

Chapter 4: Simple CSS

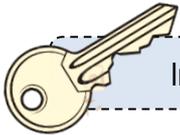
Learning Outcomes:

- ✓ Identify the three methods of applying CSS and the hierarchy in which they are applied
- ✓ Implement an external style sheet and use CSS class and id selectors
- ✓ Apply a range of CSS properties to a webpage



Prerequisite Knowledge:

- ✓ Complete Chapter 3
- ✓ Be able to use standard HTML elements, for example, headings, paragraphs, tables and lists
- ✓ Be able to hyperlink webpages together



Inline Style

Internal Style

External Style

Selectors

Keywords

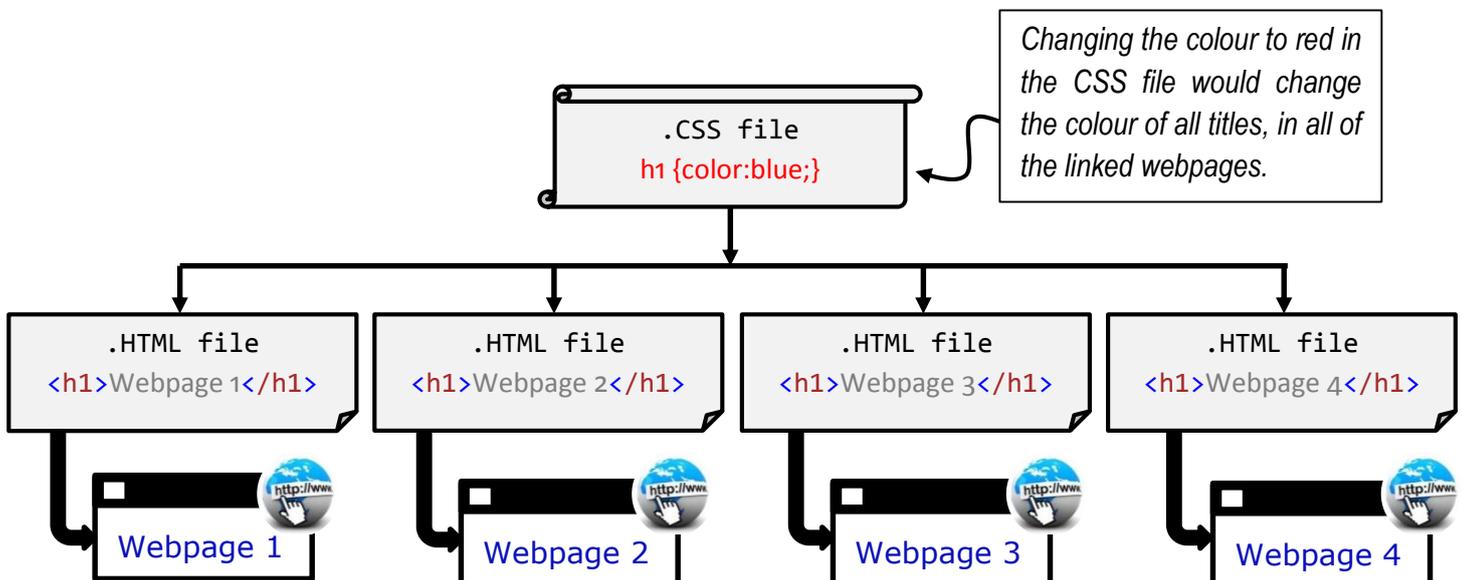
4.1 Theory: Introduction to CSS

The previous chapters implemented CSS to manipulate the appearance of several HTML elements; however, the CSS descriptions were purposely kept vague. In this chapter, however, CSS will be examined much more closely, hopefully enabling a much greater understanding of the 'realm of CSS', and more importantly, how it can be applied to modern-day websites.

