Getting Started With Website Design

4 – Cascading Stylesheets (CSS)

So far, we have created a website which is perfectly functional. It has information that is easily accessible to visitors. It looks pretty plain and boring though. As explained before, HTML is used to structure a website without specifying how it will look. If we want it to look good, that is where CSS comes in.

Cascading **S**tyle**s**heets allow you to specify styles that will *cascade* through your website. E.g. if you apply a style to the body element of your website, it will also apply to other elements such as headings and paragraphs that are inside that body element.

Defining CSS

CSS can be defined in three ways depending on where it is needed.

Linked Styles

A linked stylesheet is best when you want to create styles that will apply throughout your website. Using this method, a separate file with a .css extension is saved and each of the pages in your site that will use that stylesheet link to it. This provides consistency in your site as much of your layout and formatting can be defined from the one place which will then work on multiple pages. It also makes editing a lot easier as things only need to be modified in one place.

To make use of a CSS file, each of your HTML files will need to have a line similar to the following in the **<head>** section of your document.

This example will mean that the page will use style information that is contained in the file called styles.css

```
<head>

<title>Page Title</head>

<link href="styles.css" rel="stylesheet">

</head>
```

Embedded Styles

If you have styles that only need to be used on a single page, then they can be placed inside a style block within the head section of that page.

```
<title>Page Title</title>

<style type="text/css">

        Style information will go in here

</style>

</head>
```

Inline Styles

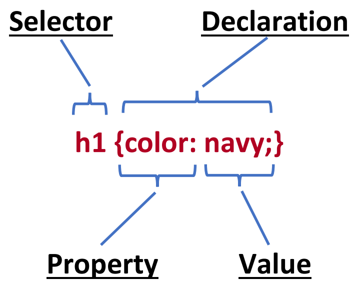
If you have style information that you only want to use in one specific part of your page, you can add it inside a tag using the style attribute. You already saw an example of this in one of the previous exercises where we added an inline style to give the contents of the col tag a yellow background.

```
<col span="2" style="background-color: yellow">
```

Style Rules

Each CSS file, embedded style block or inline style defines its styles by making use of **Style Rules**.

A Style rule consists of 2 parts. The **Selector** determines what the style rule will be referring to while the **Declaration** contains the style information for that selector. Look at the example below.



In this example, the **Selector** (h1) specifies that the style rule refers to any h1 elements in the document(s) that are affected (all of the level 1 headings).

The **Declaration** {color:navy} specifies that any part of the document matching the style selector will be formatted in navy blue. It consists of a **Property** (color) and a **Value** (navy) which are separated by a colon **:**. Declarations end with a semi-colon **;** and are enclosed in curly brackets **{ }**.

Combining Selectors

When there are several selectors that will have the same declaration, they may be grouped together with each selector separated with a comma. For example, if your level 1, 2 and 3 headings will all be formatted with a navy-blue colour, then instead of having a separate style rule for each one they can be grouped into a single rule as shown below.

```
h1, h2, h3 {color: navy}
```

Combining Declarations

Declarations can also be grouped together using CSS shorthand. For example, if your headings are going to be bold and formatted in Arial font, you can group the declarations into a single rule as shown below.

```
h1 {
  font: bold Arial;
}
```

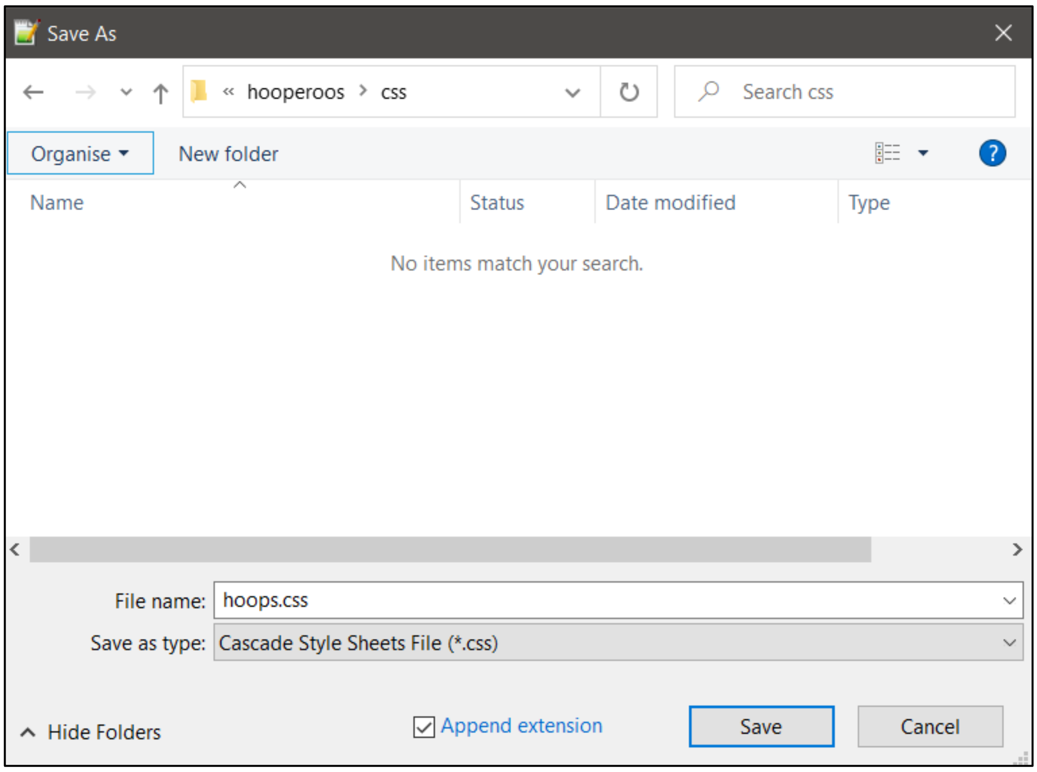
Conditional Selectors

If a style rule contains more than one selector separated by spaces, it means that the last selector will only apply if it is within the ones before it. In the example below, paragraphs will be formatted with small font size, but only if they are within the footer section.

```
footer p {
  font-size: small;
}
```

Exercise 1 – Creating a Stylesheet

1. Create a new blank file in your editor.
2. Save the file with the filename **hoops.css** (depending on your editor, it might be necessary to select Cascading Stylesheet as the file type as shown in the Notepad++ example below. Make sure the file is saved inside the **css** folder of your website.



