CSC8004: Web Technologies Assignment

Aims

The aim of this assignment is to introduce you to some of the practical skills required to create functional, standards-compliant web pages using XHTML, CSS and JavaScript.

Objectives

- To use XHTML and CSS to structure and format a web page which adheres to a (provided) specification.
- To gain experience developing a website in accordance with strict constraints on the technologies used, namely XHTML 1.0 Strict and CSS 2.1. This intends to emulate professional website development practices, working alongside a Quality Assurance department on legacy projects.
- To use JavaScript to validate data supplied in a web form and process the input prior to submitting it to a remote server.

Getting started

This assignment is in two parts, read through all of the specification carefully before starting.

Download the following files from the resource collection (Blackboard) for these tasks:

```
bsl-XHTML.html
bsl-XHTML.png
bsl-CSS.png
bsl-QUIZ.html
```

Part 1: Using CSS

Instructions

Add valid CSS to `bsl-XHTML.html` in order to present the document as shown in the resource file:

`bsl-CSS.png`

Save your finished document and submit it to NESS as:

`bsl-CSS.html`

All your CSS must be contained inside a single external file, linked to `bsl-CSS.html` and called:

`bsl-CSS.css`

Comments should be included in your style sheet to explain the basic functionality of each rule and (if you found inspiration elsewhere) the source of your CSS code.

Hints

Redundant or extraneous mark-up will cost you marks. Think "lean and clean". If you find yourself asking, "have I put too many tags in here?"… then you probably have ;-)

You will need to add some additional XHTML to `bsl-XHTML.html` in order to provide framework for your CSS. You should just need <`div`> & <`span`> tags and `class` and `id` attributes to do this. If you are rewriting large parts of the document, you have probably made it too complicated!

The final page layout and functionality should appear and behave identically in recent versions of Firefox, Google Chrome and Internet Explorer. Your page will be tested in all three on a Windows platform. For the purposes of this exercise, other browser types do not have to be explicitly catered for. Media content of your page does not need to show on Internet Explorer, since the browser recently stopped to support Flash.

The screen shot `bsl-CSS.png` is taken from a browser window set at approximately `1900` pixels wide. You can make measurements off it if you wish, however this may not help you much beyond relative proportions for margins, borders etc. You will find some other sizing clues in the assets collection. You do not have to pixel perfect to pass.

Validation is a simple yes/no check and you will know when you submit your work whether it will pass or fail. It is not a trick question... everything in this document can be presented as XHTML 1.0 Strict/CSS2.1 if you take the time to find out how.

Assets & Resources

Additional image assets, which may be useful for Part 1, can be found at:

http://homepages.cs.ncl.ac.uk/harold.fellermann/csc8004/

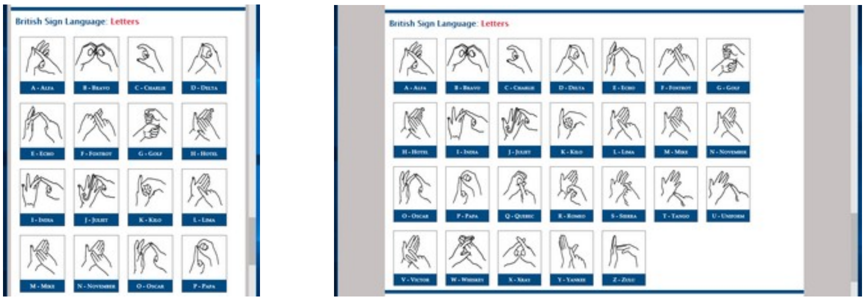
Marking Criteria

Core features

The main (white) content block should be centred in the browser window and flexible between widths of `510` and `960` pixels. It should not expand/shrink beyond those limits.

Use CSS to provide the layout for each of the main content sections. Pay particular attention to the use of background images, floats and CSS positioning. Make sure you correctly display the *all* images and media shown in `bsl-CSS.png` as well as the whitespace around them. Again, use absolute URLs for the images you include.

The sign grids (letters and numbers) should dynamically adjust to fill the width available to them within the flexible content limits i.e.



