Victoria University of Wellington

Te Whare Wananga o Te Upoko o Te Ika a Maui

# HTML Student Guide Trimester 1 - 2016

## Part One – Introduction to HTML and CSS

**INFO 101 – Foundations of Information Systems**  
School of Information Management  
Te Kura Whakaipurangi Korero

Contents

[HTML Student Guide Trimester 1 - 2016 1](#_Toc393378872)

[Part One – Introduction to HTML and CSS 1](#_Toc393378873)

[Introduction to INFO101 Workshops 3](#_Toc393378874)

[How do I prepare for workshops? 3](#_Toc393378875)

[Introduction to HTML 4](#_Toc393378876)

[Creativity 4](#_Toc393378877)

[Introducing HTML and CSS 4](#_Toc393378878)

[HTML Basics 5](#_Toc393378879)

[Tags 5](#_Toc393378880)

[Tag Nesting 5](#_Toc393378881)

[Attributes 6](#_Toc393378882)

[Values 6](#_Toc393378883)

[Comments 6](#_Toc393378884)

[Using a Text Editor 7](#_Toc393378885)

[Starting an HTML Document – Basic Outline 8](#_Toc393378886)

[DOCTYPE and HTML Tags 8](#_Toc393378887)

[Head and Body Tags 8](#_Toc393378888)

[Saving pages 9](#_Toc393378889)

[Opening your page to view 9](#_Toc393378890)

[Opening your page to edit 9](#_Toc393378891)

[Giving your webpage a Title 10](#_Toc393378892)

[Introduction to CSS – Basic Outline 11](#_Toc393378893)

[CSS Syntax 11](#_Toc393378894)

[CSS Location 11](#_Toc393378895)

[External Style Sheets 12](#_Toc393378896)

[Saving your External Style Sheet 12](#_Toc393378897)

[Using the ID and Class Selectors 13](#_Toc393378898)

[The Importance of Semicolons 15](#_Toc393378899)

# Introduction to INFO101 Workshops

INFO101 Workshops are designed to give you a practical knowledge of basic website design using HTML5 and CSS3.

## How do I prepare for workshops?

Many of you will be very new to HTML. It will be your responsibility to go through each workshop guide in order to prepare and plan well for your projects and the test.

If you prepare well, there is no reason why you will not be able to complete the projects and pass the test with flying colours.

Just make sure to check the course schedule to familiarise yourself with when the workshops are being held. They will not be held every week. However, the computer labs will be available to you at the times you have signed up, so use that time wisely.

# Introduction to HTML

The purpose of the HTML workshop material is to help you learn the fundamental code for creating a simple website. The idea here is not to build a ‘fancy’ online website but to have an understanding of how the foundations of such websites are built.

## Creativity

For your HTML projects, although you will be marked primarily on the technical skills behind your website, the more creative you are the more fun you’ll have. We encourage you to explore beyond the basics taught in the workshop materials!

## Introducing HTML and CSS

**HTML (HyperText Markup Language)** and **CSS (Cascading Style Sheets)** are programming languages used to create webpages. Using a combination of the two languages, you can format text, add graphics, sounds, video, animation and other forms of interaction to the webpages (and it’s the combination of these webpages which form a website).

You can use an ordinary text editor to create HTML documents that all web browsers can understand and interpret. HTML and CSS may at first seem to be difficult to understand, but in fact they’re very simple languages that only require some good practice. The hardest part about building webpages is not learning the code, but choosing the colours, text, images, and other elements of webpages to make them look attractive and readable.

HTML instructions, called tags, tell the internet browser **what** to display on the page. **HTML tags** describe the content of a document in detail. For example, tags tell the browser what the information is (i.e. a heading or paragraph text) and whether to display an image or a hyperlink. CSS instructions, called rules, tell the internet browser **how** to display the content. **CSS rules** are used to format the page’s text, the background colour or even can be used to lay out the information on the page.

Many people never notice the webpages’ source. When you surf the Internet you are viewing your browser’s interpretation of an **HTML source document**. To view the source document you can right click the on the page in the browser and choose *View Source*.

