# **Glasgow Caledonian University**

**Nick Connell**

**BSc (Hons) Computing**

**Client Side Web Development**

**Coursework**

Table of Contents

[Section 1: Folders and Data Structures 3](#_Toc501485834)

[Section 2: Testing 3](#_Toc501485835)

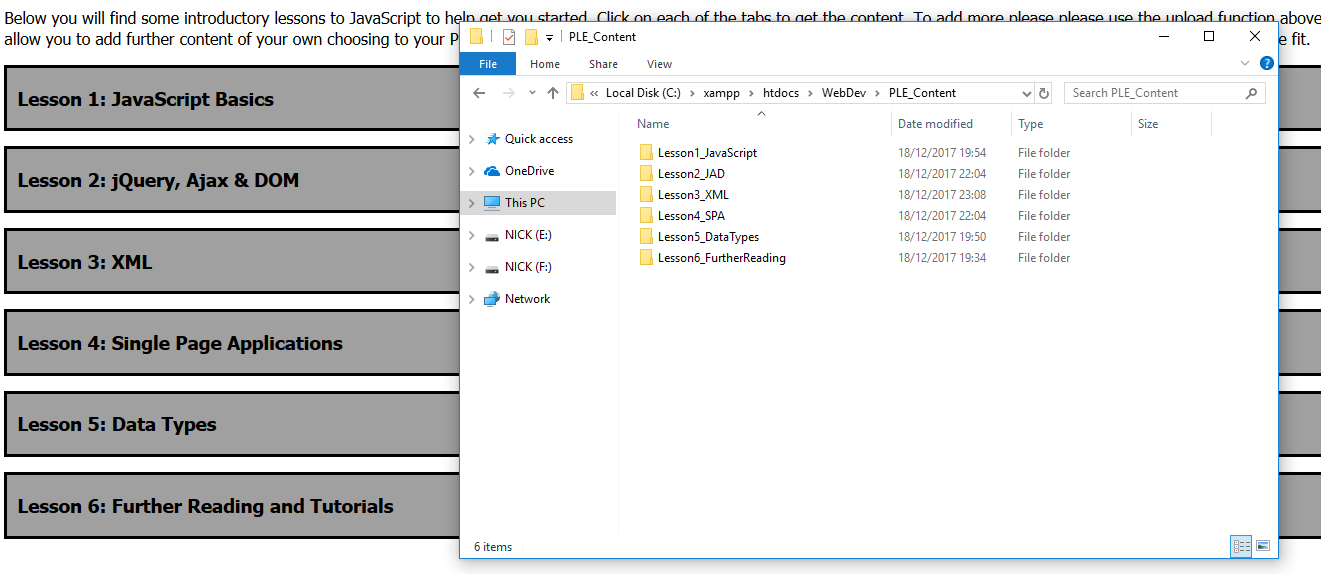
[Section 3: Reflection 4](#_Toc501485836)

[Section 4: Appendix 4](#_Toc501485837)

# Section 1: Folders and Data Structures

**File Structure**

In this project I have taken care to ensure that all data is modularised and in separate, dedicated folders to enable ease of understanding, access and modification if required. The root folder contains only 8 files. Three are the main components used to hold the site itself: two php files and a css files. There are three folders within the root: one named “php\_includes” which holds the files for the navigation bar of the website and the footer. This is advantageous to the development of the site because if changes are to be made in the future, only one file needs to be changed within this folder and this will affect all materials which refer of these files. I also have a separate folder named “scripts” which holds all my Javascript files. The last folder within the root is one I have named “PLE\_Content”. This is the main folder which holds all data which is to be used on the site. As you will see when you open the “PLE” section of the website, the folder structure of all content has been made to reflect this layout of this.



This uniform look is useful because it gives more control and organisation of what content should be stored and where. All content behind each lesson can be found in each expandable tab on the page as per the file structure. Within each lesson, I have then also separated each datatype into dedicated folders. Again, this is useful because in future if I wanted to add a new file of a specific file type to help with my learning, there will be a dedicated folder for this file to be placed in and referenced from the single page application (SPA). The generic file path to a particular file is as follows:

Root -> PLE\_Content -> Lesson -> Datatype -> File

**Meta Data**

In order for the search functionality to work for the site I have made use of some small and basic JSON code. In the metaSearch function I have hardcoded some values which are searchable within the system. These values I found through testing are case sensitive. Further improvements of the code could allow for this search to be done regardless of case. In the interests of simplicity for now at this stage I have made all searches lowercase. If the user enters “javascript” into the search bar it will expand the javascript tab below. Again, with more time and further development this could be made more user friendly – this will be discussed in Section 3 of this report.

