

Week 3 Lab – Page design and Introduction to CSS

Aims:

- Design a mock-up for your assignment
- Use an external CSS file to format an HTML file
- Use a range of CSS selectors and properties
- Gain the skills and knowledge towards completing Assignment Part 1.

Task 1: Create the html for index.html

If you haven't already done so, create the HTML content for the first page of your assignment Website.

Create a file called `index.html` and store it in a directory called `assign1`.

The HTML should include the following elements:

- Appropriate `<meta>` and `<title>` elements in `<head>`
- Within the `<body>` element, structural elements including
 - `<header>`
 - `<nav>`
 - `<section>` or `<article>`
 - `<footer>`
- Within the structural elements, content elements including
 - one or more headings
 - menu items as described in the assignment
 - some introductory content in a paragraph(s)
 - a graphic

If you haven't finalised the content or sourced a graphic yet just use some dummy text and any graphic, and replace these at a later stage.

Task 2: Design a mock-up for your Assignment

The aim of this task is to design a mock-up of the web site you will develop for your Assignment.

This design should include:

1. The standard layout common on all pages
2. The layout of individual pages, with the framework defined by your standard layout.
3. A diagram of the navigational flow between your web pages, defined by their hyperlinks.

The mock-up should clearly communicate to your 'client' (i.e. your tutor) the look, layout and overall functionality of your web site.

The mock-up can either be manually sketched on paper or developed with a computer drawing application to create a 'wireframe'. For example, using "Word" or "Powerpoint" or a specific tool like "Pencil".

Your mock-up should be saved, so you can show it to your tutor.

The standard layout should have a:

- A header (with a logo)
- A navigation area (a menu with links to your other pages)
- A content area – with sections, instructions and/or form(s)
- A footer area – document information including the author's name

Pencil is an open-source GUI prototyping sketch tool that's available for all platforms, and as a Firefox extension. The "Pencil Project" home page is at <http://pencil.evolus.vn/> development is active, and Version 3 has just been released. The Firefox Add-on at <https://addons.mozilla.org/en-US/firefox/addon/pencil-prototyping/> does not appear to work with current Firefox versions. The open-source development is active continues on Github. Try the various shapes in the collections (e.g. 'Sketchy GUI') and to create the mock-up in your browser.

An effective way to illustrate navigation flow is to draw an arrow from the source element on one page (e.g. an anchor hyperlink menu or submit button), to the target page.

Think about the most appropriate structure for your web site.
Common information structures for website organisation are:

- Hierarchical - Tree
- Linear - Linear, Tutorial
- Network (Exploratory) – Web, Cluster, Catalogue

Your Assignment website may be some hybrid of these structures.

See “Web Style Guide Edition 3 – Information Architecture”

<http://webstyleguide.com/wsg3/3-information-architecture/3-site-structure.html>

Task 3: Applying CSS rules for Typography

Step 1. Create a CSS file and link to the web pages.

In this task you will apply some simple typographic and colour CSS styling to a Web page. We will use a version of the Web page used to demonstrate basic HTML elements in lecture 2. This HTML file is in **lab03.zip** and is called **lab03.html**. Put it in your **lab03** folder on your local machine.

In your **lab03** directory, create a subfolder called **styles**. Using Notepad++ or similar editor, create a new text file called **lab03.css** and save it in **lab03/styles**. Add a comment header to the top of your CSS file similar to the following replacing the text in *italics*:

```
/*
filename: [your name]
author: [your name]
created: [enter date]
last modified: [enter date]
description: [html files it refers to (if known)]
*/
```

Create a link to this file from your web page **lab03.html** by adding a reference to the external stylesheet in the <head> element as follows:

```
<head>
  <!-- other meta stuff -->
  <link href= "styles/lab03.css" rel="stylesheet"/>
</head>
```

Step 2. Create a CSS Rule

CSS style rules have the form

```
selector      {      property1: value1;
                   property2: value2;
                   ...;}
```

where **selector** identifies the markup elements to which the style **properties** will be applied.

For example, if we if we want to make <h1> elements coloured **purple**, double size and **bold** the rule would be

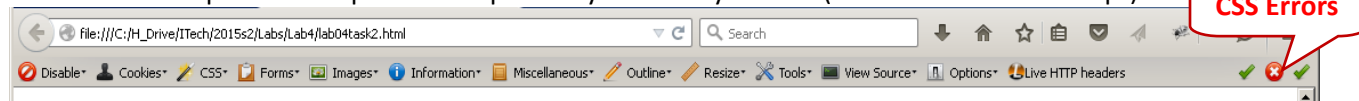
```
h1 {color:purple; font-size: 2em; font-weight:bold; }
```


Enter this rule into your stylesheet **lab03.css**.

