

(10 Marks)

Aims

- Practise how to retrieve the content of an XML document
- Practise how to display retrieved XML data in an HTML web page
- Get used to reading comments and Javascript code

Task 1: Handling Local XML Files using JavaScript

Description

You are required to develop JavaScript code that takes the URL of an XML document, retrieve XML data and display the data in a table in an HTML web page. The XML file contains data about teams from the National Basketball Association (NBA). For an image of the example result, please refer to Figure 2 on page 3.

Design

The design process starts with discussion and paper drawings. Ensure this process is completed before implementation.

Step 1: Form Creation and Presentation (HTML and CSS)

The design presented in Figure 1 will be used.

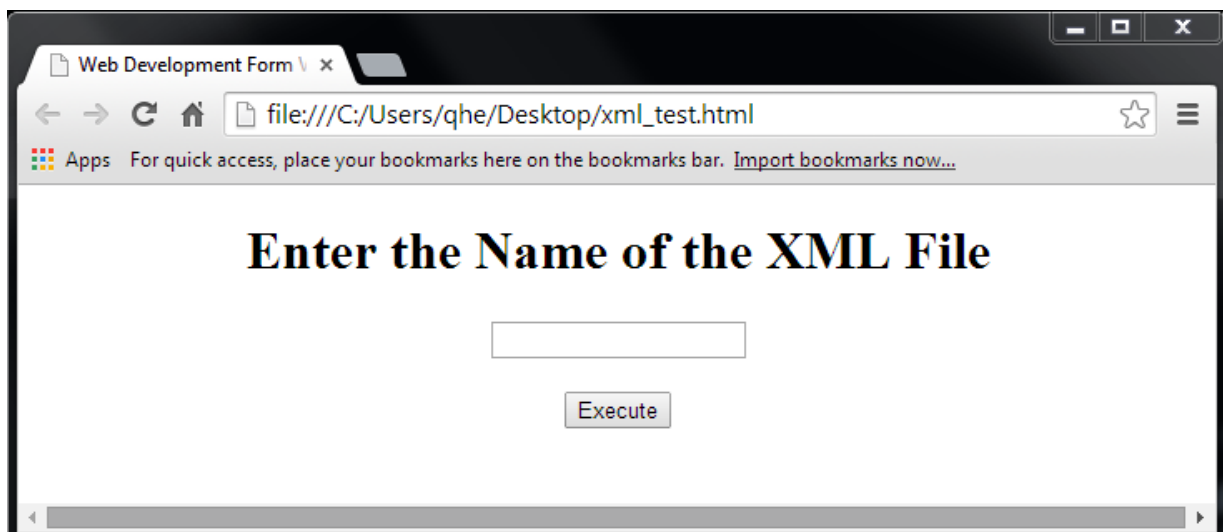


Figure 1. Form Mock Up

IMPORTANT: In this lab, there is only one XML file, named `nba.xml`. Thus, the only valid input for this form is `"nba.xml"`.

Implementation

Implementation requires the creation of HTML, CSS and JavaScript files. In this lab, we will use the files available in `lab11_files.zip` which is available on Canvas.

Step 2: Directory Set Up

- 2.1 Create a new folder 'lab11' under the unit folder on the mercury server `~/COS10005/www/htdocs`. This is the directory where all files will be uploaded.

Step 3: HTML Creation

3.1 Using NotePad++ (or SubLime Text for Mac users), open file `xml.html`.

3.2 Add missing HTML code as required.

For your convenience, the basic code and additional code is shown below:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="description" content="Web development" />
  <meta name="keywords" content="Registration Form" />
  <meta name="author" content="put your name here" />
  (1)link to CSS file
  (2)link to jQuery file
  (3)link to JavaScript file
  <title>Web Development Registration Form</title>
</head>
<body>
  <form id="xmlForm" method="post" action="#">
    <h1>Enter the Name of the XML File</h1>
    <p>
      <input type="text" id="tbFile" />
    </p>
    <p>
      <input type="button" value="Execute" id="btnExecute" />
    </p>
  </form>
</body>
</html>
```

Step 4: CSS Creation

4.1 Open files `style.css`. Review the CSS code, no changes need to be made.

Step 5: Form Data Validation

5.1 Based on the comments provided in `script.js`, complete the following functions. The rules to apply to form validation are:

- 1) The text box must not be empty.
- 2) It must be a valid XML filename.