# Section 2: Testing

Most of the testing I carried out for the site involved Unit testing. This meant testing separate and individual functions and blocks of my javascript and seeing how they acted independently. To carry out a more comprehensive and robust testing environment I would look into using the Jasmine Framework which emphasises on behaviour driven development.

On testing the site, myself it has gone through white-box testing which may involve a bit of self-bias. I know what should happen, what input is expected and how the overall site should behave in relation to this. During development, I mostly worked with expected data and in the grand scheme of things, the only area this is possible is the search function. As development progresses I could create this to be more responsive as right now if an invalid input is supplied nothing will happen at all. A warning message or alert box would be more appropriate.

A fairer approach would be to find a testing participant(s) and have them use the site. This would return a more reliable result set as other people may all think a certain section should behave in such a way different to that than the developer of the actual site.

Furthermore, if I had managed to get a login system working with options for the user to upload new datafiles, testing would have been a lot more extensive as in this section you could see how the program reacts when trying to display various different data types where not explicitly expected. Here you could work with various testing cases such as expected data, extreme data and unexpected data for example. If the upload function expected one file, what would happen if the user tried to upload two or more? Where would these be stored and could they then be dynamically displayed?

# Section 3: Reflection

**Usability**

The main objective which I wanted the webpage to hit was the simplicity in it’s use. All of the “best” websites have an extremely simple layout and functionality, Google being a notable example of this in action. Because of this, I wanted to have my site follow the “KISS” (Keep it Simple, Stupid) design principle. There is nothing worse than a webpage which isn’t instantly simple and easy to use, it can be unintuitive, counter-productive and perhaps even intimidating for the target user. As the subject and purpose of this webpage is to help educate and aide with learning, I felt it was extremely important to avoid facing any such issues. This, in itself, is the beauty of a Single Page Application (SPA) – it is by it’s very nature going to (hopefully) be simple to use as there won’t be a lot of pages of data to trawl through to consume the content. My page puts this into practice by having two pages – and with hindsight, the index page could probably be scrapped all together also. At the time of writing it contains a simple welcome message and an embedded PDF of the specification which the website had to adhere to, this was done purely for development purposes and for quick reference.

The ”Your PLE” page is very basic and hopefully apparent in regard to what is going on. There is a simple paragraph which explains what the user is expected to do. Underneath there are six expandable tabs which the user can click on to reveal the content within. These are: JavaScript Basics, jQuery-Ajax-Dom, XML, Single Page Applications, Data Types and finally a Further Reading tab (at the time of writing there is a slight CSS issue which is obscuring part of the bottom of the last tab. When it is expanded, some content is slightly hidden). I achieved this by writing a simple JavaScript file which collapses and expands the div to a certain size, hiding and showing the div through the use of an onClick operation. Within each of the tabs there is a further descriptive note to the user informing them of that the tab is representing and what their options are. Along with this there is a selection of buttons which the user can click on which will reveal yet further content – this content will make up the bulk of relevant data to help the user learn about the relevant topic in which they have opened.

At the very top of the PLE section there is a search functionality available to the user where they can search for various key words. This is done via a JSON file set within a JavaScript function. I have built this in such a way that if the user is to search for the keyword “dom”, for example, that the particular section this topic is associated will have its tab expanded. With more time I would have progressed further development on this feature to include a highlight of any references found to match the search data provided by the user. I could also have added a “Not Found” alert box in the event of a match which has no results.

**User Interface**

Following on from the KISS principle detailed in the above subsection, I have further followed this in the aesthetics of the website. There are only three contrasting colours which have been used to allow for readability and simplicity. The header and footer are black with a white typeface on top. The navigation and content tabs are a light grey with a black typeface on front and all general content areas of the website make use of a white background with black text on top. A simple font has also been used. In future developments of this particular site it would be useful to incorporate accessibility options for users with specific requirements. These could include aspects such as larger text options, different contrasting colour schemes for users with various visual impairments, and sound optionality to read the text to users with hearing disabilities.

**Software Design Patterns**

Although I didn’t fully implement a specific design pattern within this project, I have come to learn that for future reference that design patterns such as MVC will be a very effective tool to incorporate in development – particularly in projects in a professional capacity and on a much larger scale than this site. Where MVC breaks down the modelling of a website into three components - Model which manages and stores the data repository, View which displays the data, and a controller which takes charge of any interaction from the user with the website via mouse clicks and keyboard input for example – my website has loosely followed this pattern. This can be seen in that there is a folder dedicated to holding and storing all data (and subsequently different datatypes in dedicated folders) which could be likened to the Model component. Likewise, all scrips have been modularised and placed into a separate scripts folder with various files performing a particular role in the workings of the website – View – and finally the controller element of the MVC pattern; which again is held in a separate file in the main php/css files which contain all site formatting and layout of the Markup. Here user input is captured in the form of button clicks (“onClick” actions for example) and taking in user input within the search functionality.

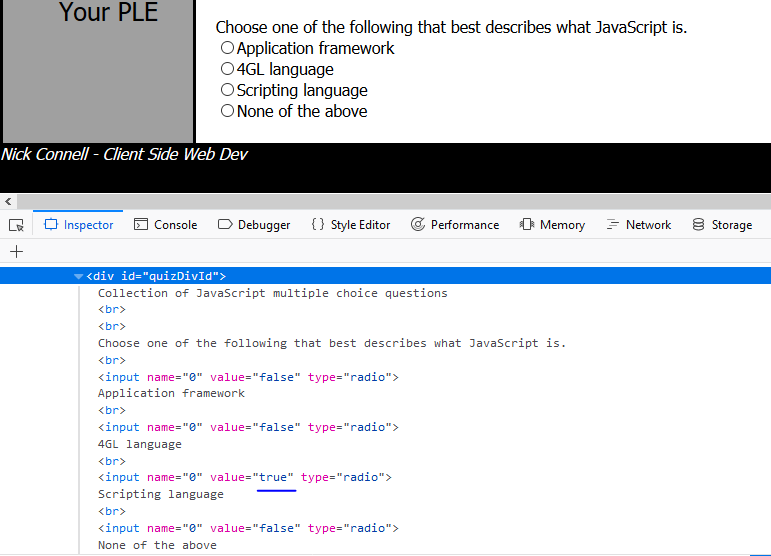
**Integrated Development Environment**

One more minor point which I personally learned throughout this project is that it is important to find an IDE which is more suitable to the job than Notepad++. A trivial point perhaps, although I have since found that JetBrains WebStorm is much nicer to use, with predictive and autocompleting syntax features, and debugging options, for example. Something to consider for any future development.

**What Went Well**

I feel that with the website at its most recent stage, the part which has been most successful and has ticked a lot of boxes, is in regard to its simplicity. Particularly the expandable and collapsible lesson tabs which can easily define, separate and display various sets of data which can allow the user to consume the content a bit easier. I believe the layout and functionality attributes to this success. This can also be attributed to the almost uniform style which I have followed for each lesson. There is a dedicated div space beneath the button options to display the data in each so they all comply to the same design styles – aiding with ease of use and familiarity as the user works through the lessons.

I also like how the XML multiple choice quiz has developed. This was an area in which I had little or no previous knowledge regarding – displaying and then formatting an xml file through JavaScript. This is probably the section I spent most time working on other than general setting up if the websites’ foundations such as the expandable tabs and the layout in the CSS. Furthermore, I enjoyed learning bit by bit how to parse and format the file by assigning radio buttons to the answer options, collating the results and displaying these via a small function which creates an output for the user to see how they performed. This was achieved by locating the answer node and obtaining the text from the file.



The scoreQuiz function then checks this Boolean value to see if the user has selected the correct answer and increments a score total if True. I enjoyed writing this piece of code because it was a new concept to me and I felt I grew into it quite quickly which gave me confidence in my own abilities. One of the biggest lessons I learned during this process was the importance of console logging certain sections of code to check for success. I found that when in a browsers’ debugging window that some of the output could be quite vague or even confusing. By adding in a simple log statement, I could check for the success, or not, of a certain function or event.