No items match your search.

File name: hoops.css

Save as type: Cascade Style Sheets File (*.css)

☑ Append extension

Since it is an empty file, there isn't any style information in there yet, but we will link to it in our html files. That way, when we start adding style rules to our css file, the pages will already be able to use them.

3. Open any one of your html files and add a link tag like the example below. It can go anywhere inside the head section of the document. Note that the css folder name has been placed in front of the filename. This assumes that you have saved your stylesheet file inside the css subfolder.

```
<title>Morley Hooperoos Basketball Club Memberships</title>

<link href="css/hoops.css" rel="stylesheet">

</head>
```

4. Copy that line into the head section of your four other html files and make sure each one is saved.

Exercise 2 – Setting up the CSS File

Now that our pages are linked to our CSS file, we are ready to add CSS rules to our CSS file.

CSS rules don't have to be written in any particular order. In fact, like html, they don't even have to be written on separate lines. It is much easier to edit your css, however, if you place your CSS rules in a logical order and use spacing and line breaks to make it easier to read.

Comments can also help to organise your CSS. Comments in CSS begin with **/*** and end with ***/**. In our website, we will use comments to divide our css rules in to logical categories which indicate where they will be used.

1. Open your CSS file in your editor and add the following comments.

```
/* General Page Layout */



/* Typography */



/* Header */



/* Navigation */



/* Main */



/* Footer */
```

Note Just like comments in html, these won't actually do anything in our web page, but they will help us to be organised. When we create a new style rule, we can put it under the appropriate comment along with other similar style rules. Just as if the comments were headings.

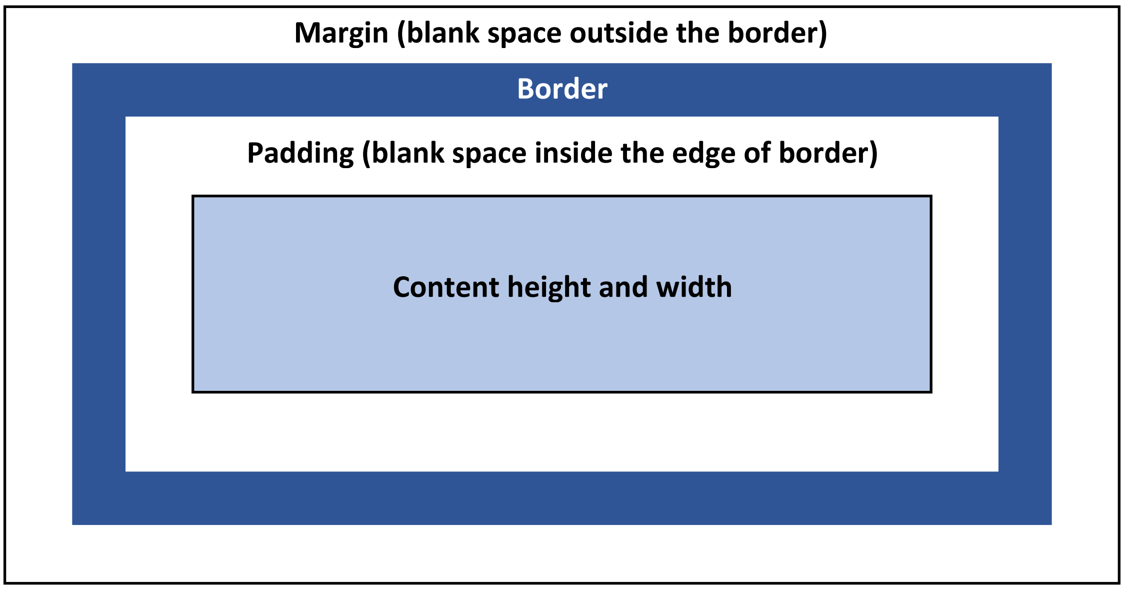
Note If you have CSS rules that contradict each other, for example a rule that says headings are blue and another rule saying headings are green, the last of the contradicting rules will be used.