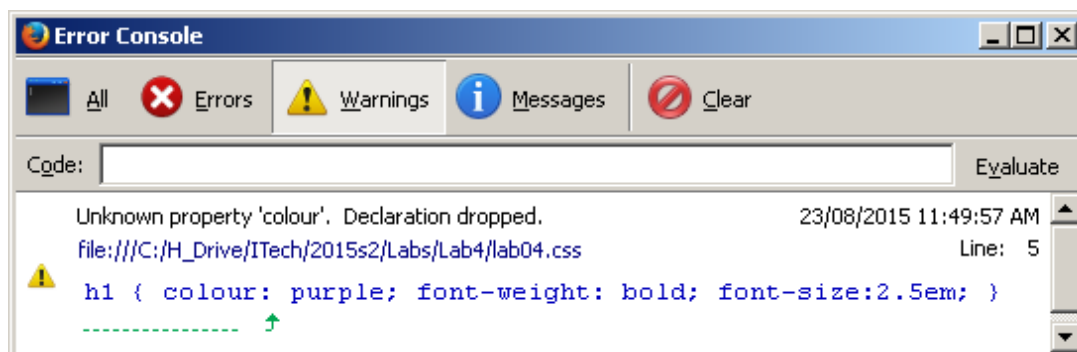
Reload your HTML file and check the Heading 1 style has changed.

Note the US spelling of
“color” and the semi-colon
after each property value

The Web Developer Toolbar provides a quick way to check your CSS (and HTML and JavaScript)



Clicking on the  brings up the Error Console with a warning explanation (eg. “colour” should be “color”!).



Step 3. Validate your CSS

1. Go to the W3C Validation Service web site at <http://jigsaw.w3.org/css-validator/>
2. Upload your CSS file and check it for errors.
3. Fix any errors.
4. Validate again this time using the Web Developer Extension of Firefox.
 - a. Load a web page(s) locally and check the local CSS.

Load the web pages from the server and check the CSS from there.

Check your browser extension is validating as CSS3:

The “Web Developer Extension” Toolbar may validate as CSS2.1 with ‘Validate CSS’ and as CSS3 with ‘Validate Local CSS’. If so, you can set the “Web Developer” Add-on in Firefox (or the similar add-on in ‘Google Chrome’) to validate against CSS3, by going to the Options menu on the toolbar. Select ‘Validate CSS’ to edit and set the validator ‘CSS3’ with the URL

<http://jigsaw.w3.org/css-validator/validator?profile=css3&warning=0&uri=>

Step 4: Use a range of selectors and properties to create CSS rules

To apply CSS properties in the following exercise we will use a range of **selectors**:
element, id, class, group contextual and pseudo.

The results of the application of these CSS rules is shown on the last page of these instructions.

Reference the Week 3 CSS Lecture notes to see how simple typographic properties are assigned values defined in various measurement units. The W3C schools website <http://www.w3schools.com/cssref/default.asp> also provides a useful ready-reference to CSS properties.

Element selectors

CSS has a *universal element selector* `*` that selects all elements. Use this to change the standard font on the Web page to “calabri”. If Calabri is not available we will default to “sans-serif”

```
* {font-family: calabri, sans-serif;}
```

If the font name has spaces in it you need to put quotation marks around it e.g. “Time New Roman”

It is good practice to place universal selector rules at the top of your CSS file for easy reference.

In Step 2 we set the colour of <h1> element to the named colour "purple". We can also use **hexidecimal** and **RGB** formats to represent colour. <http://www.w3.org/TR/css3-color/#svg-color> lists a mapping of the CSS3 named colours to these representations.

1. Set the default colour of all elements on the web page to hexadecimal **#2F4F4F** , using the *universal element selector*.

What is the colour of <h1> now?

Note that more specific selectors override more general selectors.

2. Using one style rule, set the colour of all Heading 2 elements to RGB value `rgb(0,128,128)`, and make them bold with font size 1.5 times the standard width.

Group selectors

With a *group selector* we can set the style properties of a number of different elements (or other selections) at once. Selectors are separated by commas.

3. Make the text in the <aside> and <footer> elements *italic* using the rule

```
aside, footer {font-style: italic;}
```

Notice that although the text to be made italic is inside <p> elements, the style applies to all **descendant** elements of the selection.

4. Use a *group selector* to make all heading elements (h1, h2, h3) **arial** font, or **sans-serif** if arial is not available.

5. Use an element selector to set the table element to have a border with the property

```
border: 1px solid black;
```

What is the result? Use a *group selector* to apply the above border property to the table header cells and table data cells as well as the table element.

ID selectors

If you are applying the same property value to a number of elements in a page, use a class selector (see below) .

To select a **single** element on a web page, we give that element an **id attribute**, say, `id="my_id"` , then use a CSS **hash** selector `#my_id`

6. Look at the HTML source. You will notice the **Images** heading has an id as follows:

```
<h2 id="images">Images</h2>
```

Edit your CSS file to make that heading text centre-aligned with the rule

```
#images {text-align: center;}
```

Careful: US spelling!!