**What Could Be Improved**

The website is far from finished although I have a lot of ideas about how I could progress this in the future. I would have liked to have made more progress with a log in system which would have allowed the user to upload further files into the browser via Index DB. I would have liked to have achieved this by allowing the user to dynamically create a new expandable tab giving it a name from some user input. The user could then dynamically add files into a repository pulled from a JSON file which would allow them to create a more individualised learning experience, as the current version is quite limited in this sense. Where the user could manually go in and copy and paste some code and create the various folders quite easily to achieve this, it would be much more suitable to have this happen dynamically.

Another, smaller, area which I would liked to have improved are certain CSS issues which I encountered. Namely the part where some of the content is obscured by the footer in the last lesson tab. As this was a smaller issue I gave it less priority over some of the most key features which I wanted the site to have working.

Furthermore, I would quite have liked to have added some video content for each section. I personally find videos a more intuitive way to learn as I find more abstract concepts can be conveyed a bit easier through hearing someone speak about the topic rather than reading a text based tutorial. This is, of course, completely subjective.

# Section 4: Appendix

**Source Code:**

Index

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<!--JAVASCRIPT-->

<script src="scripts/jsModules.js"></script>

<title>Index</title>

<!--STYLESHEET-->

<link rel="stylesheet" type="text/css" media="screen" href="stylesheet.css">

</head>

<body>

<div id="Container">

<!--HEADER-->

<div id="header">

<h1>Personalised Learning Environment</h1>

</div>

<!--NAVIGATION-->

<?php

include "php\_Includes/navigation.php";

?>

<!--CONTENT CONTAINER-->

<div id="ContentContainer">

<p></p>

<h3>Welcome to PLE!</h3>

<p>Please view specification below. Use the navigation bar to your left to enter the "Your PLE" section of the website.</p>

<embed src="coursework.pdf" width="100%;" height="1200px;" type='application/pdf'>

<p></p>

<div id="specSpace"> <!--Coursework Specifications to print here-->

</div>

</br>

</div>

<!--FOOTER-->

<?php

include "php\_Includes/footer.php";

?>

</div><!--END OF MAIN CONTAINER DIV-->

</body>

</html>

PLE

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device=width, initial-scale= 1.0">

<meta name="keywords" content="JavaScript">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Search</title>

<!--STYLESHEET-->

<link rel="stylesheet" type="text/css" media="screen" href="stylesheet.css">

<!--SCRIPTS-->

<script src="scripts/jquery-1.12.4.min.js"></script>

<script src="scripts/jsModules.js"></script>

<script src="scripts/collapse.js"></script>

<script src="http://code.jquery.com/jquery-3.2.1.js"

integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jcIzLy3oi35EouyE="

crossorigin="anonymous"></script>

</head>

<body>

<div id="Container">

<!--HEADER-->

<div id="header">

<h1>Your Personalised Learning Environment</h1>

</div>

<!--NAVIGATION-->

<?php

include "php\_Includes/navigation.php";

?>

<!--CONTENT CONTAINER-->

<div id="ContentContainer">

<!--SEARCH BAR DIV-->

<div id = "SearchBar" style="width: 1100px">

<!-- <form id="metaSearch" name="search"> -->

<p>Please use the search bar to find specific topics or use the button below to upload new content to your PLE</p>

<input id="metaSearch" type="text" placeholder="Keywords: eg. 'HTML5'" name="search">

<button onClick="javascript:metaSearch()">Search</button>

<br><br>

<button onclick="newTopic()">Add New Lesson</button> or

<button>Modify</button>

<!-- </form> -->

</div><!--END OF SEARCH BAR DIV-->

<p>Below you will find some introductory lessons to JavaScript to help get you started. Click on each of the tabs to get the content. To add more please please use the upload function above

which will allow you to add further content of your own choosing to your Personal Learning Environment, or or edit anything please use the modify button to edit or remove content as you see fit.</p>

<p></p>

<!-- LESSON 1: JAVASCRIPT -->

<div id="toggler" onclick="toggleText('toggleJS');"> <h3> Lesson 1: JavaScript Basics </h3></div>

<div id="toggleJS" class="togglerContent">

<p>The following sections will introduce you to the basics of JavaScript and discuss some of the main features of the language.</p>

<button onclick="javascriptOverview()">Overview</button>

<button onclick="javascriptFurtherReading()">Further Reading</button>

<p></p>

<div id="Lesson1\_displayContent">

Click the various buttons above to view content and material about JavaScript!