Although it is useful to view source code to look at layout, **copyright considerations cannot be ignored.** You should use other people’s pages for inspiration and as a learning device. You should never directly copy someone else’s work – this is plagiarism.  
  
***VIDEO: Introduction the Internet & websites:***<http://www.dontfeartheinternet.com/the-basics/not-tubes>

# HTML Basics

## Tags

Tags are commands in HTML that are written between **angle brackets: < >**. Tags instruct a browser to display page elements in a certain way. Most tags require an opening tag and a closing tag to execute the command properly.

An opening tag is written **<tag>**. A closing tag is written as **</tag>** (containing a **forward slash /** at the start). The content to be formatted is inserted between the two tags.

Text contained outside the opening and closing tags will not be affected by the commands. Tags should be written in **lowercase**. The following example shows how a portion of text can be shown as **bold:**

|  |  |  |
| --- | --- | --- |
| HTML Source Document:  Headings can be shown as <h1>headings</h1> |  | Page Viewed in Browser:  Headings can be shown as **Headings** |

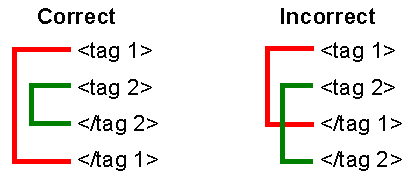
Opening Tag Formatted Text Closing Tag

### Tag Nesting

Tags can be nested within other tags. As a rule, **all tags should be closed in reverse order to how they were added**.

For example: <a><strong>text</strong></a>. So if you want to make some text **bold** and *italic,* it should be written <strong>Information <em>Systems</em></strong>. This would appear as **Information *Systems***.

When using multiple tags it is important to make sure that they are nested correctly. **Incorrectly nested tags are the cause of the majority of problems with basic HTML pages**. If you have overlapping tags, as shown on the right of the figure below, the best you can expect is that the document will be formatted in unexpected ways.



## Attributes

Tags sometimes also include commands called **attributes**. Attributes provide additional information about the HTML elements on your page. This additional information may include things such the address of your hyperlink or the location of an image you have inserted.

The attribute is entered as part of the opening tag (i.e. within <tag>). In some cases a tag can have a whole series of attributes, which are simply written one after another, with a space between each one.

**<a href=“http://www.google.com” hreflang=”en”>Click here!</a>**

Opening Tag Attribute Attribute Text Closing Tag

NOTE: The attributes are only listed in the opening tag; the closing tag never contains any attributes.

## Values

Finally, attributes always come in attribute/value pairs (attribute=“value”). Therefore when you write tags with attributes you always have to specify a **value**.

For example: <img src=“logo.jpg”>. In this example the attribute image source will tell the browser to display the logo.jpg image. A value can be a number or text. **Values must be enclosed in quotation marks (“ “) and should follow the equals ( = ) sign immediately** (no spaces) as follows:

**<tag attribute=“value”>**Content to be formatted**</tag>**

# Comments

While it's important to get the syntax written down correctly, it is also a good idea to write **comments** as you go along. Good comments will help remind you why you did something a certain way (or will help someone else out if they're reading your code without you there to explain it).

The comment tag is used to insert comments in the source code and is not displayed in the browsers.  
HTML comments look like this: 

CSS comments, on the other hand, look like this:



Remember: The computer does not look at the comments when figuring out what your HTML and CSS should do, but writing good comments is a good habit you want to pick up! Make sure to close your comments.