The Role of CSS

As cited earlier, CSS is short for Cascading Style Sheets; the word 'cascading' refers to how CSS can be cascaded across multiple web pages to keep them consistent. CSS was introduced in the more recent versions of HTML; to be exact, HTML 4.0. CSS was introduced because it solved a major problem (and headache) in web design: granting the ability to manage website appearances with relatively little effort. Before the existence of CSS, HTML introduced elements such as `` and attributes such as `color` to manipulate the appearance of web pages. The problem with such techniques is that the same elements and attributes had to be embedded into every single web page, and in a large website, this was an exhaustive and expensive process.

Thankfully, the modern-day web has embraced CSS, a language used to inform a browser how to display web pages; put simply, the same CSS can be applied to multiple web pages. One advantage of CSS is that it can be placed in an external '.css' file; that file can be then be linked to as many web pages as necessary. The advantage of such practice is that a style can be edited in a single document, removing the need to edit individual web pages (and possibly hundreds of pages); the saved changes are then reflected in each linked web page. External style sheets save developers a lot of time and are supported by all browsers.



4.2 Practical: Applying CSS

In the earlier chapters, CSS was applied using two methods: either by adding CSS syntax to the `style` attribute (of a HTML element) or by adding CSS syntax between an opening and closing `<style></style>` tag, inside the `<head></head>` section of a HTML document. Both methods will be revisited shortly and explained in full before a third, more efficient, method is introduced. Just to clarify, there are three ways to apply CSS to a HTML document:

- ✓ Inline Style (using the `style` attribute)
- ✓ Internal Style Sheet (using the `<style>` tags inside the `<head>` tags)
- ✓ External Style Sheet (a separate file altogether)

Inline Styles

An inline style is applied to HTML elements by using the `style` attribute; although this is an acceptable means of applying CSS, it is not the preferred method and thus should be avoided as much as possible. Inline styles mix presentation with content (much like the old HTML `` tag) and thus lose many of the advantages available with the other two methods, especially the manageability of a document's style. Below is an example of an inline style:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Inline Style</title>
  <meta charset="UTF-8">
</head>
<body>
  <p style="color:pink;font-family:sans-serif;margin-left:40px;">
    This is an example of an inline style.</p>
</body>
</html>
```

This is an example of an inline style.



Activity 4.1

Create a webpage that includes three paragraphs; the paragraphs can be based on any topic. Use the web to research some new styles that you are yet to encounter and apply them to the paragraphs, ensuring that each has a unique style.



HTML Tips!

The basic structure of CSS is always the `property`, followed by a colon, and then the `value` followed by a semi-colon. There is also no need to have space characters preceding the colon or semi-colon.

```
color:pink;
```



HTML Tips!

In CSS, 'px' is a measurement that is short for pixels. When specifying a value in pixels, there is no space between the numeric value and 'px'. For example, '10 px' is considered wrong and '10px' is considered correct.

Internal Styles

An internal style sheet is applied by adding the CSS syntax between the opening and closing `<style></style>` tags, found inside the head of a HTML document. This method of applying CSS should generally be used when a HTML document has its own unique style (that is not necessarily consistent with the rest of the website). An example of an inline style is demonstrated below:

Activity 4.2

Create a webpage that features a heading, two paragraphs and an ordered list. Use an internal style sheet to apply some CSS properties to each element.



```
<!DOCTYPE html>
<html>
<head>
  <title>My Internal Style</title>
  <meta charset="UTF-8">
  <style>
    p{color:brown;font-family:serif;}
  </style>
</head>
<body>
  <p>This is an example of an internal style.</p>
</body>
</html>
```

My paragraph.



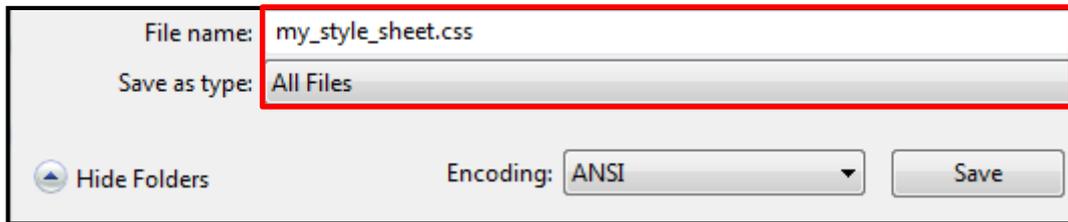
External Styles

External style sheets are the most common technique of applying CSS to web pages, and this is for good reason too, as a single external style sheet can be used to manage the styles of numerous web pages. A key advantage of external sheets is that they are very efficient; for example, the style of an entire website can be manipulated by changing just one document. To link web pages to a single style sheet, a link to the CSS file must be created in each web page. This is achieved by placing a `<link>` element in the head section of a HTML document. An example of such practice is shown below:

```
...
<head>
  <title>My External Style</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
...
```

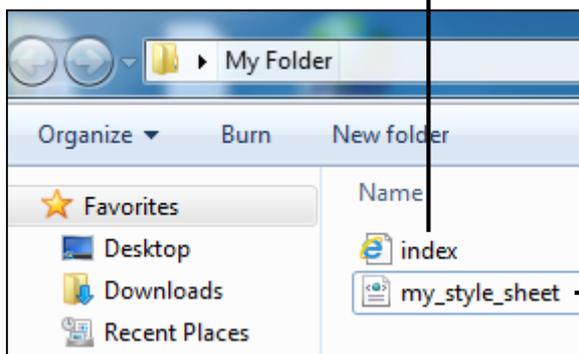
The `rel` attribute describes the relationship between the two documents; in this case, a style sheet. The `type` attribute defines the document type to the browser (CSS file) and the `href` attribute specifies the directory location of the style sheet that is to be linked.

An external style sheet can be created in a text editor similar to a HTML document, except by saving the file with the '.css' extension, as opposed to the '.html' extension. If using standard Notepad, ensure that the 'Save as type' is set to 'All Files', else the file will be saved as a text file and not a CSS file. An external style sheet should contain CSS syntax only; in other words, no HTML!



Below is an example of how an external style sheet can be applied to a HTML document:

```
<!DOCTYPE html>
<html>
<head>
  <title>My External Style</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <h1>My Webpage Rocks!</h1>
  <p>This is an example of an external style.</p>
</body>
</html>
```



```
p{color:sienna;}
h1{color:blue;}
```

In the 'index.html' file there is no presentational information; it is all stored in the 'my_style_sheet.css' file. However, because the index.html file is linked to the style sheet, all of the formatting information is applied to the index.html web page.

Activity 4.3



Create three webpages that demonstrate how to use an external style sheet (each webpage should be linked to the same CSS file). The webpages can be on any topic, but each must demonstrate a range of HTML elements (as learnt in the earlier chapters).



HTML Tips!

Numerous `<link>` elements can be added to a single HTML page to reference as many style sheets as necessary; all of the formatting properties in each linked CSS file would be applied to the linked webpage.

4.3 Theory: CSS Precedence

Multiple Style Sheets

As websites grow in size, if not managed correctly, CSS can be difficult to control. One of the most important rules to understand about CSS is the order of precedence. For example, if a HTML document has CSS properties in an internal style sheet and an external style sheet, which one would the browser use? The answer is both, but it is important to understand how the browser would apply both sets of properties, especially if both style sheets contain syntax that applies to the same element(s). Consider the CSS example below:

Internal Style Sheet Content:

```
p{
color:green;
font-size: 10pt;
}
```

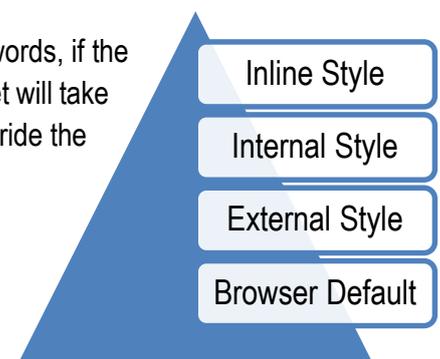
External Style Sheet Content:

```
p{
color:blue;
font-size: 12pt;
font-family: Arial;
}
```

If both of these style sheets were linked to the same HTML document, as both styles target the paragraph tag, one must consider which style the paragraph tag would inherit. The important lesson to note is that when working with CSS, the most specific rule is applied. For example, an internal style sheet is 'closer' to the document than an external style sheet, and furthermore, an inline style is more specific than an internal style. In this case, as the internal style sheet is 'closer' to the document, the paragraph tag would be green and have a font size of 10. However, as there is no 'font-family' rule in the internal style sheet, the 'Arial' font would be applied from the external style sheet. The CSS properties that would actually be applied are shown below:

```
p{
color:green;
font-size: 10pt;
font-family: Arial;
}
```

The diagram to the right summarises the CSS order of precedence (in other words, if the same element is targeted twice by different styles, it identifies which style sheet will take precedence). The method at the top has the highest precedence and will override the lower precedence CSS rules.

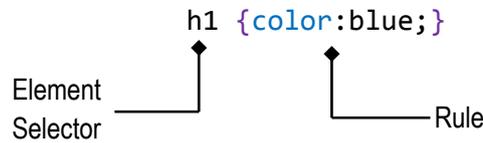


HTML Tips!

To override an internal style sheet, place the `<link>` tag (in the head section) after the internal style sheet (after the opening and closing `<style></style>` tags).

4.4 Practical: CSS Selectors

CSS is applied to a document by identifying the specific HTML elements (within the document) that the CSS rule(s) apply to; these elements are identified in the document by using 'selectors'. Below is an example of a CSS element selector. Element selectors identify the HTML element(s) that they will manipulate in the document; for example, all `<p></p>` elements could be formatted using this CSS syntax: `p{...}`. The illustration below will format all `<h1></h1>` elements in a HTML document to blue.



The above illustration is a simple example of a CSS selector; however, selectors come in various forms and thus parts of a document can be identified using other selecting types (not just by element). For example, parts of a document can also be identified using ids, classes, pseudo elements and pseudo classes.



HTML Tips!

When selecting HTML elements using an id selector, be aware that id names must not start with numbers!

The ID Selector

The id selector is used to identify a single, unique element in a HTML document. To achieve this, an id attribute is added to the HTML element (that is to be identified) and is set with a unique value. The CSS selector is then given the same name (as the id attribute value), but is preceded by a hash character (#). An id selector should only be used once in a HTML document, and thus id selectors are frequently used with divisions (`<div></div>`). This is especially true when building web page layouts, as the ids are used to identify the different sections of a web page (sections which should only occur once); for example, the footer, header, left navigation and main body. The table below demonstrates two examples of how id selectors can be implemented:



Activity 4.4

Amend activity 4.3 so that one of the CSS rules is applied using an id selector.

Example 1

CSS;

```
#footer{width:200px;}
```

HTML;

```
...
<div id="footer">
  This is my footer.
</div>
...
```

Example 2

CSS;

```
#mypara1
{
  color:blue;
  font-style:italic;
}
```

HTML;

```
...
<p id="mypara1">
  This is my paragraph.
</p>
...
```

CSS Classes

The class selector is used to identify any number of HTML elements. In CSS, a class can be applied to numerous HTML elements and they do not necessarily have to be of the same element type. To specify which HTML element is to be targeted by a CSS class, a class attribute is added to the opening HTML tag; the value of the attribute is then set to the same name as the CSS class. In a CSS document, a class is signified using the period character (.) before its name; for example, `.myClass`. Placing a HTML element tag before a class name will result in the class only being applied to that specific element type. Below are two examples of how a CSS class can be applied to a HTML document:



HTML Tips!

When selecting HTML elements using a class selector, be aware that, as with id selectors, class names must not start with numbers!

Example 1

CSS;

```
.myClass{width:200px;}
```

HTML;