The CSS Box Model

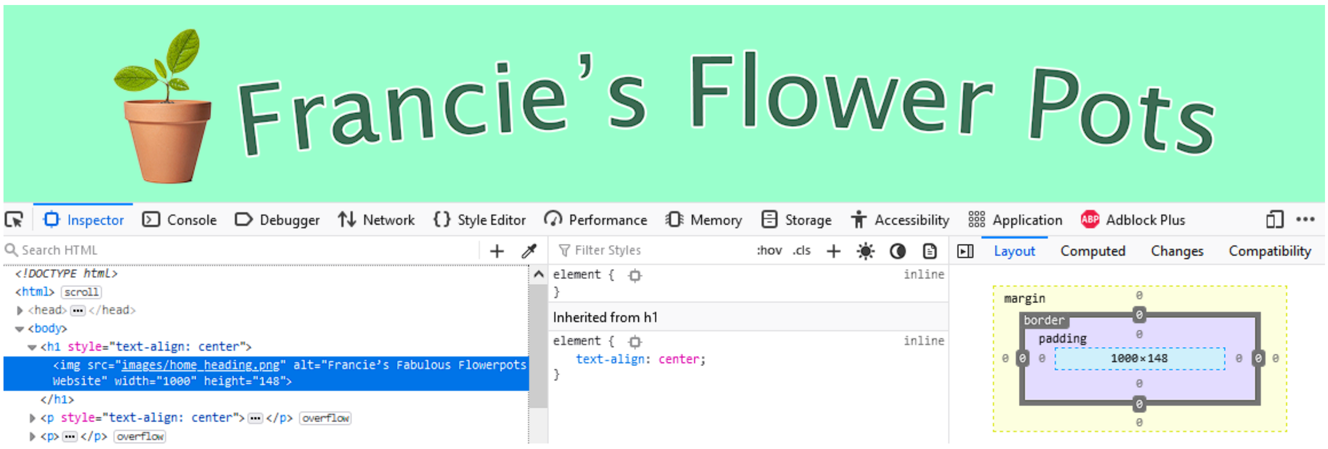
In an earlier section we described how many HTML elements such as <h1> and <p> are block elements. CSS can be used to style block elements using what is referred to as the box model. Each element can be modified using the following measures.

- **Height** and **Width** of the actual content box.
- The **Border** surrounding the box including the thickness or visibility of the border lines.
- **Padding**: The blank space between the border and the content
- **Margins**: Blank space outside the border, or space between the element and surrounding elements.

The diagram below illustrates how this works.


Margin (blank space outside the border)
Border
Padding (blank space inside the edge of border)
Content height and width

In most modern web browsers, you can right-click on part of a page and select an option to **Inspect** that element. This provides a lot of information about the selected element including the HTML code for that element and related CSS properties. Some browsers will also show the box model for the selected element. The example below shows the information Mozilla Firefox provides when we inspect an image on a page. Note the box model on the right side of the inspect details.



```
<!DOCTYPE html>
<html> scroll
  <head> ••• </head>
  <body>
    <h1 style="text-align: center">
      <img src="images/home_heading.png" alt="Francie's Fabulous Flowerpots
      Website" width="1000" height="148">
    </h1>
    <p style="text-align: center"> ••• </p> overflow
    <p> ••• </p> overflow
```

```
element {                    inline
}
Inherited from h1
element {                    inline
    text-align: center;
}
```

```
margin 0 0 0 0
border 0 0 0 0
padding 0 0 0 0
1000×148
```

In our website, we will start by defining CSS for some of the main blocks on our page. I.e. the **<html>** element (the whole document) and the **<body>** element (the page within the document).

Exercise 3 – Creating Our First CSS Rule

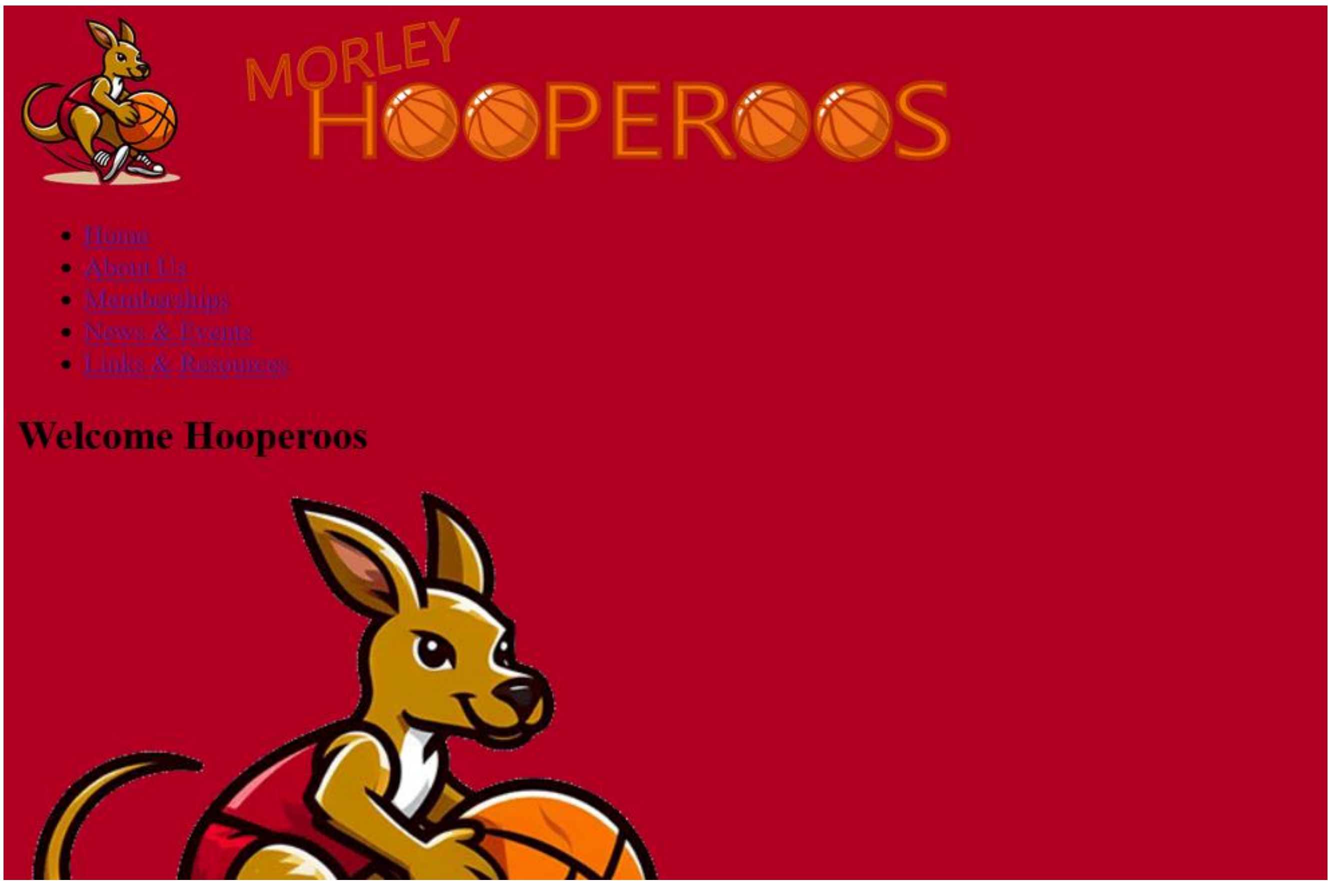
We'll start by creating a style rule that will change the background colour of the **<html>** element, which will affect the entire document.