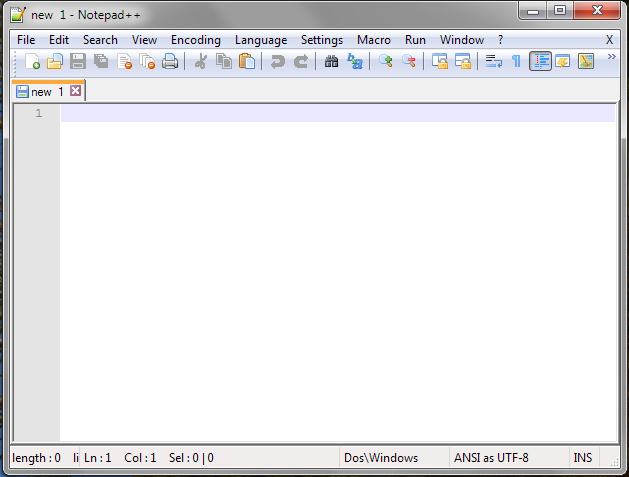
# Using a Text Editor

You can create an HTML document using any **Text Editor** such as Notepad or Text Edit. For the exercises we’ll do in these workshops, you can either use **Notepad** which is installed with **Microsoft Windows** by default or an alternative called **Notepad++** which provides more tools to help you to code HTML and CSS.

NOTE: If you are using a Mac there are a number of options available. The easiest option is using **Text Edit**, which is installed on Macs by default. There is also a program called **Text Wrangler** which is similar to Notepad++ and recommended.

#### Open Notepad++ now:

1. ****Click **Start** on the Windows taskbar. The **Start** menu appears.
2. Start typing “Notepad” and click **Notepad++** from the list that appears.
3. The following window should appear:



**Make sure you are familiar with this window as it is the program we will be using to write the webpages in all of the HTML workshops.**

# **Starting an HTML Document – Basic Outline**

Every HTML page has a basic outline that enables you to format your document correctly. There are two main parts.

## DOCTYPE and HTML Tags

At the beginning of every HTML document you need to tell the web browser what type of document it is reading. This is needed to ensure that the browser displays the page correctly. All that is needed is a single tag:

**<!DOCTYPE html>**

After the DOCTYPE tag and at the very end of every HTML page are the **HTML tags**. All of your HTML code needs to be between the opening and closing **<html>** tags.

**<html>**Website content**</html>**

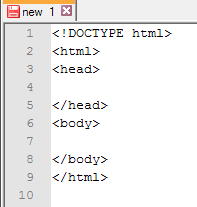
## Head and Body Tags

All HTML pages are broken up into two main sections – the head section and the body section. Each of these sections is defined by the associated tags: **<head> </head>** and **<body> </body>**.

The **Head** section contains important information about the document such as the page title and how it should be presented to things like search engines. It is also where you write the CSS to format your page. At the end of your head section, place a closing </head> tag.  
*Place the opening head tag <head> immediately after the opening <html> tag.*

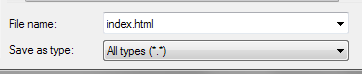
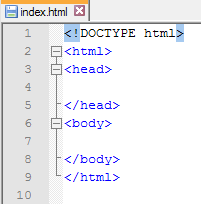
The **Body** section is where the majority of the contents of your page are contained. The <body> tag designates the beginning of the actual content. **Everything that is displayed and can be viewed on the web page will be enclosed in the <body> tags**.   
*Place the <body> tag* ***immediately*** *after the closing </head> tag.*

There should be no content in between the two sections! At the end of your document place a closing </body> tag directly before the closing </html> tag.



## Saving pages

By default, Notepad++ will save your document as a plain Text File (.txt). To be able to open the document in an internet browser, it must be saved as an HTML Document (.html).

1. Within your Notepad++ document click **File** on the Menu Bar. Click **Save As.**
2. Choose the location of where you wish to save. Then click the file name text box and type **index.html *(This is because the first thing webservers look for is the index page of a website).***
3. Change the **Save as type** to read **All types (\*.\*) (it’s at the top of the list).**
4. Click **Save**. Notice the title bar has changed to reflect the new name.  
   

**Note: It’s important to remember that your homepage must ALWAYS be named index.html**

## Opening your page to view

indexhtmllogo.pngThe **index.html** document now appears in your H Drive as a webpage icon (shown below). You can view the document in a web browser by double-clicking it.

When viewing your file as a webpage you can view the source (i.e. code) at any time. To do this simply right click anywhere on the webpage and choose **View Page Source.** However to edit this code, you must open it in a text editor as described below.