7. On the HTML page, create a suitably named **id** attribute for just the <h3>Headings</h3> element and change its colour to red.

If you are only styling a single element, do not use a class selector

Class selectors

To select **multiple** elements on a web page, we give that element a class *attribute*, say, `class="my_class"`, then use a CSS **dot** selector `.my_class`

8. Look at the HTML code. You will notice that in-line ` ...` elements have been put around the special characters, and given a class attribute ``
Edit your CSS file to make the special class darkorange colour

```
.special {color: darkorange; font-weight: bold;}
```

9. We now want to make the month table headings `<th>Jan</th>` and `<th>Feb</th>` right aligned in their cells. On the HTML page, create a suitably named class attribute in these elements. Then, in the CSS, write a style rule that selects this class style to right align these headings.

Contextual selectors

A contextual selector matches the style when an element is an arbitrary descendent of some ancestor element (i.e., it may be *any* generation below the ancestor element). A contextual selector is made up of two or more selectors separated by white space.

10. Having hyperlink underlined on the menu can make a menu look too busy. But we don't want to remove underlines on all hyperlinks - only those who are children of a `<nav>` element. In the same style rule we want to add a bit of space around the menu items and add a border:

```
nav a { text-decoration: none;
padding: 0.2em 0.6em;
border: 4px solid #ccc;}
```

11. The `list-style-type` property allows us to set the number style of an ordered list e.g.

```
ol {list-style-type: lower-roman;}
```

or the appearance of a bullet point in an unordered list e.g.

```
ul {list-style-type: square;}
```

12. Create a rule with a contextual selector that selects only unordered lists that are *nested within* an ordered list, and use this to make the bullet points square.

Pseudo classes

Pseudo classes are used to define style rules for the current state of an element being viewed in the browser; e.g. a mouse-over or visiting a hyperlink.

For example, the `a:hover` selector will respond when the mouse moves over an anchor element.

13. Let's make a pseudo class that makes the `<aside>` element invisible when we move the mouse over it.

```
aside:hover {display: none;}
```

Try `visibility: hidden;` instead

14. Using the properties `background-color` and `border-color`, add a pseudo class selector to change the background colour of the menu items to `lightblue` and the border to `darkblue` when the mouse moves over them.

Validate your HTML and CSS using the appropriate validators.

Check that your Webpage now looks similar to the style in the screenshot on the next page.

HTML 5 Basic Content Elements with some Simple CSS

[Heading and Paragraphs](#)[Special Characters](#)[Lists](#)[Tables](#)[Images](#)[Hyperlinks](#)

HTML page without CSS styling looks a bit boring!!

Headings and Paragraphs

Headings and paragraphs are BLOCK elements

Headings

There are 6 levels of heading. Headings are logical marked. Do not skip levels. If you want to change the size of a heading, don't change its level - change the CSS.

Paragraphs

Paragraphs contain the body text of a web page. You cannot put another block element inside a paragraph.

Special Characters

While HTML supports many named special characters like © (#174), XML only supports 5 namely: " ' & < and > . So to keep you HTML compliant you need to use numbers for all the others.

Lists

There are two types of lists: unordered and ordered

- First bullet point
- Second bullet point
- Third bullet point

1. First item
2. Second item
3. Third item

Lists can also be nested inside each other

1. First item
2. Second item
 - First sub-item
 - Second sub-item
 - Third sub-item
3. Third item

Tables

Table of Monthly Savings

Month	Savings		
	Salary	Interest	Total
Jan	\$60	\$40	\$100
Feb	\$40	\$40	\$80
Grand Total			\$180

Hyperlinks

This is a link to an external site [Blackboard](#)

This is a link to the [List Section](#) of the Web page

This is a link using [email](mailto:example@example.com) rather than an http protocol

Images

This thumbnail image is used as a hyperlink to a larger version of the image



Task 4: Apply Typographical Style to your Assignment index.html

If you haven't already done so, apply some appropriate typography to the first page of your assignment `index.html`.

Create a file called `style.css` in a folder called `styles` under `assign1` folder and link it to the `index.html` that you created in Task 1.

Based on the design you created in Task 2, apply some styling for the fonts (color, font type, size etc.).

Get your tutor to check your lab work.

Want some more practice with CSS?

See the "External Links" on Canvas.

Mozilla Developer Network has a CSS learning area:

<https://developer.mozilla.org/en-US/docs/Web/CSS>

w3schools has some good CSS tutorials and CSS exercises at:

<https://www.w3schools.com/css/>

Codecademy has some introductory CSS exercises at:

https://www.codecademy.com/courses/web-beginner-en-TlhFi/0/2?curriculum_id=50579fb998b470000202dc8b