|  |  |
| --- | --- |
| **Front End Development**  Diploma in IT, CSF  2019/20 Semester 2 | Week **14** |
| 4 Hours |
| **Introduction to AJAX** | |

In this practical you will learn to use AJAX to:

* Update a web page without reloading the page
* Request data from a server - after the page has loaded
* Receive data from a server - after the page has loaded
* Read XML data into a table

**Activities**

1. Read the following resources:
   * What is Ajax? (Textbook p.633)
   * jQuery and Other Libraries (Textbook pp.633 – 637)
   * AJAX Introduction from <https://www.w3schools.com/xml/ajax_intro.asp>
   * AJAX for Front-End Designers from

<https://webdesign.tutsplus.com/series/ajax-for-front-end-designers--cms-967>

1. Follow the instructions below to complete the tasks.

If you have any question, please contact your tutor via Microsoft Teams.

**References:**

* + AJAX Introduction from <https://www.w3schools.com/xml/ajax_intro.asp>
  + An Introduction to AJAX for Front-End Designers from

https://webdesign.tutsplus.com/tutorials/an-introduction-to-ajax-for-front-end-designers--cms-25099

**Task 1: Display information from a server using onclick() event**

1. Create a new project from Blank Solution template in Visual Studio and name it as **Week14Practical**.
2. Change to the Folder View.
3. Add folders **css**, **js** and **txt** into the project.
4. Add a new HTML page and name it as **ajax**.
5. In the ajax.html file,
   1. Add meta tag for author and your name as content.
   2. Add meta tag for description and “Week 14 Practical – AJAX Demo” as content.
   3. Add the title “AJAX Demo”.
   4. Add the following to the <body> element of ajax.html.

<section id="demo">

     <h1>Let AJAX change this text</h1>

     <button type="button" onclick="loadDoc()">Change Content</button>

</section>

This creates a button that calls a JS function loadDoc() when the button is clicked.

1. Add a new JavaScript File to the **js** folder and name it as **ajax**.
2. Copy the following **loadDoc()** function from w3schools to **ajax.js**.

function loadDoc() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
     document.getElementById("demo").innerHTML = this.responseText;  
    }  
  };  
  xhttp.open("GET", "ajax\_info.txt", true);  
  xhttp.send();  
}

Refer to <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/readyState> for details of readyState values.

Refer to <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status> for values of status.

1. Link **ajax.js** to **ajax.html**.

*Refer to XMLHttpRequest Object Properties from* [*https://www.w3schools.com/xml/ajax\_xmlhttprequest\_create.asp*](https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp)

1. Add a new Text File into **txt** folder and name it as **ajax\_info**.
2. Copy the following contents to ajax\_info.txt.

<h1>AJAX</h1>

<p>AJAX is not a programming language.</p>

<p>AJAX is a technique for accessing web servers from a web page.</p>

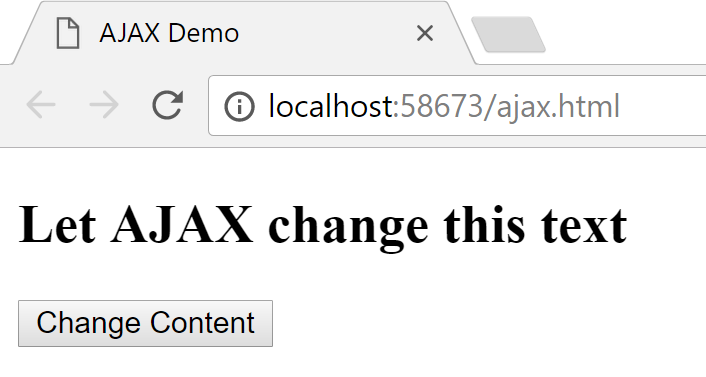
<p>AJAX stands for Asynchronous JavaScript And XML.</p>