<div id="imageHere">

</div>

</div>

</div>

<!-- LESSON 2: JQUERY AJAX + DOM -->

<div id="toggler" onClick="toggleText('toggleJAD');"> <h3> Lesson 2: jQuery, Ajax & DOM </h3></div>

<div id="toggleJAD" class="togglerContent">

<p>The following section will look into JQuery, Ajax & DOM.</p>

<button onclick="jQueryF()">jQuery</button>

<button onclick="Ajax()">Ajax</button>

<button onclick="DOM()">DOM</button>

<button onclick="nicktest()">JSON</button>

<p></p>

<div id="Lesson2\_displayContent">

Click the various buttons above to view content and material about jQuery, Ajax and the DOM!

<div id="pdfData" style="display: none;">

<button class="prev2">Previous</button>

<div></div>

<button class="next2">Next</button>

</div>

</div>

</div>

<!-- LESSON 3: XML -->

<div id="toggler" onClick="toggleText('toggleXML');"> <h3> Lesson 3: XML </h3></div>

<div id="toggleXML" class="togglerContent">

<p>The following section will look into XML.</p>

<button onclick="xmlOverview()">Overview</button>

<button onclick="xmlFurtherReading()">Further Reading</button>

<button onclick="xmlTut()">XML + JSON Tutorial</button>

<button onclick="xmlMultiChoice()">Multiple Choice Quiz</button>

<p></p>

<div id="Lesson3\_displayContent">

Click the various buttons above to view content and material about XML!

</div>

<h4>XML + JSON Tutorial</h4>

<textarea id="txtArea">

Click the XML + JSON Tutorial button to learn more here!

</textarea>

</div>

<!-- LESSON 4: SINGLE PAGE APPLICATION -->

<div id="toggler" onClick="toggleText('toggleSAP');"> <h3> Lesson 4: Single Page Applications </h3></div>

<div id="toggleSAP" class="togglerContent">

<p>The following section will look into Single Page Applications.</p>

<button onclick="spaOverview()">Overview</button>

<button onclick="spaFurtherReading()">Further Reading</button>

<p></p>

<div id="Lesson4\_displayContent">

Click the various buttons above to view content and material about SPA's!

</div>

<div id="slideshow" class="container">

<h4>Single Page Application Lecture Slides</h4>

<div class="slider-outer">

<button class="prev" alt="Prev">Previous</button>

<div class="slider-inner">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide1.jpg" class="active">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide2.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide3.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide4.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide5.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide6.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide7.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide8.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide9.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide10.jpg">

<img src="PLE\_Content/Lesson4\_SPA/jpg/slides/Slide11.jpg">

</div> <!-- end of slider outer -->

<button class="next" alt="Next">Next</button>

</div> <!-- end of slider outer -->

</div>

</div>

<!-- LESSON 1: DATATYPES -->

<div id="toggler" onClick="toggleText('toggleDataTypes');"> <h3> Lesson 5: Data Types </h3></div>

<div id="toggleDataTypes" class="togglerContent">

<p>The following section will look into Data Types.</p>

<button onclick="datatypeOverview()">Overview</button>

<button onclick="datatypeFurtherReading()">Further Reading</button>

<p></p>

<div id="Lesson5\_displayContent">

Click the various buttons above to view content and material about Data Types!