1. Add a new CSS rule like the following. For the sake of organisation, place it under your General Page Layout comment/heading.

```
/* General Page Layout */

html {

        background-color: #B10022;

}
```

In this style rule the selector is **html**. This means it will apply to the *html* tag in any pages that refer to this css file. The declaration changes the background property. The value of that property is to the right of the colon with a semicolon completing the rule.

2. Save the CSS file.
3. Preview one of your html files in your web browser. If the html file is correctly linked to your css file, the web page will now have a coloured background.



- Home
- About Us
- Memberships
- News & Events
- Links & Resources

Welcome Hooperoos

It's a little difficult to read text on the background at the moment but we will add other CSS rules later to help with that.

CSS Reset

Most web browsers add a default amount of spacing around elements in our web page. This can vary between browsers. We will start by adding some style rules that set that spacing to 0 to over-ride the browser defaults. We will then be free to set spacing for specific parts of our page without the browser defaults interfering. Using a ***** for a css selector means that it will apply to everything on the page. We can then create other css rules that will over-ride this spacing for specific parts of the page.

4. Add a new section under the *General Page Layout* comment as shown below (above the html css rule).
5. /* General Page Layout */

```
* {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
}
```

6. Preview the changes to your page to see the difference with the adjusted spacing. It will probably look like everything is too close together but we will fix that with other rules.

Colours in CSS

Colours can be described several ways in CSS.

Names

For common colours you can use their names. E.g. red or blue. Below is a list of 16 colour names that are recognised by all web browsers. Newer browsers can recognise even more names.

Black	White	Gray	Silver
Maroon	Red	Purple	Fuchsia
Green	Lime	Olive	Yellow
Navy	Blue	Teal	Aqua

Tip You can view a list of colour names that will work in CSS from the Worldwide Web Consortium. This list also shows the hexadecimal value and RGB value next to each colour.

Hexadecimal Codes

In computing, numbers are commonly represented using hexadecimal notation. This same notation can also be used to represent colours in CSS. E.g. #000000 for white or #ffffff for black. This includes the many thousands of colour shades that cannot be represented by colour names in CSS. Many graphics applications such as Adobe Photoshop will show the hex code of a colour while you are selecting colours as shown on the next page.

RGB Values

The pixels that make up an image on your computer screen can be represented as a combination of Red, Green and Blue light. Colours are often defined by using RGB numbers to specify the amount of red, green and blue that combine to make a colour. For instance the colour red would be rgb(255, 0, 0) where it has the maximum amount of red (255) with zero green and zero blue.