<p></p>

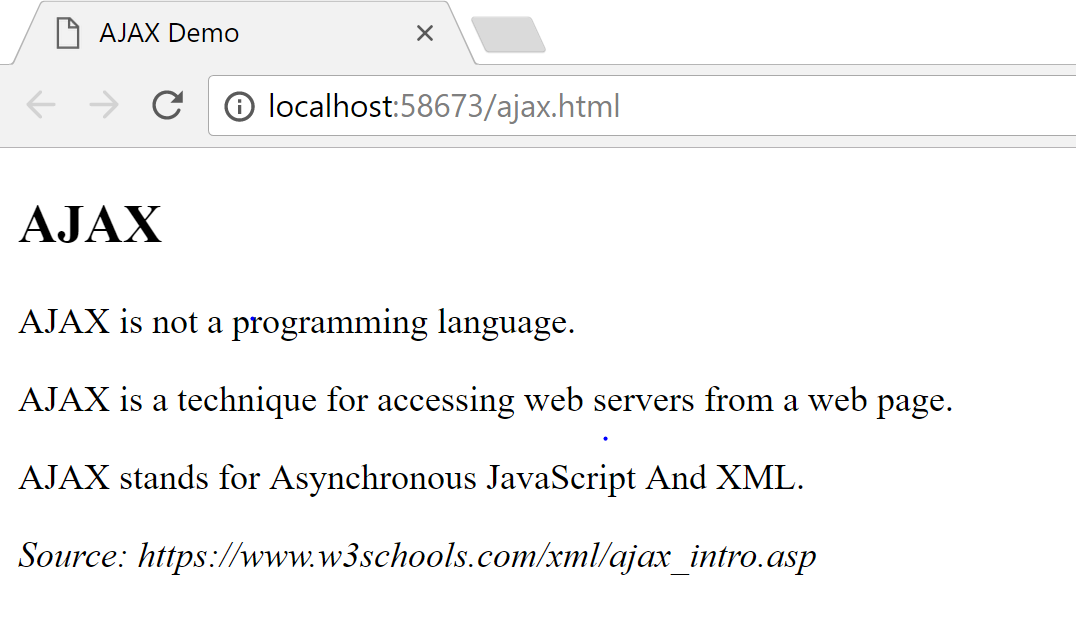
<em>Source: https://www.w3schools.com/xml/ajax\_intro.asp</em>

Notice that you can add styling to the text file.

1. Save the files and view the ajax.html file in browser. You will see the display as follows.



1. When you click the button “Change Content”, contents from ajax\_info.txt will replace the contents of the web page as shown below.



Add a stylesheet to style the following elements:

* A font for the body
* h1 – font size and colour
* p – a different font size and colour
* em – a different font colour

**Task 2: Display information from an external server**

In this task you will follow the tutorial from:

<https://webdesign.tutsplus.com/tutorials/an-example-of-ajax-with-vanilla-javascript--cms-25763>.

1. Add a new HTML page and name it as **index**.
2. In the **index.html** file,
3. Add meta tag for author and your name as content.
4. Add meta tag for description and “Week 14 Practical – AJAX with Vanilla JavaScript” as content.
5. Add the title “AJAX Example with JavaScript”.
6. Copy the HTML code at the end of the page (Here’s the embedded Codepen demo:) from the **HTML** tab from link above into the <body> element of **index.html**.
7. Add a new Style Sheet to the **css** folder and name it as **styles** and copy the contents from **SCSS** tab into this file.

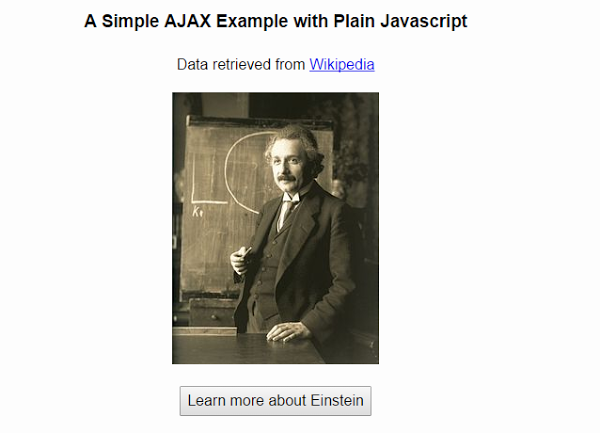
**Remember to click VIEW COMPLETED button to convert SCSS code to CSS** before copying**.**

1. Add a new JavaScript File to the **js** folder and name it as **scripts** and copy the JavaScript code from **JS** tab into this file.
2. Link **styles.css** and **scripts.js** to **index.html** as below.

<link rel="stylesheet" href="css/styles.css" />

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

1. Save the files and view the HTML file in browser. You should see the web page as in the next page.



1. Click the button **Learn more about Einstein**.

When you click the button, the button is hidden and the text in Bio.txt at the link below will appear on the page

<https://s3-us-west-2.amazonaws.com/s.cdpn.io/162656/Bio.txt>

**Task 3: Reading XML data using jQuery**

In this task you will learn to read XML data and display it in a table using jQuery.