</div>

</div>

<!-- LESSON 6: FURTHER READING -->

<div id="toggler" onClick="toggleText('toggleTuts');"> <h3> Lesson 6: Further Reading and Tutorials </h3></div>

<div id="toggleTuts" class="togglerContent">

<p>A list of further learning resources and tutorials to help with learning:</p>

<p><a href="https://developer.mozilla.org/bm/docs/Web/JavaScript">Official Mozilla Documentation</a></p>

<p><a href="https://www.tutorialspoint.com/javascript/">Tutorials Point: JavaScript</a></p>

<p><a href="https://www.codecademy.com/learn/introduction-to-javascript">Code Academy</a></p>

<p><a href="https://www.youtube.com/watch?v=vZBCTc9zHtI">YouTube Totorial</a></p>

<p><a href="https://stackoverflow.com/tags/javascript/info">Stack Overflow</a></p>

<p><a href="https://www.w3.org/XML/">XML Documentation</a></p>

<p><a href="https://www.w3schools.com/html/">w3schools tutorial</a></p>

<p><a href="https://jquery.com/">jQuery</a></p>

<p></p>

</br>

</div>

</div>

</div>

</div><!--END OF CONTENT CONTAINER-->

<!--FOOTER-->

<?php

include "php\_Includes/footer.php";

?>

</div><!--END OF MAIN CONTAINER DIV-->

</body>

</html>

CSS

body {

margin:0;

padding:0;

height:100%;

}

/\* PARENT CONTAINER \*/

#Container {

height:100%;

position:relative;

}

/\* HEADER \*/

#header {

background: black;

color: white;

height: 120px;

text-align: center;

position:fixed;

width: 100%;

}

/\* CONTENT CONTAINER FOR ALL DATA \*/

#ContentContainer {

height:2000px;

width: 1400px;

padding-left: 13%;

padding-top: 120px;

}

#ContentContainer toggler{

text-align: center;

padding-right: 15px;

width: 1350px;

}

/\* FOOTER \*/

#footer {

position:fixed;

left:0px;

bottom:0px;

height:50px;

width:100%;

background: black;

color: white;

text-align: centre;

font-style: italic;

}

/\* FOR SEARCH.PHP - TRADITIONAL WORD SEARCH DIV \*/

#SearchBar{

border-style: solid;

border-width:medium;

height: 140px;

width: 220px;

display: inline-block;

margin: 15px;

margin-right: 15px;

padding-left: 10px;

padding-right: 300px;

}

#resultsTable{

width: 1350px;

padding-bottom: 60px;

}

/\* TEXT APPEARANCE \*/

#header, #footer, #ContentContainer {

font-family: Tahoma, Geneva, sans-serif;

}

/\* LAYOUT OF CONTACT US PAGE \*/

/\* HEIGHT OF QUESTION BOX \*/

#question {

height:200px;

}

#questionLabel{

position: top;

}

/\* ALL NAVIGATION STYLING IN HERE \*/

body {

margin: 0;

}

nav {

list-style:none;

list-style-type: none;

margin: 0;

padding: 0;

width: 12%;

background-color: #A0A0A0;

position: fixed;

height:100%;

overflow: auto;

text-align: left;

font-style: bold;

font-size: 25px;

top: 120px;

border-style: solid;

border-width:medium;

font-family: Tahoma, Geneva, sans-serif;

}

nav ul {

list-style: none;

text-align: centre;

}

li a {

display: block;

color: #000;

padding: 8px 16px;

text-decoration: none;

}

#txtArea{

height: 500px;

width: 1200px;

}

/\* STYLING FOR SPA SLIDESHOW \*/

.container{

width: 800px;

margin: 40px auto;

overflow: auto;

}

.slider-inner{

width: 500px;

height: 300px;

position: relative;

overflow: hidden;

float: left;

padding: 3px;

border: #666 solid 1px;

}

.slider-inner img{

display: none;

width: 500px;

height: 300px;

}

.slider-inner img.active{

display: inline-block;

}

.prev, .next{

float: left;

margin-top: 130px;

cursor: pointer;

}

.prev{

position: relative;

margin-right: -45px;

z-index: 100px;

}

.next{

position: relative;

margin-left: -45px;

z-index: 100px;

}

/\* STYLING FOR TOGGLE CONTAINERS \*/

#toggler {

text-align: left;

background-color: #A0A0A0;

border-style: solid;

border-width: medium;

margin-top: 15px;

width: 1350px;

padding-left: 10px

}

.togglerContent {

display: none;

text-align: left;

width: 100%;

padding-left: 10px

}

/\* FORMATTING OF SEARCH RESULTS \*/

table, td, th {

border: 1px solid black;

margin-bottom: 10px;

width: 1350px;

padding-left: 10px;