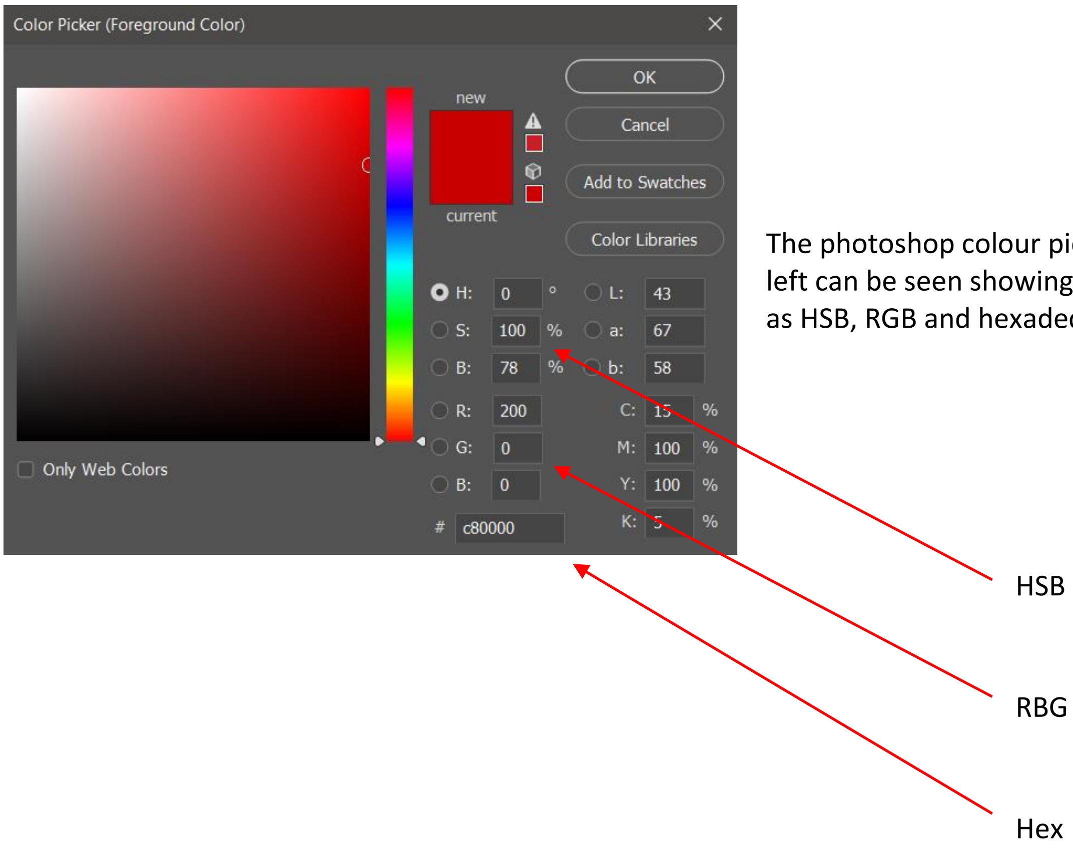
Hue Saturation Brightness

Colours can also be defined by three numbers representing the hue, saturation and brightness values.

Tranparency

RGB and HSB can both have a 4th value that represents the transparency level of a colour (referred to in many graphics software as an alpha channel).

Note When you change colours in CSS you must use the USA spelling or it will not work. E.g. *background-color* will work while *background-colour* will not work. The same applies for alignment where *center* will work but *centre* will not. Also, the colour *gray* instead of *grey*.



H: 0 °
S: 100 %
B: 78 %
R: 200
G: 0
B: 0
# c80000
L: 43
a: 67
b: 58
C: 15 %
M: 100 %
Y: 100 %
K: 5 %

The photoshop colour picker shown to the left can be seen showing the same colour as HSB, RGB and hexadecimal.

Exercise 4 – Creating CSS Rules for Page Layout

Now we will add a style rule for the <body> element on the page. This will include declarations for how wide the body will be on the page as well as the background colour the body section will use.

1. Add a style rule for the body selector beneath the style rule you already created like the following.

```
html {
        background-color: #B10022;
}
body {
 background-color: rgba(255,255,255,0.90) ;
 width: 80%;
 max-width: 1200px;
 margin: auto;
}
```

2. Save and preview the changes.

- The first declaration sets the background colour to a white colour which is 90% opaque (or 10% transparent) so the background colour for the page partially shows through.
- The second one sets the body to only take up 80% of the available width.
- The max-width property means that on particularly large screens, the width will be no more than 1200 pixels.
- The last one sets the margin following the box model discussed earlier. According to the box model, we can define an object's border, an object's padding (blank space inside the border) and an object's margin (blank space outside the border). This declaration specifies that the blank space for the margin on either side will be evenly shared with equal amounts of space on the right and left, so the body will end up in the center of the screen.

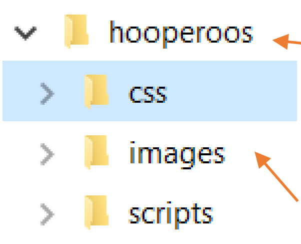
Background Images

Background images can look good when used effectively. An important consideration is to make sure that using a background image doesn't make text difficult to read. We will use a background image for our HTML. The slightly transparent colour for the body will allow the background to show through slightly, while still leaving our text readable.

3. Add a background image declaration to the html rule as shown below.

```
html {
   background-color: #B10022;
   background-image: url(../images/main-court.png);
}
```

Note the referencing. The two dots mean the folder above the current one. Since the css file is in a css sub folder and the image is in an images sub folder, the reference needs to go up one level to the main folder, then down into the images sub folder.

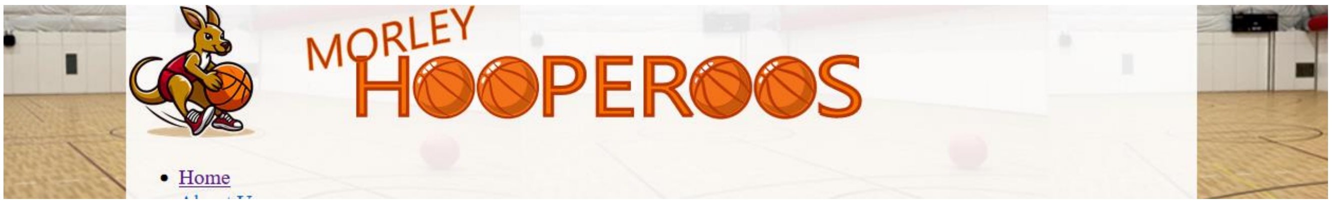


../ means up one folder level

images/ means down to the images folder

4. Save the css file and preview your html file

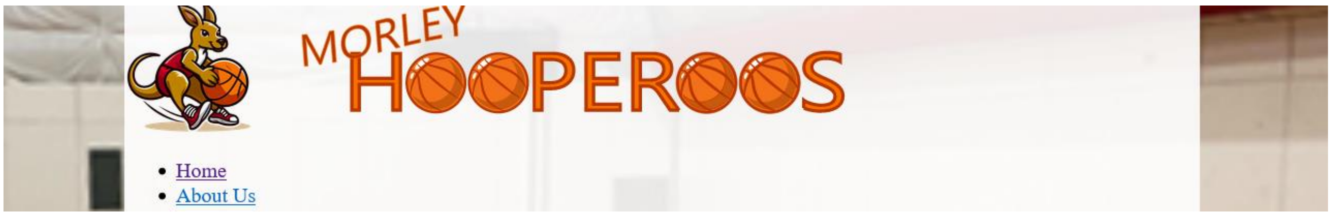
By default, background images tile and repeat which might not look good with our current page. Fortunately, CSS gives us plenty of options for background images.