Narrow and wide page sign display

6 marks

Colours and typography

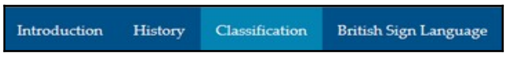
You should be resourceful enough to find out the correct *hexadecimal* codes for the colours used in the final version. Do so... and use them correctly.

The core typeface in use is Palatino Linotype, with Lucida Sans Unicode for the headings. A little research should enable you to build simple font stacks to implement them correctly using CSS.

2 marks

Navigation bar

The navigation menu should be presented, as shown in `bsl-CSS.png`, using CSS only. You must use CSS to move the navigation bar – you cannot re-order the XHTML (see below). In addition, you should change the background colour of the menu items when the user moves the move over them e.g.



Colour change occurs when user moves over the Classification menu item

3 marks

Underlying framework

If the CSS is removed from `bsl-CSS.html`, the underlying page must still look identical to the reference example shown in the resource file `bsl-XHTML.png`

This means you must not re-order the supplied content.

2 marks

Validation

Your finished style-sheet `bsl-CSS.css` must validate with no errors or warnings when checked against CSS level 2.1 at:

http://jigsaw.w3.org/css-validator/

Your finished document `bsl-CSS.html` must contain a suitable, full DTD and validate automatically as XHTML1.0 Strict when checked by file upload at:

http://validator.w3.org

4 marks

Total available for Part 1: 17 marks

Part 2: Client-side Scripting

Instructions

This part uses the following file from the resource area for these tasks:

```
bsl-QUIZ.html
```

This is a basic XHTML form which presents a short quiz about bsl. View this file in a text editor and note that it links to a single external JavaScript source file called:

```
validate-QUIZ.js
```

At the moment that file doesn't exist – your primary task is to create it, according to the criteria given below.

You will not be able to edit `bsl-QUIZ.html` in any way – all you will be submitting is your new JavaScript source file.

Also notice that `bsl-QUIZ.html` also links to a CSS style sheet called:

```
bsl-QUIZ.css
```

This is a CSS style sheet that you can create as a secondary objective for this assignment. See the specification for details.

Hints

All the functionality provided by your JavaScript must be triggered by the function called `validate()`

The final `return` value from `validate()` will determine whether the form submits to the PHP script or not. The event handler and call to this function is already in `bsl-QUIZ.html`

The `action` for the form is set to automatically submit to a PHP script which simply returns the values to the screen so you can check they arrived safely. Make sure that the input from the form and the score show up on that page

The CSS you use in Part 2 does not need to fully replicate that of from Part 1! For a start you cannot make any changes to the XHTML, so you will have to use what you have learned about CSS selectors to apply your style rules. The goal is to make the quiz look *consistent* with the main bsl information page – not identical. At the very least the colours and fonts should be pretty easy…

Marking Criteria

Input checking

Your JavaScript should check all four questions and ensure that the user provides a name and has selected *at least one response* for each before continuing:

- If any questions have not been answered, display an alert to the user telling them which question(s) they still need to complete.
- The incomplete question(s) for this attempt should be highlighted in yellow in the page
- Include functionality to ensure the user *only* selected 2 options for question 2
- The form should not submit to the server (in order to allow the user to return and complete it)
- The user should not be alerted about their score at this point

6 marks

Answer scoring

When all the questions have been attempted, your JavaScript should also mark the quiz as follows:

- Award 1 point for each correct answer (so there will be a max of 5)
- Accept only the correct spelling for question 4, but allow for the fact that the user could type it in any case (upper lower or mixed).
- Display an alert to the user telling them how many they scored (out of the maximum of 5) and that their answers and score will be sent to the server.
- Make sure you know the correct answers!

5 marks

Submit score

Finally when all the validation is complete and passed, your JavaScript should allow the browser to submit the form results *and score* to the PHP script on the server.

3 marks

Errors

Your final JavaScript should run error free in recent versions of Firefox, Google Chrome and Internet Explorer (8+).

2 marks

Style

Create `bsl-QUIZ.css` and add rules to give the quiz a look and feel which is *consistent* with `bsl-CSS.html`. There's not many marks available for this, so don't spend too much time on it!

2 marks

Total available for Part 2: 18 marks