts):

```
@Scripts.Render("~/Scripts/jquery-ui.js")
@Scripts.Render("~/Scripts/jquery.ui.datepicker.js")
```

Add your jQuery function (match the id):

```
$(function () {
  $("#yearPub").datepicker();
});
```

Date Picker Continue

The Date field ID is the name of the date field in the database e.g.:
yearPub

MVC automatically specifies the id of controls as being the same as their database attribute name.

Add the Stylesheet references, which are included in the jQuery UI download to the *Views/Titles/Create* View.

```
<link href="~/Content/themes/jquery.ui.all.css" rel="stylesheet"
    type="text/css" /> <link href="~/Content/themes/
    jquery.ui.datepicker.css" rel="stylesheet" type="text/css" />
```

jQuery UI Dependent Scripts

In the layout file add the relevant jQuery scripts and style sheets

```
@Styles.Render("~/Content/css")
@Scripts.Render("~/bundles/modernizr")
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/jqueryui")
@Styles.Render("~/Content/themes/base/css", "~/Content/css")
```

In an ASP.Net MVC application, where can the jQuery javascript files be included

In an ASP.Net MVC application, where can the jQuery javascript files be included

A. Included as a script file

B. Rendered as a script bundle: @Scripts.Render("~/bundles/jquery")

C. Rendered as a script file: ("~/scripts/jquery.js")

D. All of the answers

E. None of the answers

jQuery ui css styles can be added to an ASP.Net MVC application using

jQuery ui css styles can be added to an ASP.Net MVC application using

A. An HTML stylesheet link

B. Styles bundle: @Styles.Render (“~/Content/themes/base/css”)

C. A Styles Render: @Styles.Render("~/Content/JQuery-ui.css")

D. All of the answers

E. None of the answers

Data Table

Data Table

- Download from: http://datatables.net/download/
- Use (Again match the id of the element)

```
$(document).ready(function () {
  $('#data').dataTable({ "sDom": '<"nav"lf>t<"nav"i>' });
});
```

Where table id defined:

```
<table id="data">
```

Configuring Data Table

Many options available

l - Length changing
f - Filtering input
t - The Table!
i - Information
p - Pagination
r - pRocessing
< and > - div elements
<"#id" and > - div with an id
<"class" and > - div with a class
<"#id.class" and > - div with an id and class

The jQuery selector #data needs to match the

The jQuery selector #data needs to match the

A. ID of the relevant DOM object

B. Name of the relevant DOM object

C. Type of the relevant DOM object

D. All of the answers

E. None of the answers

Lecture Summary

- Recap: Development Workflow
- JavaScript
- JQuery
- JQuery UI library
- Date Picker, Data Table etc.

Week 5 Studio Overview

- Notify.js

```
38
39      @Scripts.Render("~/bundles/jquery")
40      @Scripts.Render("~/bundles/bootstrap")
41      @RenderSection("scripts", required: false)
42      <script src="~/Scripts/notify.min.js" defer></script>
43      <script>
44          $(document).ready(function () {
45              $.notify("Testing on notify JS", "success");
46          });
47      </script>
48  </body>
```

- Bootstrap datepicker

Next week: Validations

- Validations