5. Add additional declarations to the html rule as shown below. Preview the changes.

```
  background-size: cover;
  background-attachment: fixed;
  background-position: center;
}
```

- background-size – this can be set in measurement units, but it can also be set to **cover**. This means the image will stretch as needed to cover the screen area.
- background-attachment determines what the background image will do when a user scrolls down the page. The fixed option means the background will stay in place instead of scrolling with the rest of the page.
- The last rule ensures that the background image is centred on the screen.



Note In this case we have used a smaller image which looks pixelated or blurry as it is displayed at a larger size. You can avoid this by using a larger image or using a vector (svg) image.

CSS Measurement Units

Now that we've seen a couple of different ways of specifying size and distance in CSS, we'll have a quick look at some of the numerous ways that CSS allows you to measure things. Measurement methods can be categorised as absolute and relative measurements.

Absolute Measurements

These measurements use standard units of measurement to specify exact size.

Unit	**Description**
cm	Centimetres
mm	Millimetres (10 mm in each cm)
in	Inches
pt	Points (72 points in an inch)
pc	Picas (12 points in a pica)
px	Pixels (the dots on a screen)

Relative Measurements

These measurements can vary depending on factors such as the size of the screen that is being used to view your website. Some, such as percentage % are easy for most people to understand but others might be new. Relative measurements can be useful for websites where visitors to your website will have screens of different sizes that they will view your site on.

Unit	**Description**
%	Percentage of the available space or parent element. E.g. this could refer to % of screen width
rem	Relative to the size of the root element. We will use this shortly to set text sizes.
em	Relative to the size of the parent element. For example if the size for the body section is set then elements inside the body could use em to set sizes in relation to that parent size.
vw	Viewport width. This sets the size as a portion of the viewable screen width. E.g. 10vw means 10% of the screen width.
vh	Viewport height

Exercise 5 – Set up Typography for Our Site

We will create a new CSS rule for the html element in our typography section. We could simply add to the html rule we already have in our css but putting text specific rules together in their own section can make it easier to find parts of the css that you need to edit.

First, we will define the Font Family. This means the font style used for text, such as Times New Roman or Arial. Font names that consist of more than one word (such as Times New Roman) must have quotation marks around them.

Note Not everyone has the same fonts on their computer. If you specify a font that people might not have on their computer, you can specify alternative fonts to use. If a person using your computer doesn't have a font, then their computer will use the next font if available. It is common practice to specify a generic font as an alternative.

1. Add a new rule for the html selector in the typography section of your CSS file.

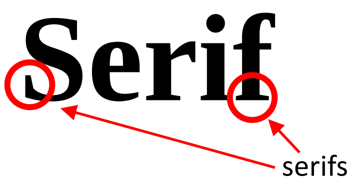
```
/* Typography */

html {

        font-family: "Segoe UI Emoji", Arial, Sans-serif;

}
```

This will mean that text in our html document will use the Segoe UI Emoji font. On computers that don't have that font, they will instead use Arial which is a more common font. On computers that don't have either one, they will use whatever font is their system's default sans-serif font.

Serif vs Sans Serif

You may have seen the words *Serif* and *Sans* in font names. If a font has serifs, it refers to the small decorative bits on the end of lines in the characters. If a font is sans serif (sans = latin for "without") then it means that font doesn't have serifs.

Serif Font



Sans Serif Font

Now we will adjust the size of our fonts. We will do this by using an absolute measurement (pixels) to set a baseline size for fonts in our site. We will then use a relative size (rem) so that for different parts of our site the text size will be set in relation to the base size.

Add a new declaration to our html rule to sets 10 pixels as the base height for our text. This is a pretty small font size, but we will use that as a baseline to adjust the size for text in different sections of our site.

```
html {

   font-family: "Segoe UI Emoji", Arial, Sans-serif;

   font-size: 10px;

}
```

Now we can create new style rules for different elements in our site that each set the size in relation to the baseline html text size.

2. In the typography section, add the following rules for our different types of headings.

```
body {
   font-size: 1.8rem;
}
h1 {
   font-size: 2.5em;
}
h2 {
   font-size: 1.7em;
}
h3 {
   font-size: 1.2em;
}
```

3. Save and preview the changes. You can easily adjust sizes in your CSS if you think the text is too big or small.

We now have a rule which says text in the body of the page it 1.8 times as big as our 10 pixel base font (meaning text that is 18 pixels high). **h1** which is 2.5 times as big as the size it inherits from the body element (2.5 times as big as 18 pixels high. For **h2** headings they will be 1.7 x 18 pixels high and **h3** headings will be 1.2 x 18 pixels high.

4. To add a bit of uniformity in our headings, we'll also add a css rule that applies to all of them.

```
h1, h2, h3 {
   color: #B10022;
   font-weight: 900;
}
```

Font weight makes the text bold. You can specify font weight as a word (*normal, bold, bolder, lighter*) or as a number from *100* to *900*. Normal text is usually about 400.

5. We'll add more typography changes later, but we'll add one last one for now. A declaration to make our level 1 headings centre aligned (remember to use the USA spelling). Modify the h1 rule to add an alignment declaration like the following. Text alignment has options including left, center, right or justified.

```
h1 {
  font-size: 2.5em;
  text-align: center;
}
```

Note that if you preview other pages in your site, they will have a consistent look since they are all using the same CSS file. Click the navigation links to the other pages to test them.

Exercise 6 – Floating Elements

HTML elements can be positions using CSS in several ways. One common way is to use the *float* property We will demonstrate this with some css rules in our *Main* css section.