```
...
<div class="myClass">
  This is 200 pixels wide.
</div>
<div class="myClass">
  This is also 200 pixels wide.
</div>
...
```

Example 2

CSS;

```
p.myClass2
{
  color:blue;
  font-style:italic;
}
```

HTML;

```
...
<p class="myClass2">
  This will be affected.
</p>
<div class="myClass2">
  This will not be affected.
</div>
...
```

Activity 4.5

Amend activity 4.4 so that the remaining CSS rules are applied using class selectors.



Grouping Selectors

Being able to group selectors will help speed up development; for example, if elements have the same style there is no need to write them out individually, simply group them together using a comma. For example:

This is the same as....

```
h1 {color:yellow;}
h2 {color:yellow;}
h3 {color:yellow;}
h4 {color:yellow;}
```

This...

```
h1, h2, h3, h4 {color:yellow;}
```



HTML Tips!

To comply with W3C, if the color property is defined, so should the background-color property.

4.5 Practical: Text

CSS provides several properties for formatting the appearance of text. One formatting property that has already frequently been used is the `color` property. This property is used to change the foreground colour of text (for example, `color:#000000;`). Below are some other examples of text formatting properties.

Activity 4.6

Amend activity 4.5 so that the text-alignment and letter-spacing properties are demonstrated.

Text Alignment

The text align property is used to define the horizontal positioning of text in relation to its surrounding container. Text can be aligned either to the centre, left, right or justified. A justified alignment stretches the text so that the left and right margins are even (similar to that of a newspaper column).

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Alignment</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <p class="one">I am centre aligned.</p>
  <p class="two">I am left aligned.</p>
  <p class="three">I am right aligned.</p>
</body>
</html>
```

I am left aligned.

I am centre aligned.

I am right aligned.

American Spelling

```
p.one{text-align:center;color:red;}
p.two{text-align:left;color:blue;}
p.three{text-align:right;color:brown;}
```

Letter Spacing

The letter spacing CSS property can be used to set the distance (in pixels) between each letter of text found a HTML element. The distance can also be set to a negative value to reduce the space between each letter. Below is an example of how the letter spacing property can be applied:

```
<!DOCTYPE html>
<html>
<head>
  <title>Letter Spacing</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <p class="one">I have a letter spacing of 3 pixels.</p>
  <p class="two">I have a letter spacing of minus 1 pixel.</p>
</body>
</html>
```

```
p.one{letter-spacing:3px;}
p.two{letter-spacing:-1px;}
```

I have a letter spacing of 3 pixels.

I have a letter spacing of minus 1 pixel.

Text Decoration

Text decoration is another CSS property used to manipulate the appearance of text; in particular, it is used to add or remove decorations (such as underlines). If the text-decoration property is set to none, it will strip the text of any decoration, including underlines applied to hyperlinked text. However, if the property is set to underline, it will underline the text (be aware that it is generally considered bad practice to underline non-hyperlinked text in HTML as it is confusing to users).

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Decoration</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <p class="one">I am decorated with an underline.</p>
  <p class="two">I am decorated with a line-through.</p>
  <p class="three">I am decorated with an overline.</p>
  <p>I am a <a href="#">hyperlink</a>with no decoration.</p>
</body>
</html>
```

```
p.one{text-decoration:underline;}
p.two{text-decoration:line-through;}
p.three{text-decoration:overline;}
a{text-decoration:none;}
```

I am decorated with an underline.

I am decorated with a line through.

I am decorated with an overline.

I am a [hyperlink](#) with no decoration.

Text Transformation

The text-transformation property can be used to specify the capitalisation of text; for example, upper case, lower case or the capitalisation of the initial letter of each word. Below is an example of how this property can be used:

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Transformation</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <p class="one">i am in capitals.</p>
  <p class="two">I AM IN LOWER CASE.</p>
  <p class="three">i have a capital le
```

```
p.one{text-transform:uppercase;}
p.two{text-transform:lowercase;}
p.three{text-transform:capitalize;}
```

I AM IN CAPITALS.

i am in lower case.