```
//function execute() will execute function parseXML() only when the form validation
succeeded
```

```
function execute() {
  if (validate()) {
    parseXML();
  }
}
```

```
/* function validate() will validate form data */
```

```
function validate() {
  .....
}
```

```
//link functions to elements' events
```

```
function init() {
  .....
}
```

```
//the initialise function
$(document).ready(init);
```

Note: Please read the comments in the provided JavaScript file to understand how it works and how to complete those functions.

Step 6: Handling the XML Document

6.1 Based on the comments provided in script.js for function `parseXML()`, complete function `parseXML()` which will parse the XML document and display the data in a table.

```
function parseXML() {
    .....
}
```

Note: Refer to the lecture slides and the comments in script.js about how function `parseXML()` works and how to complete it.

The final solution will display a page as presented in Figure 2.

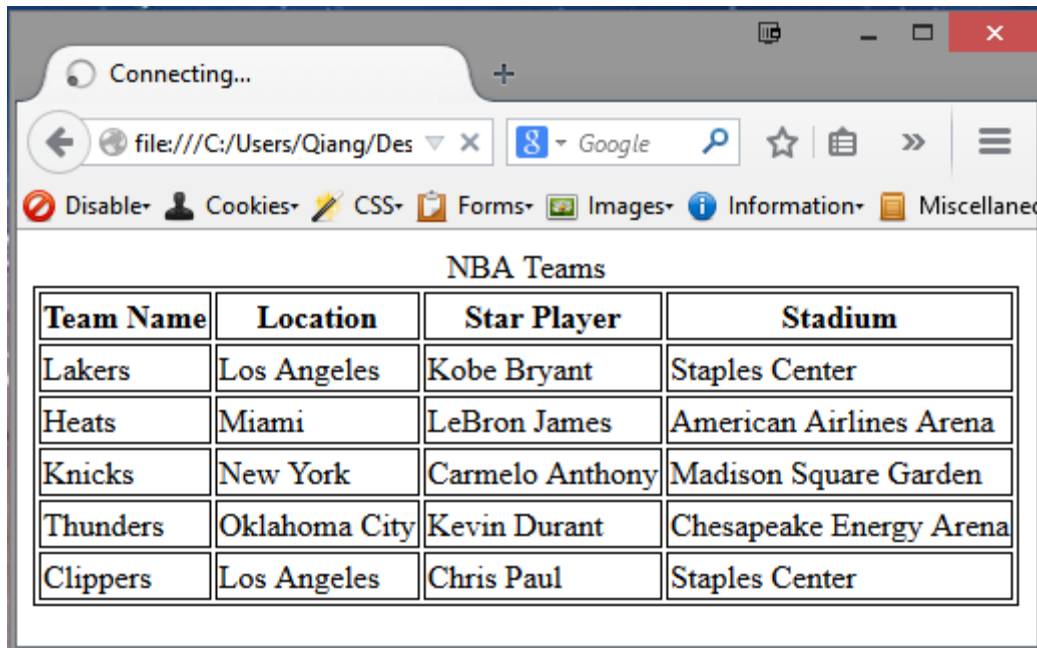


Figure 2. Example Result of Task 1

Testing and Quality Assurance

Test your code for errors.

Note: For simplicity, please save your HTML, CSS, JavaScript and XML files in a same folder.

Step 7: Test and view web pages.

1.1 Using WinSCP, upload **all** your files onto Mercury.

1.2 To view the pages through http, use any Web browser and type in the following address,

`http://mercury.swin.edu.au/<your_unit code>/s<your Swinburne ID>/<folder>/<filename>`

Please refer to the following examples to identify the URLs of your web pages.

| Folder on Mercury Web Server | URL |
|---|---|
| ~/cos10005/www/htdocs/index.html | http://mercury.swin.edu.au/cos10005/s1234567/index.html |
| ~/cos60002/www/htdocs/lab11/xml_test.html | http://mercury.swin.edu.au/cos60002/s1234567/lab11/xml_test.html |

Step 8. HTML and CSS Validation

- 2.1 To validate the HTML file, use the validator at <http://validator.w3.org>.
- 2.2 To validate the CSS file, use the <http://jigsaw.w3.org/css-validator>.