Currently each page has an image which sits on the side of the page with text continuing below it. We will add a css rule which changes the image so that it "floats" on the side of the page with text flowing or wrapping around it. We will do this by creating a css class. This allows us to create a style rule which can be added to html tags wherever needed. CSS classes start with a dot, followed by the name you want to give to the class. We will create a new css class called *image-right* that will float an image to the right with a small amount of margin space around the image. Later we will add some other css declarations to the same class to control the image size.

1. Add the following rule in the **main** section (be sure to include the dot in front of the class name).

```
/* Main */
.image-right  {
  float: right;
  margin: 2px;
}
```

This css rule won't do anything yet since the selector is the name of a class and we need to add an attribute to any tags that need to use it.

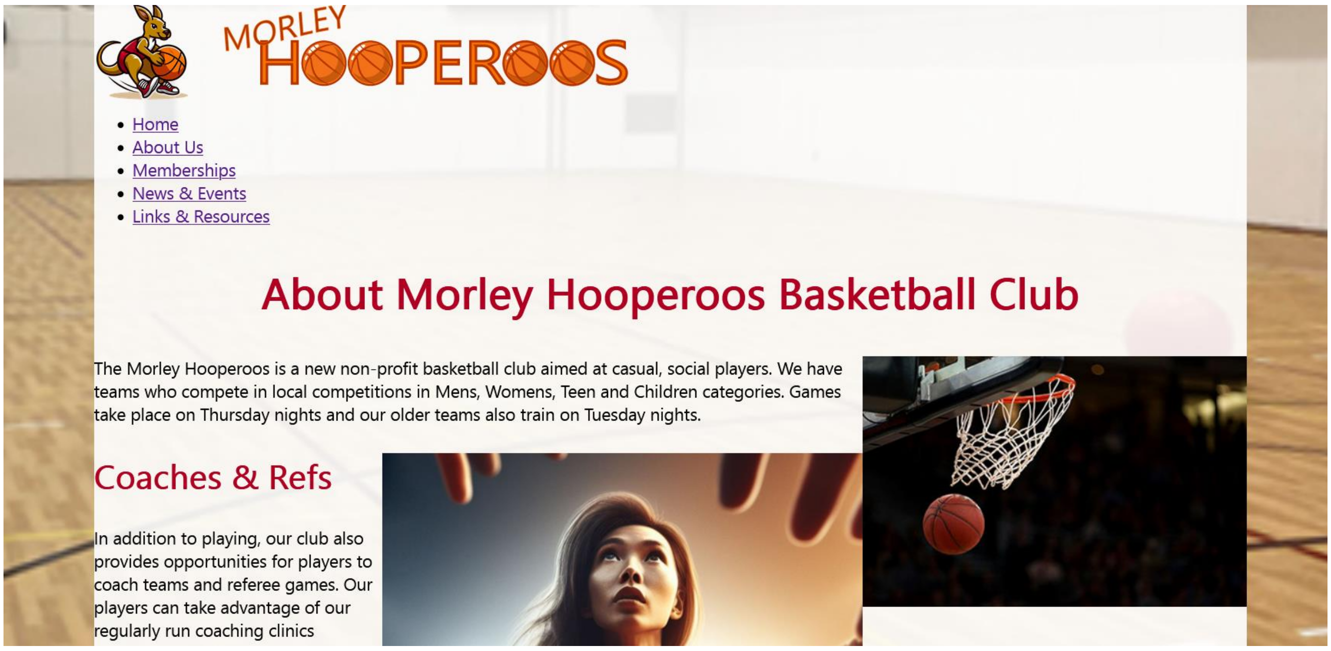
2. Go to the html for the index page and locate the image under the level 1 heading. Add an attribute **class="image-right"** to the image tag like the example below.

```
<h1>Welcome Hooperoos</h1>
        <p> <img src="images/main_logo.png" class="image-right" alt="Morley Hooperoos Logo"
        title="Morley Hooperoos Logo" width="400" height="400">Welcome to the Website of the
```

Now after saving both the css and html file, if you reload the page, the image will now appear on the right with text wrapping around it.

3. Go to each of the other html pages and add a **class="image-right"** attribute to the images in the main section of each page. For pages that have 2 images, add the attribute to both.

On the pages that have 2 images, you might notice that the float property means that the second image is sitting next to the first one rather than sitting beneath it. If we would prefer that this didn't happen, we can make use of the clear property. The clear property clears any floats from the left margin, right margin or both margins.



- Home
- About Us
- Memberships
- News & Events
- Links & Resources

About Morley Hooperoos Basketball Club

The Morley Hooperoos is a new non-profit basketball club aimed at casual, social players. We have teams who compete in local competitions in Mens, Womens, Teen and Children categories. Games take place on Thursday nights and our older teams also train on Tuesday nights.