I Have A Capital Letter For Each Word.

Text Indentation, Text Shadow and Word Spacing

There are many other properties available in CSS that can be used to manipulate the appearance of text. Some other useful properties worth being familiar with are the text indentation, text shadow and word spacing properties. The `text-indent` property can be used to indent text from the left margin, by specifying the indentation size in pixels. The `text-shadow` property can be used to add a shadow to typed text to make it visually appealing and the `word-spacing` property can be used to specify the distance between each typed word. Below is an example of how these properties can be applied.

```
<!DOCTYPE html>
<html>
<head>
  <title>More Text Properties</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <h1>The Amazing CSS Properties</h1>
  <p class="one">I have been indented by 50 pixels :)</p>
  <p class="two">I have a fancy text shadow, like the title :)</p>
  <p class="three">I have a word spacing property set to 50
  pixels.</p>
</body>
</html>
```

```
p.one{text-indent:50px;}
h1, p.two{text-shadow:2px 2px #00FF00;}
p.three{word-spacing:50px;}
```

The Amazing CSS Properties

I have been indented by 50 pixels :)

I have a fancy text shadow, like the title :)

I have a word spacing property set to 50 pixels.



HTML Tips!

When setting the `text-shadow` property, the first argument specifies, in pixels, the distance of the shadow to the right. The second is the distance of the shadow downwards and the final argument is the colour (in hexadecimal).

Activity 4.7

Create a sports article web page; it can be on any sport of preference. The web page can have any appropriate content, but must feature a title, an image, a list, a table and three paragraphs. Use CSS properties to improve the presentation. The web page must demonstrate an external style sheet, the use of class selectors and several text decoration properties.

»» 4.6 Theory: Fonts

In previous chapters, the `font-family` property was used to specify the font face of displayed text. However, there are some important considerations to be aware of when working with CSS and choosing fonts. One such consideration when specifying a font face, is whether it is generic or specific; for example, the words 'Serif', 'Sans-Serif' and 'Monospace' are all generic types and do not refer to an actual font family (in other words, it instructs the browser to find a font of this style). However, 'Arial', 'Times New Roman' and 'Lucida Console' are all specific fonts. To help understand the differences between each available font style, the table below summarises each font:

Sans-Serif

F

Sans-serif fonts are fonts that do not have the small lines at the end of some of the characters. Examples of popular sans-serif fonts include Arial, Verdana and Helvetica.

Serif

F

Serif fonts have small lines at the end of some of the characters. They are also considered easier to read on a screen than sans-serif fonts. Examples of a serif font family, include Times, Times New Roman, Georgia and Bookman.

Monospace

F

Monospace fonts are font styles where each character has the same width; they are often used to display computer code or computer related text. Examples of popular monospace fonts include Courier, Courier New and Lucida Console.

When specifying which font-family to use (in CSS), it is considered good practice to specify several font names (separating each of them with a comma); for example, `font-family:"Times New Roman", Times;`. This is because if the first font is unavailable/unsupported by the browser, the next font family name (in the list) will be applied instead. When specifying the names, it is best to start with the most desired font, followed by a second desired font, before ending with a generic font type (for example, serif or sans-serif). Below is an example of the best considered practice when choosing font-family names; `font-family:Arial, Verdana, Helvetica, Sans-Serif;`



HTML Tips!

Note that if a font family name has spaces, it must be placed inside double quotes; for example, "Times New Roman".

4.7 Practical: Fonts

Font Sizes

In CSS, the `font-size` property is used to specify the size of text on a web page; however, there is a lot of consideration to be done when managing the size of text. Font sizes can be categorised as two types: relative or absolute. A relative size is based on its surrounding elements and allows a user to change the text size using a browser's built-in tools. Absolute sets a text to a specified size and does not allow a user to change the text size (and, therefore, is not considered disability friendly). An absolute text size can only be increased by zooming in on a web page (which also increases the size of the entire page, not just the text).