1. Add a new HTML Page to the project and name it as **books.html**.
2. Add a new Style Sheet to the **css** folder and name it as **books.css**.
3. Add a new JavaScript file to the **js** folder and name it as **books.js**.
4. Download **books.xml** file from MeL into **txt** folder of the project.
5. Add the code from the Appendix to the respective files.
6. Save the files.
7. View the books.html in browser. You should see the output as in the Appendix.

**Note:** You can see the output only when you run the HTML file within Visual Studio. Alternatively you can open the HTML file in Microsoft Edge.

**Task 4: jQuery show() and hide()**

1. Add a new HTML Page to the project and name it as **jQueryShowHide.html**.
2. Add the following to the <head> element.

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>

    function displayImage(Pet)

    {

        if (Pet == 'Cat')

         $('#ImageSpace').html('<img src="images/cat.jpg"/>');

       else

         $('#ImageSpace').html('<img src="images/dog.jpg"/>');

    }

</script>

1. Add the following to the <body> element.

<button onclick="displayImage('Cat')">Cat</button>

<button onclick="displayImage('Dog')">Dog</button>

<br />

<div id="ImageSpace" style="width:600px; height:200px">

</div>

1. Add images of a cat and a dog to the images folder of the project.
2. Save the file and View in Browser.

You will see two buttons Cat and Dog. When you click the button, the respective image appears on the screen.

1. Add a third button named Hide as shown below.



1. Add the necessary HTML and jQuery code to hide the image of the animal when Hide button is clicked. You must make sure that the image of the animal appears when you click the button of the respective animal again.

**Task 5: jQuery Slide Show**

1. Add a new HTML Page to the project and name it as **jQuerySlideshow.html**.
2. Add the following to the <head> element.

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>

function fadeImage(I) {

var Image = '#' + I; //String concatenation

$(Image).fadeIn();

$(Image).fadeOut(3000, function () { fadeImage((I == 1) ? 3 : I - 1); });

}

</script>

1. Add the following to the <body> element.

<body onload="fadeImage(3);">

<img id="1" src="images/1.jpg"/>

<img id="2" src="images/2.jpg"/>

<img id="3" src="images/3.jpg"/>

</body>

1. Add 3 images from Week 13 Practical exercise to the images folder of the project and rename them as 1, 2 and 3.
2. Add styling for the image to be positioned at top left corner with absolute positioning with suitable width and height.
3. Save the file and View in Browser.

You’ll notice the image fades and the next image appears in 3 seconds. You may play with the timing by changing the value.

1. Add another 2 images to the images folder and modify the HTML file to display 5 images.
2. Save the file and View in Browser.

**Submit the entire Week14Practical folder to your folder in //ictspace.ict.np.edu.sg/FED.**

== End of Worksheet ==

**Appendix**

**books.html**

<!DOCTYPE html>

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

<head>

    <meta charset="utf-8" />

    <title>Reading XML Data Using jQuery</title>

    <script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.js" type="text/javascript"></script>

    <script type="text/javascript" src="js/books.js"></script>

    <link rel="stylesheet" href="css/books.css" />

</head>

<body>

    <input type="button" id="btn" value="Generate table" onclick="generateTable()">

    <div id="content"></div>

</body>

</html>

**books.css**

table {

    font-family: Arial;

    border: solid 2px;

    border-collapse: collapse;

}

th, td {

    border: solid 1px;

    padding: 4px;

}

caption{

    font-size: 1.5em;

}

**books.js**

// JavaScript source code

function generateTable() {

    $(document).ready(function () {

        $.get('txt/books.xml', function(d) {

            var data = "";

            //Defining table header row

            var startTag = "<table><caption>Book List</caption><tr><th>Code</th><th>BookName</th><th>Category</th><th>Price</th></tr>";

            var endTag = "</table>";

            //Extracting from XML file into the table

            $(d).find('List').each(function() {

                var $url = $(this);

                var Code = $url.find('Code').text();

                var BookName = $url.find('BookName').text();

                var Category = $url.find('Category').text();

                var Price = $url.find('Price').text();

                data += '<tr><td>' + Code + '</td>';

                data += '<td>' + BookName + '</td>';

                data += '<td>' + Category + '</td>';

                data += '<td  style=\"text-align: right\">' + Price + '</td></tr>';

            })

            // Writing to the HTML page

            $("#content").html(startTag + data + endTag);

            ;

        });

    });

}

**Output**