Coaches & Refs

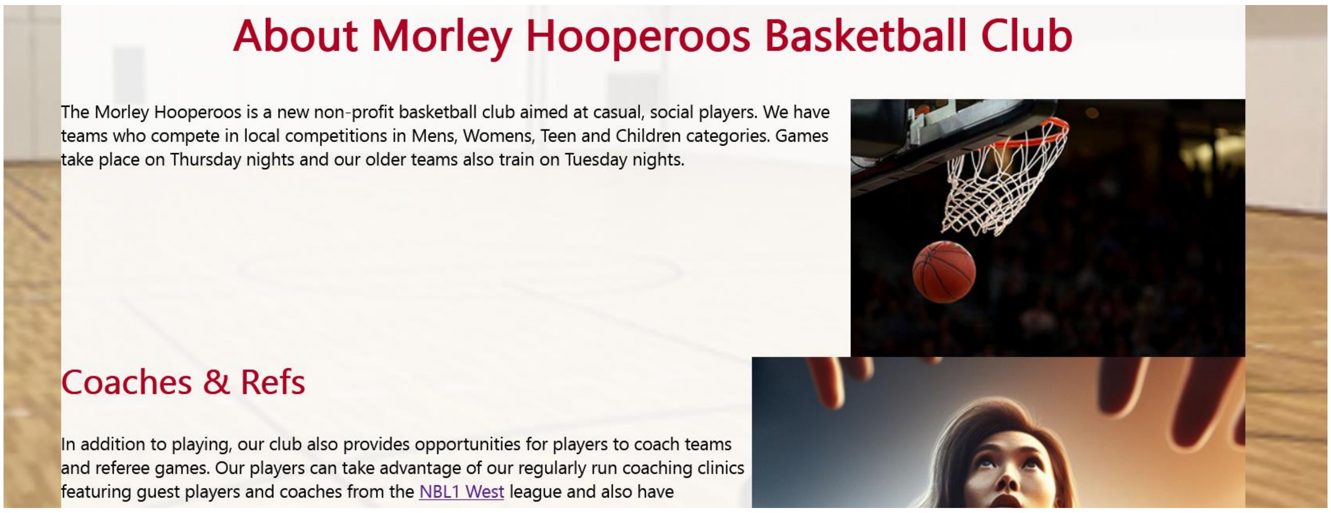
In addition to playing, our club also provides opportunities for players to coach teams and referee games. Our players can take advantage of our regularly run coaching clinics

We will add a new declaration to make level 2 headings start on a new line with no image floating.

4. Add a declaration to the h2 selector like the following.

```
h2 {
  font-size: 3.5rem;
  clear: both;
}
```

Since level 2 headings will no longer be wrapping around images, this should fix the problem of images sitting next to each other.



About Morley Hooperoos Basketball Club

The Morley Hooperoos is a new non-profit basketball club aimed at casual, social players. We have teams who compete in local competitions in Mens, Womens, Teen and Children categories. Games take place on Thursday nights and our older teams also train on Tuesday nights.

Coaches & Refs

In addition to playing, our club also provides opportunities for players to coach teams and referee games. Our players can take advantage of our regularly run coaching clinics featuring guest players and coaches from the NBL1 West league and also have

Exercise 7 – Image Sizing

When we first placed images in the document, we put the size information in the img tag. We can also use CSS to set the size of images. A benefit of doing it this way is that we can use CSS to resize the image. Such as making it smaller when the page is viewed on smaller screens.

1. On each of the pages, locate the first image in the main section.

```
<img src="images/main-hoop.jpg" class="image-right" alt="News / Events" width="400" height="282">
```

2. Now in each of those image tags, add an id attribute to give the image the name *main_image*. This will give that page element (in this case an img tag) a name that can be referred to where needed in css as well as programming scripts.

```
<p><img src="images/main-hoop.jpg" class="image-right" id="main_image" alt="News / Events"
width="400" height="282">
```

Now to add a new rule in our css file. In css we refer to an id by using a selector with a hashtag followed by the id name.

3. Add a new selector for the main_image id below the class we added previously.

```
.image-right  {
  float: right;
}
#main_image {
  width: 400px;
  height: auto;
}
```

4. Save all files and preview the changes. Now each of the main images will be the same width and we can easily use css to change the width. Setting the height to auto will automatically set the height in proportion to the

We can also use css to set the image size according to the size of the screen being used to view the web page.

5. Change the rule you just created so it looks like the following.

```
#main_image {
  max-width: 400px;
  height: auto;
  width: 35vw;
}
```

6. Preview the changes.

The width of the image is now set to 35% of the viewing area. Initially you might not see a difference but if you resize your web browser window, once it is small enough for 35% of the width to be less than 400 pixels, then the image will resize accordingly.

Note You can use the same ID name for elements on different pages. Just don't use the same name more than ones on the same page.

Exercise 8 – Hyperlink CSS Formatting

CSS can be used to create special formatting for hyperlinks and can be done in more than one way. As we have already seen, hyperlinks are created using the anchor (a) tag. So we could create a css rule for the **a** selector which formats all links. However, links can show differently depending on the link being in one of several states. Each of these states can be formatted separately using an anchor "pseudo class".

Link state	Selector	Meaning
Unvisited	a:link	Selects all links to locations that the user hasn't already visited (blue by default)
Visited	a:visited	Selects all links to locations that the user has already visited. I.e. Locations that are in the browser history (purple by default)
Active	a:active	Select links that are active. E.g. a link that a user has clicked on (red by default)
Hover	a:hover	Selects links that the user has moved their mouse over
Focus	a:focus	Selects links that have focus. I.e. selected using the tab key on the keyboard

We will create one rule using an **a** selector that sets certain properties for all of our links. Then we will create another rule that applies properties to links when a mouse hovers over them or when the link gains focus (such as from the tab key on the keyboard or pressing it on a touch screen).

1. Add the following css rules in the typography section (the comments are there to explain the rules – you don't need to include them)

```
a {
 text-decoration: none; /* removes the underline that links usually have */
}
a:hover, a:focus {
 text-decoration: underline; /* underline on mouseover */
 color: #B10022; /* colour change on mouseover */
}
```

2. Save the changes and preview your pages to see how your links look normally, and when the mouse is over them or when you press **tab** to select them.

Tip You can add quite a lot of formatting to links. For example, the css shown below would change your links so that when the mouse moves over them the background colour changes, the text becomes italic, a border appears above and below the text. Even the text size changes. Making a lot of changes like this would be a little extreme but you could try it to see how much difference it makes.

```
a:hover {
   color: #B10022;
   border-top: thin solid #B10022;
   border-bottom: thin solid #B10022;
   background-color: yellow;
   font-style: italic;
   font-size: larger;
}
```