An absolute text size is set by specifying the value as a number, followed by 'px'; for example, `font-size:16px`, which sets the font size in pixels. A relative size is set by specifying a number too, but followed by the letters 'em'. The value '1em' is the default browser size and is equivalent to '16px' absolute pixels. Therefore, if a web designer requires font size to be 1.5x larger than normal, then 1.5em is used. If half the normal font size is required, then 0.5em is used. To convert the 'em' unit into pixels, multiply it by 16; therefore, 0.5em is the same as 8 pixels. An advantage of using the 'em' unit is that it allows browsers to resize the text for users who may have a visual impairment.

```
<!DOCTYPE html>
<html>
<head>
  <title>Font Sizes</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <h1>I am a title</h1>
  <p class="one">I am a paragraph</p>
  <p class="two">I am a paragraph</p>
</body>
</html>
```

```
p.one{font-size:16px;}
p.two{font-size:1em;}
h1{font-size:50px;}
```

I am a title

I am a paragraph

I am a paragraph



HTML Tips!

In web design it is considered bad practice to adjust the text size to make paragraphs look like headings or vice versa; appropriate HTML tags (i.e. `<h1>` or `<p>`) should be used.



HTML Tips!

The 'em' size unit is recommended by the W3C and is often used by developers to avoid resize problems that occur with older browsers.



HTML Tips!

For backwards compatibility with older browser versions, it is considered good practice to set the font size of the body to 100%. This removes resizing text issues that occur in older browsers.

```
body{font-size:100%;}
```

Font Style

The `font-style` property can be used to change the appearance of text. The most useful purpose is to italicise text, although the property can be set with three values: normal (no change to appearance), italic, and oblique (learning text, similar to italic, but is not supported by all browser versions).

```
<!DOCTYPE html>
<html>
<head>
  <title>Font Style</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="my_style_sheet.css">
</head>
<body>
  <h1>I am a title</h1>
  <p class="one">I am a paragraph</p>
  <p class="two">I am a paragraph</p>
</body>
</html>
```

```
body{font-size:100%;}
p.one{font-size:0.5em;}
p.two{font-size:1.5em;}
h1{font-size:5em;
font-style:italic;}
```

Activity 4.8

Amend the website that was created for activity 4.7 so that the text size property is applied using the 'em' unit. Also, apply a font style to a HTML element.

I am a title

I am a paragraph

I am a paragraph



Chapter Summary



- ✓ CSS is short for Cascading Style Sheets and is used to apply formatting to a HTML webpage.
- ✓ There are three ways in which CSS can be applied: inline, internal and external. Inline styles are 'closer' than internal styles and internal styles are 'closer' than external ones; this is the order of precedence (in which CSS properties are applied) if more than one style sheet is used
- ✓ An id selector can be used once to apply a style to a specific HTML element. When writing CSS id selectors, the hash tag is used; for example, `#footer{...}`.
- ✓ A class selector can be used to apply the same CSS properties to numerous HTML elements. When writing class selectors, the period is used; for example, `. myClass{...}`.
- ✓ The `text-align` property can be used to set the alignment of text in a webpage.
- ✓ The `letter-spacing` property can be used to adjust the space between each text letter.
- ✓ The `text-decoration` property can be used to add or remove decoration from text.
- ✓ The `text-transform` property can be used to transform text to upper case, lower case or an initial capital for each word.
- ✓ The `text-indent` property is used to indent text from the left margin.
- ✓ The `text-shadow` property is used to add a shadow around text.
- ✓ The `text-spacing` property adjusts the space between each word.
- ✓ Sans-serif fonts are fonts that do not have the small lines at the end of some of the characters.
- ✓ Serif fonts have small lines at the end of some of the characters.
- ✓ Monospace fonts are font styles where each character is the same width.
- ✓ In CSS, it is considered good practice to specify several font names when specifying the font family.