



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

200
1822 - 2022

CSS-part 2

Man Geni Addysg Uwch yng Nghymru
Where Higher Education Began in Wales

www.uwtsd.ac.uk

Where do you define CSS Styles?

Reminder - Three Choices

- **Inline Style (avoid!)**

- applies to only a single element
- add a ***style attribute*** with the definitions to the element

- **Internal Styles**

- applies to a single web page document
- add a `<style></style>` element to the `<head>` section

- **External Style Sheet (best practice)**

- ideal for styling a multi-page website and for organising your CSS style definitions.
- style definitions are in a separate file and linked to the HTML document with a `<link>` tag in the head section

Adding CSS to HTML Content

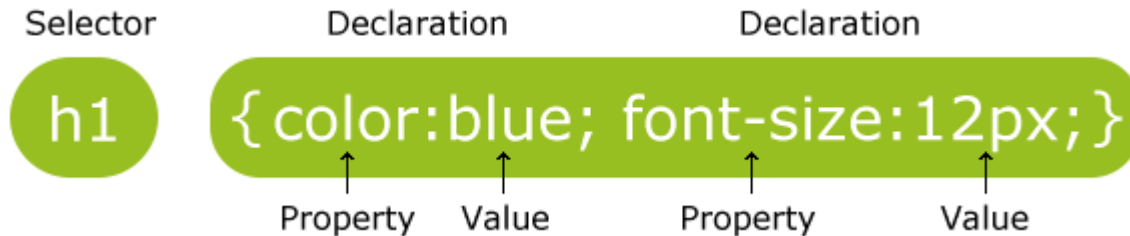
CSS can be applied to:

- A ***single*** element
 - e.g. a specific paragraph.
- A ***group*** of elements
 - e.g. to a set of paragraphs (all/ some of the paragraphs on the page),
 - or a mixed combination, e.g. paragraphs & headings.
- ***Part of*** an element
 - e.g. highlight some text in a paragraph.

Most Commonly Used CSS Selectors

- **Element selector:**
 - Used to style a *particular type of element* e.g. <h1>, <p>, <body>, <div>, etc.
 - Defined using the element type's tag name.
- **id selector:**
 - Used to style a *single unique element*
 - Defined using a **"#"** followed by the element's id attribute value.
- **Class selector**
 - Usually used to style *one or more similar elements*
 - Defined using a **"."** followed by the element class the style applies to.

Defining CSS Style Rules



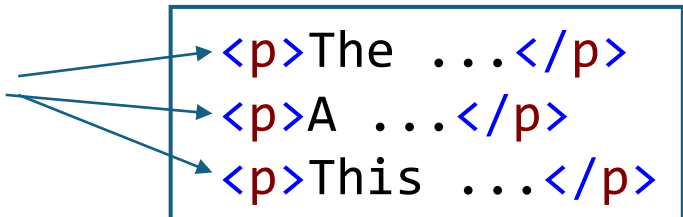
- **Selector:** chooses the HTML element to style.
- **Declarations:**
 - One or more rules enclosed within curly braces { ... }.
 - Each declaration is made up of:
 - A **Property Name**: the CSS property you want to style/format e.g. colour, background, font, width, etc.
 - A **Property Value**: the value you want to set the CSS property that you are styling to.
 - Semi-colons “;” used to separate multiple declarations.

Commonly Used CSS Selectors

Examples

- The **element** selector: applies to all `<p>` elements

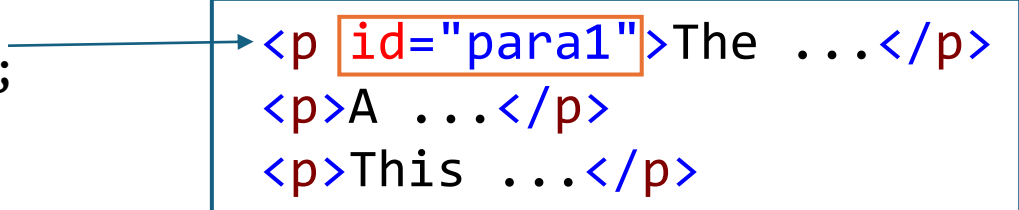
```
p {  
  text-align: center;  
  color: red;  
}
```



The diagram shows three HTML paragraph elements: `<p>The ...</p>`, `<p>A ...</p>`, and `<p>This ...</p>`. Three blue arrows originate from the `text-align: center;` and `color: red;` rules in the CSS block and point to each of the three paragraph elements, indicating that the styles are applied to all `<p>` elements.

- The **id** selector: applies to an element with id `para1`

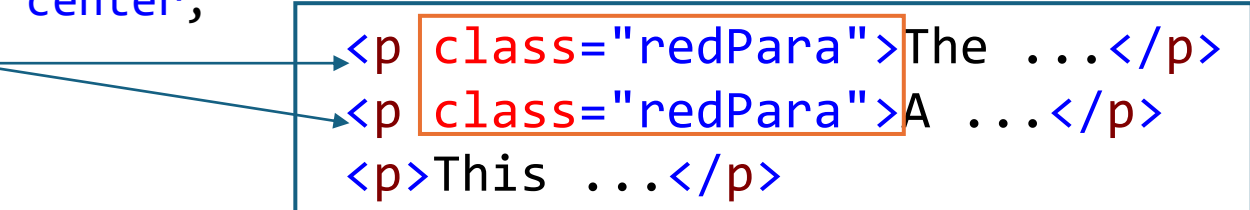
```
#para1 {  
  text-align: center;  
  color: red;  
}
```



The diagram shows three HTML paragraph elements. The first element, `<p id="para1">The ...</p>`, has its `id="para1"` attribute highlighted with an orange box. A blue arrow points from the `text-align: center;` rule in the CSS block to this specific element, indicating that the styles are applied only to the element with the id `para1`.

- The **class** selector: applies to any element with a class `redPara`

```
.redPara {  
  text-align: center;  
  color: red;  
}
```



The diagram shows three HTML paragraph elements. The first two elements, `<p class="redPara">The ...</p>` and `<p class="redPara">A ...</p>`, have their `class="redPara"` attributes highlighted with an orange box. Two blue arrows originate from the `text-align: center;` and `color: red;` rules in the CSS block and point to these two elements, indicating that the styles are applied to all elements with the class `redPara`.

The arrows show which paragraph elements the styles apply to.

Important HTML Attributes for CSS

- The **id** attribute:

- Uniquely identifies a **single element** on the page.
(No two elements on the same page should have the same id value.)
- Useful for CSS and also JavaScript programming.

```
<p id="para1"> . . . </p>
```

- The **class** attribute:

- A class name to group (class) together and identify **a set of multiple elements** to style

```
<p class="redPara"> . . . </p>
```

```
<p class="redPara"> . . . </p>
```

Grouping CSS Selectors

- If you have **different elements with the same style definitions**, you can **group them together**.
 - *It minimises code and eliminates duplication.*
- Separate each selector for the different elements with a comma before the style declarations:
 - This definition applies to all h1 and h2 elements as well as any element with a class attribute set to “redPara”:

```
h1, h2, .redPara {  
  text-align: center;  
  color: red;  
}
```

Dividing up the content – Div and Class

- Now we are going to Div(ide) up the page into sections using `<div>`
- Note that `<div>` itself doesn't actually change any formatting, unlike say `<p>`
- We can also make up some 'class' names and assign these to different sections
- Then we can set parameters such as colour and spacing to these classes in the CSS file

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> CSS demo</title>
  <link rel="stylesheet" href="example.css">
</head>
<body>
  <div class="container">
    <div class="top-para">
      <h1> Heading 1 text </h1>
      Lorem Ipsum dolor sit amet, consectetur adipiscing elit. Duis tellus.
      Donec ante dolor, iaculis nec, gravida ac, cursus in, eros. Mauris vestibulum,
      felis et egestas ullamcorper, purus nibh vehicula sem, eu egestas ante.
      Fusce tincidunt, lorem neque dapibus consectetur, leo orci mollis ipsum, eget suscipit eros purus in ante.
    </div>
    <div class="bot-para">
      <h2> Heading 2 text </h2>
      <p>At ipsum vitae est lacinia tincidunt. Maecenas elit orci, gravida ut, molestie non, venenatis vel,
      lorem. Sed lacinia. Suspendisse potenti. Sed ultricies cursus lectus.
      In id magna sit amet nibh suscipit euismod. Integer enim. Donec sapien ante, accumsan ut, sodales commodo, auctor quis,
    </div>
  </div>
</body>
</html>

```

```
Body    {
    background-color:#7f7f7f;
    color: white;
    font-family: Courier;
    font-size:16px;
    line-height:1em;
}

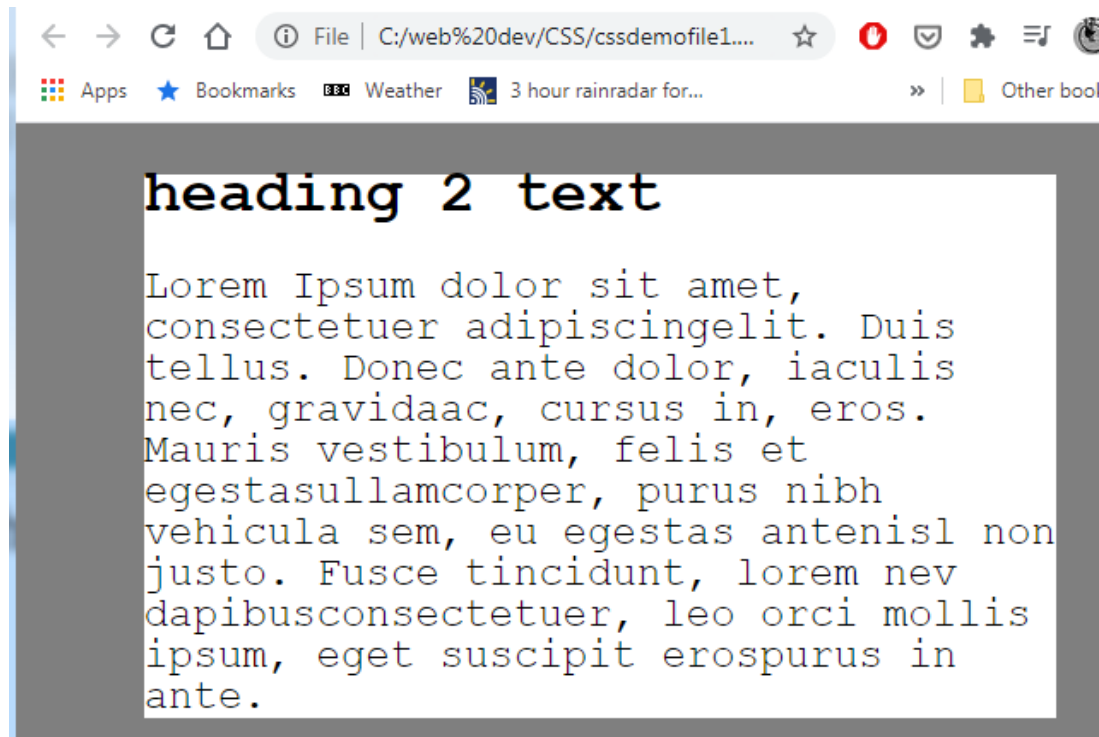
.top-para{
    background-color:white;
    color:black;
    
    text-align: left;
}

.container{
    width:200px;
    margin:auto;
}
```

Vary the size of the browser window and observe what happens.
Now change the width to say 80% and repeat.

Spacing

- We can see that by default the text is very close to the edge. We often need to change this.



The CSS Box Model – simplified

A rectangular box that represents HTML elements. Margins, borders and padding wrap around the actual content:

Margin: Clear, Transparent Space around the Border. No Background Colour!

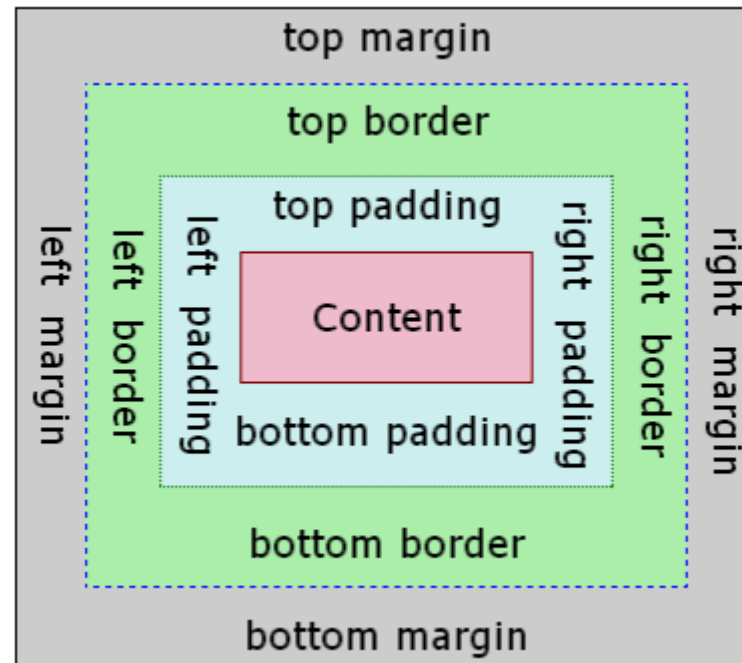
Border: an Edge that goes around the content and padding

Padding: Clear space around content. Same background colour as content

Content: the actual HTML content (text, images, etc.)

The CSS Box Model - full

Note - If needed, even more control can be achieved by varying the parameters of each side.



Exercise

In the paragraph class, experiment with setting the parameters of the box model. E.g. to set an overall padding:

```
.top-para{  
    background-color:white;  
    color:black;  
    padding:20px;  
}
```

Click here for the syntax of the various options:

https://www.w3schools.com/cssref/pr_border.asp

'Class' vs 'id'

What is the difference between the id and class attribute:

- The only difference between them is that "id" is unique in a page and can only apply to one element, while "class" selector can apply to multiple elements

HTML CODE

```
<header>
  <h1>Welcome to My Page </h1>
</header>

<div class="card">

  <h2>About Me</h2>
  <p>This is my custom design using plain
HTML & CSS.</p>

</div>
```

CSS CODE

```
body {
font-family: Arial;
background: #f4f4f4;
}

.card {
background: white;
padding: 20px;
margin: 10px;
border-radius: 8px;
}

header {
background: navy;
color: white;
padding: 10px;
text-align: center;
}
```

HTML CODE

Using 'id'

CSS CODE

```
<div class="card">
  <h2>About Me</h2>
  <p>This is my custom design using plain HTML & CSS.
</p>

</div>

<div id="specialNote">
  This section is styled using an <b>id selector</b>.
</div>
```

```
.card {
  background: white;
  padding: 20px;
  margin: 10px;
  border-radius: 8px;
}

#specialNote {
  background-color: gold;
  color: black;
  font-weight: bold;
  padding: 15px;
  text-align: center;
  border-radius: 8px;
}
```

What styles are we using today?

- Adding a splash of colour
- The text – font, size, etc.
- Adding a background or two
- Adding a border
- Spacing things out – paddings and margins
- Changing the size of stuff
- Styling links and navigation menus

Demonstration

Styling a HTML page with CSS.

CSS Colour

- Two ways to specify colour in CSS, using:
 - Red (**R**), Green (**G**) and Blue (**B**) colour component values
 - Hue, Saturation and Lightness of a colour
- **CSS3** also allows you to add transparency to a colour using the alpha (**a**) channel.

See more:

https://www.w3schools.com/css/css_colors.asp

CSS Colour

Red (**R**), Green (**G**) and Blue (**B**) Colour Values

- Hexadecimal Colours:
 - **#RRGGBB**
 - RR, GG, BB is a hex value between 0 and FF
 - Old Browsers understand this format!
- RGB and RGBA Colours:
 - **rgb**(red, green, blue)
 - **rgba**(red, green, blue, alpha)
 - red, green, blue is an integer value between 0 and 255
 - alpha defines opacity: 0.0 (transparent) 1.0 (opaque)
- *No need to memorise colours! Use sites such as:*

■ [Adobe Color CC](http://adobe.com/colorcc)

■ <http://paletton.com/> or similar

Reminder - The CSS Box Model

A rectangular box that represents HTML elements. Margins, borders and padding wrap around the actual content:

Margin: Clear, Transparent Space around the Border. No Background Colour!

Border: an Edge that goes around the content and padding

Padding: Clear space around content. Same background colour as content

Content: the actual HTML content (text, images, etc.)

CSS Border

- Draws a border line/strip around an element.
- Also used to specify no border wanted!
- The CSS border properties:
 - ***border-style***: must specify if you want a border.
 - ***border-width***: the thickness of the border
 - ***border-color***: the colour of the border
- Can also specify styles for *individual* border edges
 - Just insert what edge (-edge) after border
 - E.g. ***border-left-style***, ***border-bottom-width***

CSS Border

- CSS3 has some [new border properties](#).
 - The new rounded borders property is very useful.
- There is also a shorthand property that allows all styles to be specified in one line:

`border: border-width border-style (required) border-color;`

E.g. *border: 1px solid red;*

Margin

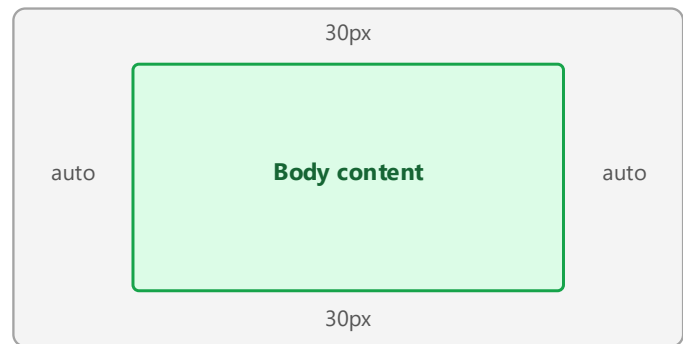
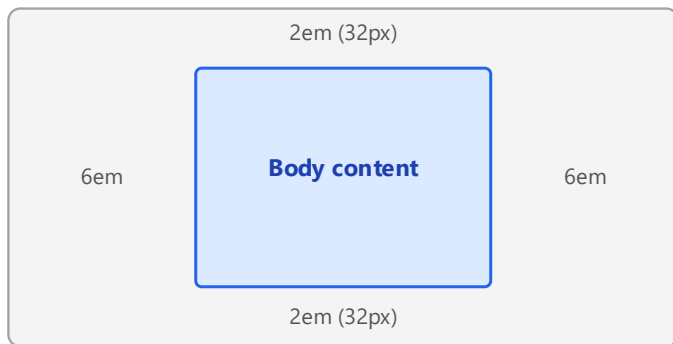
`margin: 2em 6em;`

- First value (2em) – Top and bottom margins.
- Second value (6em) – Left and right margins.
- Assuming the font size of the body is 16px:
 - 2em = 32px 2 × font size
 - 6em = 96px 6 × font size

em is a relative CSS unit. For margins, it is based on the computed font size of the element.

`margin: 30px auto;`

- 30px – Creates a 30-pixel margin at the top and bottom.
- auto – Automatically calculates the left and right margins to centre the element horizontally when space is available.



CSS Padding

- Adds space around an element (***but inside the border***).
- The value of the padding can be specified as a value in px, cm, pt, etc. or as a percentage %.
 - You can specify the ***same padding value for all sides*** using *padding: value;*
padding: 5px;
 - Or ***different for each side***, by inserting what edge before padding, e.g.
left-padding:10px;
right-padding: 0;

CSS Padding

Shorthand property:

- *padding: top right bottom left;*
padding: 10px 5px 5px 0px;
- *padding: top&bottom right&left;*
padding: 5px 10px;
- *padding: all sides equal;*
padding: 10px;

Styling Text I - CSS Text

CSS Properties used to format text:

- **color**: the text colour (note the American spelling!)
- **text-align**: the horizontal alignment of a block of text, i.e. center, left, right or justified
- **text-decoration**: allows you to sort of decorate your text, i.e. underline, line-through, overline, none
- **text-transform**: specify uppercase or lowercase text or capitalise first letter only
- **text-indent**: specifies indentation for the first line of a block of text

Styling Text II - CSS Font

CSS Properties to define the font:

- ***font-family***: what font do you want to use. Ideally specify several – the browser will use the first it finds
- ***font-style***: normal font, italic or oblique (leaning text which is actually very similar to italic text)
- ***font-size***: sets the size of the text
 - The default text size is 16px.
 - For best practices, specify font size in em units (see notes)
 - 1em is equal to the current default font size.
- ***font-weight***: how bold you want the text

CSS size/length units

CSS allows many ways of defining these –conceptually quite complicated – see:

<https://www.w3.org/Style/Examples/007/units.en.html>

- **Bottom line:**
- Stick to 'em' , 'px' or '%' unless you have a specific need or are going to print your page

	Recommended	Occasional use	Not recommended
Screen	em, px, %	ex	pt, cm, mm, in, pc
Print	em, cm, mm, in, pt, pc, %	px, ex	

CSS size/length units

The default font-size for text in browsers is 16px.

For best practices, create a style definition for the <html> and <body> tags and set the font-size to 100%

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

Then, for all other elements that you wish to specify a font-size for, use em units.

1em is equal to the current default font-size.

For example, to set a font size to 24px, this will be 1.5em, so

```
p {  
    font-size: 1.5em;  
}
```

To set a font-size to 8px, this will be 0.5em, so

```
p {  
    font-size: 0.5em;  
}
```

CSS Background

- Define background styling effects for an element:
 - **background-color**: the colour
 - **background-image**: what image
 - **background-repeat**: should a background image repeat horizontally (repeat-x) or vertically (repeat-y)
 - **background-attachment**: is the background image fixed or does it scroll with the rest of the page.
 - **background-position**: sets the starting position of a background image.
- CSS3 has some [new background properties](#).

Setting/resetting defaults

Let change the look of all the hyperlinks

```
a{  
    color:black;  
}  
a:hover{  
    color:yellow;  
}
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8" />
```

```
  <title>My first Webpage</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1>Welcome to My Page</h1>
```

```
  </header>
```

```
  <div class="card">
```

```
    <h2>About Me</h2>
```

```
    <p>This is my custom design using plain HTML and CSS.</p>
```

```
  </div>
```

```
</body>
```

```
</html>
```

```
body
{
font-family: Arial;
background: #f4f4f4;
}
header
{ background: navy;
color: white;
padding: 10px;
text-align: center;
}
.card
{ background: white;
padding: 20px;
margin: 10px;
border-radius: 8px;
}
.card:hover {
background-color: #e0e0e0; /* light gray background on hover */
transform: scale(1.03); /* slightly enlarges the card */
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* adds shadow */
color: Red;
}
```

Welcome to My Page

About Me

This is my custom design using plain HTML and CSS.

Welcome to My Page

About Me

This is my custom design using plain HTML and CSS.

CSS Media Queries

- You can use @media to interrogate the browser/device and format accordingly.
- **Media queries** are special CSS rules that let your webpage **change its style** depending on the **device or screen size**.

You can check things like:

Width and Height

Resolution

Orientation

see:

- https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Example

```
@media (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Condition	Example	Meaning
max-width	(max-width: 600px)	When the screen is 600px or smaller
min-width	(min-width: 768px)	When the screen is 768px or wider
orientation	(orientation: portrait)	When the device is in portrait mode
prefers-color-scheme	(prefers-color-scheme: dark)	When user prefers dark mode

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
  <link rel="icon" type="image/x-icon" href="courageous_17225402.png">
```

```
</head>
```

```
<body>
```

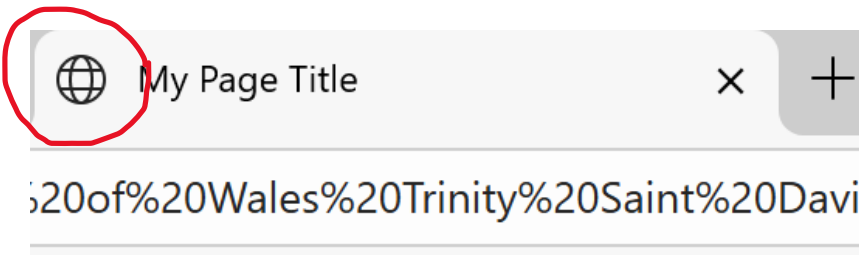
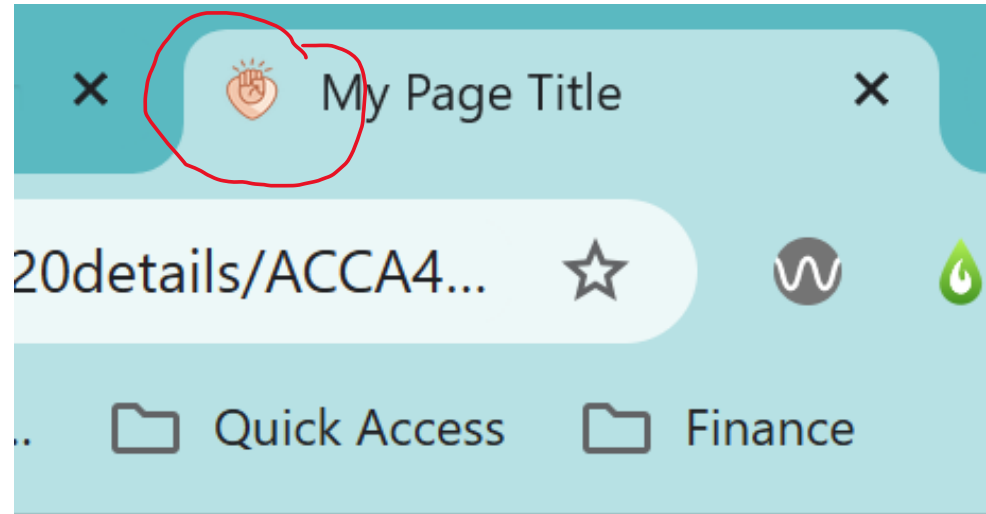
```
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

Favicon



Favicon

Common favicon formats

Format	Use case	Notes
.ico	Traditional, universally supported	Works in all browsers (even older ones). Recommended if you need maximum compatibility.
.png	Modern browsers	Preferred for higher quality and transparency. Common sizes: 32x32, 48x48, or 64x64.
.svg	Scalable, ideal for modern sites	Great for high-resolution displays. Supported by most modern browsers (except some older Edge/IE versions).
.jpg	Works, but not ideal	No transparency and can look blurry at small sizes.

Further Reading :

Part II: HTML Markup for Structure

Part III: CSS for Presentation

Robbins, J., 2018, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, O'Reilly Media