}

/\* ALL MEDIA QUERIES HERE \*/

@media screen and (min-width: 320px) {

#container {width: 100%;}

#header {width: 100%; height: 120px;}

#ContentContainer {width: 85%; height: 100px; padding-top: 120px;}

#footer {width: 100%;}

#SearchBar {width: 100%; width: 100%; padding-left: 10px;}

#toggler {width: 100%;)

.togglerContent {width: 100%;}

nav {width: 15%; height: 100%; text-align: center;}

}

jsModules

/\* ######################### FUNCTIONS FOR INDEX ######################### \*/

function loadSpec() {

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("specSpace").innerHTML = this.responseText;

}

};

xhttp.open("GET", "ajax\_info.txt", true);

xhttp.send();

}

/\* ######################### FUNCTIONS FOR PLE ######################### \*/

/\* -------------------- LESSON 1: JAVASCRIPT BASIC SECTION -------------------- \*/

function javascriptOverview(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson1\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson1\_Javascript/txt/Lesson1\_JavaScriptBasics.txt", true);

xhttp.send();

}

function javascriptFurtherReading(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson1\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson1\_Javascript/html/Lesson1\_JavaScriptBasics.html", true);

xhttp.send();

}

function javascriptHelloWorld(){

var img = $("#testtt").append("<img />").attr('src', "Slide1.jpg").on('load', function() {

console.log('Test to get this')

if(!this.complete || typeof this.naturalWidth == "undefined" || this.naturalWidth == 0) {

alert('broken image!');

}else{

$("#imageHere").append(img);

}

});

}

/\* -------------------- LESSON 2: JQUERY + AJAX + DOM SECTION -------------------- \*/

function jQueryF(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson2\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson2\_JAD/html/Lesson2\_jquery.html", true);

xhttp.send();

}

function Ajax(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson2\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson2\_JAD/html/Lesson2\_ajax.html", true);

xhttp.send();

}

function DOM(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson2\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson2\_JAD/html/Lesson2\_dom.html", true);

xhttp.send();

}

function nicktest(){

console.log("hi");

}

/\* -------------------- LESSON 3: XML SECTION -------------------- \*/

function xmlOverview(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson3\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson3\_XML/html/Lesson4\_xml.html", true);

xhttp.send();

}

function xmlFurtherReading(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson3\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson3\_XML/html/Lesson4\_xml\_furtherReading.html", true);

xhttp.send();

}

function xmlTut(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("txtArea").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson3\_XML/xmlJson.tut", true);

xhttp.send();

}

function xmlMultiChoice(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

//console.log(this.responseText);

var xml = this.responseText,

xmlDoc = $.parseXML(xml),

$xml = $(xmlDoc),

$description = $xml.find("description");

//console.log($xml)

var quizDiv = $("<div id='quizDivId'>");

quizDiv.append( $description.text()).append("<br>").append("<br>");

$xml.find('mcq').each(function(questionNumber) {

quizDiv.append(escapeHtml($(this).find("question").text())).append("<br>");

//quizDiv.append("<ol>");

var answerPosition = $(this).find("answer").text();

$(this).find("option").each(function(optionIndex) {

var optionValue = false;

if (answerPosition == (optionIndex + 1)) {

optionValue = true;

}

//console.log(optionValue);

quizDiv.append("<input type='radio' name='" +questionNumber+"' value='" + optionValue + "'>" + $(this).text() + "<br>");

//console.log($(this).text());

//console.log("nnnn", questionNumber)

// Console logging used for debugging purposes

});

quizDiv.append("<br>");

});

quizDiv.append("<button type='button' onclick='javascript:scoreQuiz();'>Score</button>");

$( "#Lesson3\_displayContent" ).append(quizDiv);

}

};

xhttp.responseType = "xml";

xhttp.open("GET", "PLE\_Content/Lesson3\_XML/xml/mcqs.xml", true);

xhttp.send();

}

function scoreQuiz() {

// result text to display with place holders,

var resultText = "You answered <b>{numAnswered}</b> questions and got <b>{numACorrect}</b> correct.";

// use jquery to get number of radio button checked

var numAnswered = $( "input:radio" ).filter(":checked").length;

var numCorrect = 0;

// for each checked radio button, get the value and count which are correct.