## Opening your page to edit

To open your webpage again in Notepad++ (for further editing):

1. Select the file on your H Drive (or USB Stick).
2. **Right-Click** on it and choose **Edit with Notepad++** from the menu that appears.

Alternatively:

1. Open Notepad++ from the Start Menu.
2. Choose **File** on the Menu Bar. Choose **Open...**
3. Change the type to read **All Files.** Select your file.

## Giving your webpage a Title

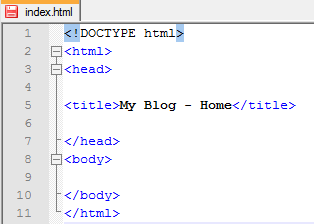
Every HTML page should have a title. The title appears in the browser’s **Title bar**, which appears at the top of the browser’s window. **Search engines, bookmarks** and **favourites** also use the page’s title so it should provide an accurate page description and perhaps a keyword, to increase the likelihood that web surfers searching for that topic will pick your page.

Titles are created using Title tags. These **Title** tags are placed within the **head section** of your document – remember the tips on nesting!

**Title Tips:**

* Only letters and numbers can be included in your title.
* A title cannot contain formatted text or images.
* Use a common element to start related pages: e.g. XYZ Company – Annual Report
* Your document can have only one title tag

1. Open your **index.html** in **Notepad++.**
2. On the line below your opening <head> tag type: **<title> My Blog – Home** **</title>**
3. Your document should look like the picture below:



1. Click on the **File** menu and click **Save** (or CTRL+S) to save the HTML source document.
2. If you your page you will see your title at the top of the window:



Title as it appears in a web browser

***NOTE:*** *If you simply write your title inside the head tags, and do not enclose it in the specified title tags, it will appear as text at the top of your webpage.*

# Introduction to CSS – Basic Outline

Now that you have a basic knowledge of HTML and how to create the backbones of a webpage, you can learn how to make the page look more attractive with formatting. To format your webpage you will use CSS or Cascading Style Sheets.

## CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations. CSS works by applying your rules to one or more **selectors**. These selectors can be HTML elements like the heading tags (h1 to h6), paragraph tags (p) or the body tags. The selectors can also be unique IDs for each item on your page. For example if you wanted a particular paragraph to be different to all of the other paragraphs, you would give it a unique ID and list this as a CSS rule. You will find out more about this later.



The **selector** is normally the HTML element you want to style (e.g. h1 or body).

Each **declaration** consists of a **property** and a **value** and is separated with a semi colon “**;**”.

The **property** is the style attribute you want to change. Each property has a **value** after the colon “**:**”.

## CSS Location

There are three types of CSS styles that can be used alongside a HTML document. There are: inline style, internal style, and external style sheets.

#### Internal (Embedded) Style

**Internal (embedded) styles** are placed inside the <head> section of the HTML structure using the style tag <style type=”text/css”> </style>. These styles can be used only for the webpage in which they are embedded. Therefore, you would need to create these styles over and over again for each webpage you wish to style.

#### Inline Style

Inline styles are placed directly inside an HTML element using the style attribute. This style can only be used for that one paragraph. It cannot be reused at all. This style is very inefficient.

#### External Style

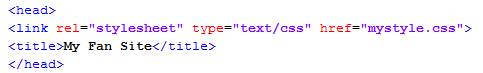
And external style sheet is a separate file that is linked to the web page. This particular style allows the user to reuse the style sheet as many times as the user needs as all the formatting is done on one sheet. We will go into more detail about External style sheets shortly.

***For this course we will only be using external style sheets.***

# External Style Sheets

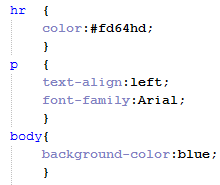
Styles are normally saved in external **.css** files. External style sheets enable you to change the appearance and layout of all the pages in a web site, just by editing one single file.

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire web site. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the <head> section next to the <title> tag.



