**Web Application Design**

Daily Exercises

Various

© FDM Group Ltd 2014. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

# Table of Contents

Daily Exercises 3

1.1 Aims, objectives and key topics 3

1.2 Project overview 3

2 Exercise Set Up 4

2.1 Web Site Set Up 4

2.2 Using the Web Server 5

3 Exercises 6

3.1 Day 01 6

3.2 Day 02 11

3.3 Day 03 14

3.4 Day 04 16

# Daily Exercises

## Aims, objectives and key topics

|  |  |
| --- | --- |
| **Aims** | This daily exercises introduce trainees to application design concepts such as modelling requirements using UML and developing applications using common web technologies. |
| **Learning objectives** | After completion of the exercises, delegates should be able to:   * Document business requirements using UML * Design & build web pages using HTML5 and CSS3 suitable for a range of devices * Implement basic dynamic behaviour on web pages using JavaScript * Implement basic validation of data entry using JQuery * Develop valid and well formed XML data structures |
| **Topics covered** | * Modelling requirements using Unified Modelling Language (UML) for Use Cases and Activity Diagrams * Introduction to Hypertext Mark Up Language – HTML5 * Introduction to Cascading Style Sheets – CSS3 * Introduction to Responsive Web Design * Introduction to XML, XSD & XSLT * Introduction to JavaScript * Introduction to JQuery |

## Project overview

|  |  |
| --- | --- |
| Length of exercises | Each set of daily exercises should be allocated no more than 90 minutes. |
| **Pre-requisites** | Exercises may be undertaken as soon as delegates have covered the relevant material for those exercises. |

# Exercise Set Up

This section details how to use the web server provided by FDM for use during the training, exercises and project.

It is important you host your web site correctly to avoid problems caused through browser security settings.

## Web Site Set Up

Your trainer will provide you with a set of files to install on your workstation.

You should follow the instructions provided with the files or by your trainer to correctly set up your working environment.

You should have this folder structure:

* WAD
  + WebSite
    - images
    - scripts
    - styles
    - xml

Use the folders as follows:

* WebSite – place your html documents here
* images – place any images you wish to use here
* scripts – place javascript/jquery files here
* xml – place your xml data files here

## Using the Web Server

It is important you view web pages hosted within a web server and not just directly by opening them from the file system to avoid problems.

Some browsers have strict security settings in operation when running from the file system. By hosting your work in a web server you can overcome these restrictions.

### Start the web server

In the “WAD” folder, you should find a file called “launchWebSite.cmd”. Double click this file to launch the web server and host the files you have created in the “website” folder.

You should now have a Command Window running called “Web Site”. You can minimise this and ignore it. Do not close it.

To view a web page hosted by the web server, enter the following into your web browser:

http://localhost:7777/nameOfYourWebPage.html

Notes:

* The web server hosts your website on a special address “localhost:7777”
* The web server is not accessible from any other computer

### Adding content AFTER the web server has started

Most of the time, if you add new content in the folder structure it will be picked up automatically. Simply refresh your web browser or navigate to the new content.

If your change is not apparent – you may need to stop and start the web server.

### Stop the web server

To stop the web server at any time, type “Q” in the command window running the web server.

No other action is required to stop the web server.

# Exercises

This section contains the exercises appropriate for Day 01 of the training. Exercises are not formally assessed.

Each exercise contains a table of tasks and may contain some associated items to investigate.

## Day 01

The following exercises are to be attempted prior to completion of relevant project work.

### UML

There are no practice exercises for UML outside the training session itself.

### HTML5

The following exercises cover introductory HTML and HTML5 and allow you to discover elements and usage.

#### Create an HTML5 compliant web page

*This exercise will help you understand how an HTML5 document is declared.*

|  |  |
| --- | --- |
| Task | Investigate |
| Construct a basic HTML5-compliant document | DOCTYPE |
|  | Elements: html, head, body |

Navigate to the document using your web browser.

#### Apply a layout using HTML5 semantic elements

*This exercise will help you determine which layout elements may help your project design.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add semantic elements to the document for overall layout | Elements: header, nav, section etc... |
| Add a block element with some text | Element: h1, p, table |
| Add an inline element with some text | Element: span, img |
| Add a div element containing other elements | Element: div |

View the results in your web browser.

#### Try some HTML5 elements to display data

*This exercise will help you determine which HTML/HTML5 elements may help your project design.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add a progress bar to your document | Element: progress |
| Add a meter to your document | Element: meter |

View the results in your web browser.

Question: How does a progress element differ from a meter?

#### Create a basic user feedback form

*This exercise will help you determine which form input elements may help your project design.*

Design and implement a user feedback form which will request the following information:

* User first name, last name
* User age
* User email
* User’s own website
* Rate this site (“0 is bad, 5 is awesome!”)

|  |  |
| --- | --- |
| Task | Investigate |
| Add a form, give the form a legend | Elements: form, legend  Attributes: action, method |
| Within a fieldset, add input elements to accept user information | Elements: fieldset, input  Attributes: input “type” |
| Add a datalist | Element: datalist |
| Add an input element that uses the datalist to provide a shortlist of acceptable entries |  |

View the results in your web browser.

Question: How what are the form ACTION and METHOD attributes designed for?

#### Add validation to this form

*This exercise will help you understand how you can validate data with HTML5 in your project.*

|  |  |
| --- | --- |
| Task | Investigate |
| Ensure all fields are “required” |  |
| Ensure user age is greater than 13 and less than 30 | Input type=number |
| Correctly formed email address entered | Input type=email |
| Correctly formed website address entered | Input type=url |
| User feedback is rated between 0 and 5 only | Input type=range |

View the results in your web browser.

Question: What other element might allow you to display a range selection instead of just entering a number?

### CSS

The following exercises cover introductory CSS. More detailed CSS3 exercsies follow on Day 02.

#### Add an external style sheet

*This exercise will help you understand how to link an external style sheet.*

|  |  |
| --- | --- |
| Task | Investigate |
| Create an external style sheet and link it form your HTML document | Linking CSS external style sheets |
| Add some text to the body of your document |  |
| Add a style in the external sheet, for the body element, red font, 24pt |  |

Navigate to the document using your web browser.

#### Investigate Style Precedence

*This exercise will help you understand how CSS “cascading” works.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add a style definition for the body element within the head element of your document. Ensure the font color is different from the external style definition. |  |
| Add a style definition for the body element as an attribute of the body element. Ensure the font color is different to both the previous styles defined. |  |

Navigate to the document using your web browser as you complete each task to view the difference.

#### Investigate the CSS Box Model

*This exercise will help you understand the CSS box model.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add a div element to your page. |  |
| Within the div, add a paragraph element  Add text: “Here is the content” |  |
| Add a style declaration for div:  {width: 20%; border: 5px dashed black} |  |
| Add a style declaration for p:  {border: 3px solid red; margin: 5px; padding: 10px } |  |
| Amend the style for p. Alter the padding |  |
| Amend the style for p. Alter the margin |  |
| Alter the padding:  top/bottom 25px, left/right 50px | Padding “shorthand” property |

Navigate to the document using your web browser as you complete each task to view the difference. *You may need to pay very close attention to observe the difference...*

#### Element Dimensions

*This exercise will help you understand how the CSS Box Model impacts overall sizes.*

|  |  |
| --- | --- |
| Task | Investigate |
| Calculate the overall width of an element with this style:  {border:5px solid red; padding:25px 50px 75px 100px; margin:25px 50px; } | Padding shorthand property  Border shorthand property  Margin shorthand property  Calcualting overall width for an element |

Question: If the element was contained by another element, woud the containing margin, border, padding apply?

#### Investigate CSS Pseudo Classes

*This exercise will help you understand pseudo-classe and pseudo-elements.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add pseudo classes to style anchor (<a>) elements | Anchor pseudo classes |
| Add a div element, containing two p elements.  Add text to both p elements |  |
| Add a style declaration that would ONLY impact the first child p of the div element | :first-child pseudo class |
| Style the first line of text in a paragraph differently from the remainder of other text | :first-line pseudo element |
| Style the first letter of text in a paragraph different form the remainder of the text | :first-letter pseudo element |
| Use a pseudo element to INSERT additional content before some fo your text | :before pseudo element |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

Question: when styling anchor elements using pseudo classes, is the order you list the pseudo classes important?

#### Styling Navigation Links

*This exercise will help you understand how you can use CSS to radically alter the loko/feel of basic elements.*

|  |  |
| --- | --- |
| Task | Investigate |
| Add an unordered list containng the following items:  “Home”, “Music”, “”Books”, “DVD” | HTML elements:ul, li |
| Each list item should also be an anchor element. The href is unimportant at this time |  |
| Style the unordered list to remove bullets, margin and padding | list-style-type |
| Style the anchor elements as block elements with a consistent width to give an consistent look | CSS: display: block |
| Switch to a horizontal menu style by changing the list items as “inline”  Hint: you may need to comment out the previous anchor style... |  |

Questions: What is the impact of “block”? What is the impact of “inline”?

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

## Day 02

The following exercises are to be attempted prior to completion of relevant project work.

### CSS & CSS3

The following exercises cover some CSS and CSS3 topics to further develop your skills and knowledge.

*All styles should be defined with in the document HEAD or in an external style sheet.*

#### Investigate CSS Positioning

*This exercise will help you understand how to position elements using CSS.*

Create a new HTML document called: “cssPositioning.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Using the website: <http://www.lipsum.com>  Generate sufficient text to create a lengthy web page which would reult in vertical scrolling required to get to the end. |  |
| Add a paragraph element with text: “Hello World” |  |
| Style the paragraph to ensure it remains in view near the top right of the browser window even whislt scrolling | position: fixed |
| Add a div element to contain a new paragraph element. Add text to this paragraph.  Style the div with fixed position. |  |
| Style the paragraph to ensure it remains in a specific position relative to its containing div regardless of the flow of the document | position: absolute |
| Add a div element to contain a new paragraph element. Add text to this paragraph.  Style the div with fixed position. |  |
| Style the div to place it relative to its position in the flow of the document | position: relative |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

#### Investigate CSS Floating Behaviour

*This exercise will help you understand how to float elements using CSS.*

Create a new HTML document called: “cssFloat.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Using the website: <http://www.lipsum.com>  Generate sufficient text for 4 mulitple line paragraphs | CSS: float |
| Add the paragraphs to the html document |  |
| Add an image to the html document. |  |
| Style the image to float to the far left / right |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

#### Investigate CSS Floating Behaviour

*This exercise will help you understand how to float elements using CSS.*

Create a new HTML document called: “cssFloat.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Using the website: <http://www.lipsum.com>  Generate sufficient text for 4 mulitple line paragraphs | CSS: float |
| Add the paragraphs to the html document |  |
| Add an image to the html document. |  |
| Style the image to float to the far left / right |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

#### Investigate CSS3 Media Queries

*This exercise will help you understand how to implement simple responsive web design using media queries.*

Create a new HTML document called: “cssMediaQueries.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Using the website: <http://www.lipsum.com>  Generate text for one multiple line div |  |
| Style the column for 5 columns | CSS: columns |
| Add media queries to detect changes in browser window maximum width:  max-width: 1280px  max-width: 800px  max-width: 640px  max-width: 320px  To prove the the media query is working, change the body background color for each one. | CSS: media queries |
| For each media query, reduce the number of columns used for display by applying a different style |  |
| Write an additional media query, suitable for tablet/mobile devices that would detect portatrait/landscape mode. |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

#### Proportional Layout Calculations

*This exercise will help you understand how to calculate layout sizes using proportional layout and “ems” instead of pixels or percentages.*

Create a new HTML document called: “cssProportionalLayout.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Add a div element to act as “context” | Proportional Layout |
| Add a p element within the div to be the “target” |  |
| Assume context width:  800px  Assume target width is:  640px  Calculate the target width as a percentage of the context to ensure it sould stay in proportion, |  |
| Assume the browser default font-size is 16px,  and the font for our paragraph is 32px | Proportional Layout & “ems” |
| Calculate the proportional font size in “ems” to use for the paragraph |  |
| Test your calculations by adding a div and p to your document and styling them appropriately |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour

## Day 03

The following exercises are to be attempted prior to completion of relevant project work.

### XML

#### Build an “well formed” XML document

*This exercise will help you understand how to create simple, unvalidated XML.*

Create a new XML document called: “trainees.xml”

|  |  |
| --- | --- |
| Task | Investigate |
| Add an XML document declaration | XML document declarations |
| Add a root element “trainees” |  |
| Add elements and data within the root element to model the following “trainee” record structure:  firstname  lastname  university  degree  date of graduation |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

### XSD

The following exercises cover some XSD topics to further develop your skills and knowledge.

#### Create an XML Schema Document to Validate XML

*This exercise will help you understand how to create simple XSD for validation.*

Create a new XSD document called: “trainees.xsd”

|  |  |
| --- | --- |
| Task | Investigate |
| Use the XML document “trainees.xml” (created during Day 2 exercises) |  |
| To be a “valid” XML document the following rules must be applied using XSD:  All trainee records must be complete  There may be 0, 1 or many trainee records  Required datatypes:  firstname: string  lastname: string  university: string  degree: string  date of graduation: date | XSD: datatypes  XSD: complextypes |
| Add a link to your XSD from the XML document |  |

Use an online XML Validation tool (e.g. <http://www.xmlvalidation.com>) to validate your trainee data file using your validation rules.

### XSLT

The following exercises cover some XSLT topics to further develop your skills and knowledge.

#### Create an XSLT document to style XML

*This exercise will help you understand how to create simple XSLT style documents for presentation of your data.*

Create a new XSLT document called: “traineesWithStyle.xsl”

|  |  |
| --- | --- |
| Task | Investigate |
| Use the XML document “trainees.xml” (created during Day 2 exercises) |  |
| In the XSLT document:  Match the root element of trainees.xml  The template should create an html5 document | XSLT: templates match |
| In the XSLT document add templates for:  firstname and lastname  (combine these in one template)  university  (hyperlink e.g. http://www.unversity.ac.uk) | XSLT: variables |
| In the template that matched the root element:  Loop through each trainee, apply the correc template(s) for the other data | XSLT: for-each  XSLT: apply-templates |
| In the template that matched the root element:  Add a style sheet link to a new stylesheet |  |
| In the stylesheet:  Add styles to improve presentation of the data | HTML: linking external style sheets  CSS |
| Add a link to your XSLT from the XML document |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

## Day 04

The following exercises are to be attempted prior to completion of relevant project work.

### JQuery

The following exercises cover some JQuery topics to further develop your skills and knowledge.

#### Investigate using JQuery to Manipuate Elements

*This exercise will help you understand how to manipulate elements using JQuery.*

Create a new HTML document called: “jQueryElements.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Add two div elements and some text for each. Give each div a unique id attribute |  |
| Style both divs using CSS and apply the styles using the class attribute of the elements |  |
| Add an input element (button) to the document | JQuery: Actions in response to Events |
| Add a script element in the document head | Scripting: JavaScript script element |
| Write a JQuery script that, when the button is clicked, changes the text of one of the div elements | JQuery: text() method |
| Amend you script so that it also changes the style applied to the same div | JQuery: css() method |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.

#### Investigate using JQuery to Retrieve XML Data

*This exercise will help you understand how to navigate XML using JQuery.*

Create a new HTML document called: “JQueryXML.html”

|  |  |
| --- | --- |
| Task | Investigate |
| Use the XML document “trainees.xml” (created during Day 2 exercises) |  |
| Write a JQuery script to query the web server for the trainee.xml data | JQuery: AJAX GET |
| Once successfully retrieved, search the xml for all universities within the data | JQuery: .find() method.  Selecting XML nodes and using the “:contains” syntax with the find() method |
| Display the list of universities within a div element |  |

Navigate to the document using your web browser as you complete each task to view the differences and behaviour.