$( "input:radio" ).filter(":checked").each(function(i, item) {

if (item.value == "true") {

console.log("in", item.value)

numCorrect++;

}

});

// Update the results text and display

$("#quizDivId").append("<br>");

$("#quizDivId").append(resultText.replace("{numAnswered}", numAnswered).replace("{numACorrect}", numCorrect));

}

// Forces to ignore string literals found within question 3 of XML quiz

// This will allow to display as normal text

function escapeHtml(unsafe) {

return unsafe

.replace(/&/g, "&amp;")

.replace(/</g, "&lt;")

.replace(/>/g, "&gt;")

.replace(/"/g, "&quot;");

}

/\* -------------------- LESSON 4: SINGLE PAGE APPLICATION SECTION -------------------- \*/

function spaOverview(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson4\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson4\_SPA/html/Lesson4\_spa\_overview.html", true);

xhttp.send();

}

function spaFurtherReading(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson4\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson4\_SPA/html/Lesson4\_spa\_furtherReading.html", true);

xhttp.send();

}

/\* -------------------- SPA SLIDESHOW -------------------- \*/

$(document).ready(function(){

$('.next').on('click', function(){

console.log("clicked");

var currentImg = $('.active');

var nextImg = currentImg.next();

if(nextImg.length){

currentImg.removeClass('active').css('z-index', -10);

nextImg.addClass('active').css('z-index', 10);

}

});

$('.prev').on('click', function(){

var currentImg = $('.active');

var prevImg = currentImg.prev();

if(prevImg.length){

currentImg.removeClass('active').css('z-index', -10);

prevImg.addClass('active').css('z-index', 10);

}

});

});

/\* -------------------- LESSON 5: DATA TYPE SECTION -------------------- \*/

function datatypeOverview(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson5\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson5\_DataTypes/html/Lesson5\_datatypes\_overview.html", true);

xhttp.send();

}

function datatypeFurtherReading(){

var xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

document.getElementById("Lesson5\_displayContent").innerHTML = this.responseText;

}

};

xhttp.open("GET", "PLE\_Content/Lesson5\_DataTypes/html/Lesson5\_datatypes\_furtherReading.html", true);

xhttp.send();

}

/\* -------------------- MISCELANIOUS -------------------- \*/

function metaSearch(){

// This will return one object per section

var spaResourceMetaInfo =

[

// Further tags could be added to return a larger set of

// searchable results within the site

{ id:1,

tags:['js', 'javascript'],

togglerId:'toggleJS'

},

{ id:2,

tags:['jquery', 'ajax', 'dom'],

togglerId:'toggleJAD'

},

{ id:3,

tags:['xml'],

togglerId:'toggleXML'

},

{ id:4,

tags:['html', 'spa'],

togglerId:'toggleSAP'

},

{ id:5,

tags:['data'],

togglerId:'toggleDataTypes'

},

{ id:6,

tags:['tut'],

togglerId:'toggleTuts'

}

];

searchTerm = document.getElementById("metaSearch").value.toLowerCase(); // Case sensitive

console.log("searchterm", searchTerm);

for (var i=0; i<spaResourceMetaInfo.length; i++) {

if(spaResourceMetaInfo[i].tags.indexOf(searchTerm) > -1) {

// If search term matches it will expand the relevant tab.

// Clicking again will close the tab - something which could be improved

toggleText(spaResourceMetaInfo[i].togglerId);

break;

}

}

}

// Function to create a new topic - Not relevant for now

function newTopic(){

// Capture user input

var newTopicName = prompt("New Topic:", "New Topic e.g. 'HTML5'");

document.getElementById("NewTopic0").innerHTML = newTopicName;

var but = document.createElement("button");

$("#NewTopic").append(but);

$("#NewTopic").append("<div>Add your new content here</div>")

$("#NewTopic").append("<button >Upload Content</button>")

$("#NewTopic").append("<p id='NewData'>Your content</p>")

$("#NewTopic").append("<p>Your content</p>")

$("#NewTopic").append("<p></p>")

$("#NewTopic").append("<p></p>")

}