**NOTE:** *mystyle.css* will be substituted with the name you have given to your own style sheet.

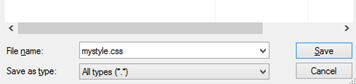
An external style sheet can be written in any basic text editor (not Microsoft Word or Pages). The file **should not** contain any HTML tags. Your style sheet should be saved with **a .css** extension.   
An example of a style sheet file is shown below:



### Saving your External Style Sheet

To save your style sheet as a **.css** file, you follow a similar process to saving your **.html** files:

1. Go to the File menu and click **Save As.**
2. Write the name of your file with **.css** at the end and select All types (\*.\*) from the menu. Then click **Save**.



***More HTML elements, CSS selectors and declarations will be covered in the upcoming guides.***

## Using the ID and Class Selectors

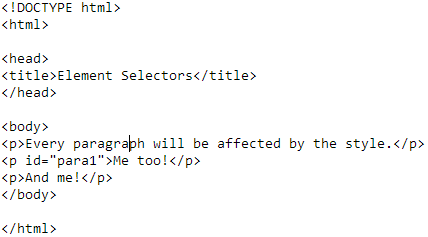
CSS selectors allow you to select and manipulate HTML elements. They are used to “find” or “select” HTML elements based on their id, classes, types, attributes, values of attributes and much more.

#### Element selectors

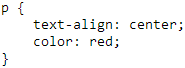
These are what you have been using so far and apply to every HTML element of a particular type. For example body, p, h1, h2. This is the most general and easiest to use because you don’t have to add anything to the HTML part of the code like with the other types.

For example:

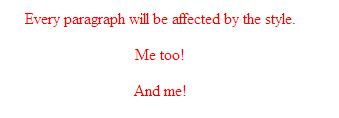
If you had a **HTML file** that contained the following code:



And a **CSS file** that contained this following code:

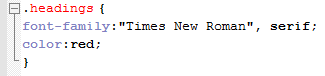


Then it would results in **web page** like:



#### Class selectors

These are the next level down and usually used to apply formatting to a group of objects on the page, for example all the different levels of headings or a group of paragraphs. To use the class selectors you write a **full stop** “.” before the name like this:

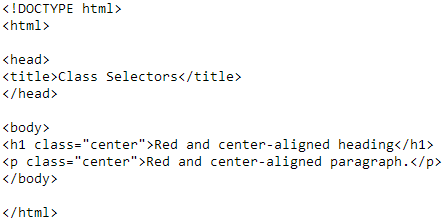


You then have to put the **class** attribute on all HTML elements that you want to include in the class like this:

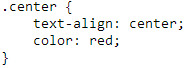


For example:

A **HTML file** containing:



And a **CSS files** containing:

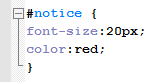


Would result in a **web page** like:



#### ID selectors

These are the most specific of the selector types. In fact there must only be one of each ID on a page. To use id selectors, you write **#** before the name like this:

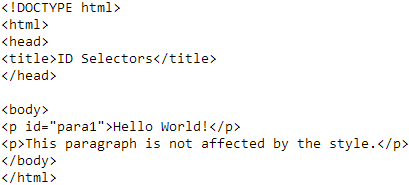


Similar to with the class selectors, you must add an **id** attribute to the HTML element you want to format:

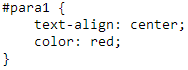


For example:

A **HTML file** containing:



And a **CSS file** containing:



Will result in a **web page** like:



The difference between an ID and a class is that an ID can be used to identify one element, whereas a class can be used to identify more than one.

Don’t worry if you don’t understand the difference at first, this is one of the most useful aspects of CSS but also one of the most difficult! However once you learn how to use the different selector types, it will make formatting your website much easier.

## The Importance of Semicolons

As you start adding more and more property-value pairs for each CSS selector, it's important to remember to put a **semicolon** (;) at the end of each line. The semicolon tells CSS that one property-value pair is over and it is time to move on to the next one.

Also, don't forget: all property-value pairs for a selector are surrounded by curly braces ({ }).

At the end, your CSS file should look something like this:

