

Prac – Using CSS

For this prac you will use the basic.html page in the prac folder.

Aims:

- Learn to create an embedded CSS style in a HTML page
 - Learn how to create an external CSS file and use it with a web page
 - Learn to use styles with standard HTML selectors
 - Learn to create pseudo classes in CSS
 - Learn to create classes for the #id attribute in a selector
1. Open basic.html in your code editor. Look at the structure. Now double-click on the file to view it in a web browser. We suggest using Chrome or Firefox.
 2. Looking through the code in your text editor you should notice that basic.html references a style called navbar". We will write this CSS class in the prac today.
 3. Add the code below after the </title> tag and before the </head> tag:

```
<style type="text/css">

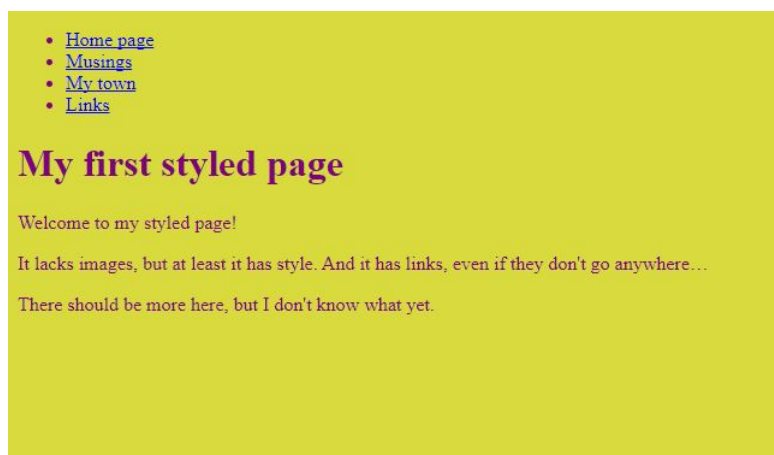
  body {

    color: purple;

    background-color: #d8da3d }

</style>
```

4. Save your file, refresh the web browser, and notice what has changed.



Note that the body text is now purple and the background color is an awful color of green.

Styles in CSS are made up of *rules*. Each rule has three parts:

- The **selector** (in the example: "body"), which tells the browser which part of the document is affected by the rule;
- The **property** (in the example, 'color' and 'background-color' are both properties), which specifies what aspect of the layout is being set;
- The **value** ('purple' and '#d8da3d'), which gives the value for the style property.

This example shows that rules can be combined. We have set two properties, so we could have made two separate rules:

```
body { color: purple }  
  
body { background-color: #d8da3d }
```

But since both rules affect the body, we only wrote "body" once and put the properties and values together.

Note:

Colors can be specified either using a predefined name for basic colors (green, blue, purple, etc) or specific RGB (red, green, blue) values.

See: https://www.w3schools.com/colors/colors_picker.asp for an online color picker to save time. Click on the HTML 5 button and the colors are laid out in red, green and blue values.

5. Use the color picker above to pick a different background for your page. We chose **#dcfaf9** for this example.
6. Add the code below (in red) to specify the font to be used by the page. Remember that we can add alternate fonts in case the user doesn't have the first font on their computer.

```
<style type="text/css">

body {

    font-family: Georgia, "Times New Roman", Times, serif;

    color: purple;

    background-color: #d8da3d}

h1 {

    font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }

</style>
```

7. Save and run your file and note the differences. We have made a style to change all body font to Georgia font (or Times New Roman). The h1 heading is now using Helvetica.
8. The list at the top of the HTML page is meant to become a navigation menu. We will put it on the left side, because that is a little more interesting than at the top...
9. The menu is already in the HTML page. It is the list at the top. The links in it don't work, since our "Web site" so far consists of only one page, but that doesn't matter now.

So we need to move the list to the left and move the rest of the text a little to the right, to make room for it. The CSS properties we use for that are 'padding-left' (to move the body text) and 'position', 'left' and 'top' (to move the menu).

10. Add the code below to our script tag to position our list of links for the navigation:

```
<style type="text/css">

body {

    padding-left: 11em;

    font-family: Georgia, "Times New Roman", Times, serif;

    color: purple;

    background-color: #d8da3d }

ul.navbar {

    position: absolute;

    top: 2em;

    left: 1em;

    width: 9em }


```

```
h1 {  
    font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }  
  
</style>
```

11. Try running your page, it should look something like the picture below:



You should notice that although the heading comes first on its own line (although indented 11EM), the menu has moved to the far left.

12. EM is a spacing amount. One EM equals the space taken up by a letter using the current font.

The 'position: absolute' says that the ul element is positioned independently of any text that comes before or after it in the document and the 'left' and 'top' indicate what that position is. In this case, 2em from the top and 1em from the left side of the window.

13. The navigation menu still looks like a list, instead of a menu. Let's add some style to it. We'll remove the list bullet and move the items to the left, to where the bullet was. We'll also give each item its own white background and a black square. (Why? No particular reason, just because we can.)

14. We also haven't said what the colors of the links should be, so let's add that as well: blue for links that the user hasn't seen yet and purple for links already visited:

15. Add the code below to your page and run it:

```
<style type="text/css">  
  
body {  
    padding-left: 11em;  
  
    font-family: Georgia, "Times New Roman", Times, serif;  
  
    color: purple;  
  
    background-color: #d8da3d }  
  
ul.navbar {  
  
    list-style-type: none;  
  
    padding: 0;
```

```

margin: 0;

position: absolute;

top: 2em;

left: 1em;

width: 9em }

h1 {

    font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }

ul.navbar li {

    background: white;

    margin: 0.5em 0;

    padding: 0.3em;

    border-right: 1em solid black }

ul.navbar a {

    text-decoration: none }

a:link {

    color: blue }

a:visited {

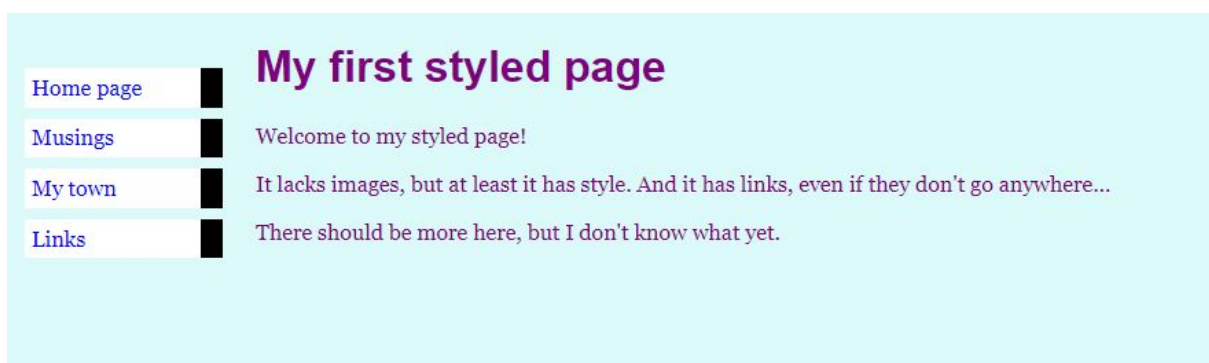
    color: purple }

</style>

```

Note that our style tag is getting pretty long. We should place it in an external file soon.

Your page should now look something like the image below:



We have done quite a lot in this last change:

- We've changed the list style to none, meaning that we don't get bullet points anymore
 - (note that this is done by overriding the tag and specifying how it will be displayed)
- We've overridden the tag to specify how each item in the list will display regarding margin, padding and border
- We've set text-decoration (the underline) for the anchor (link) tag to none
- We've specified that anchor tags (link tags) will be blue before they are clicked and purple after they have been clicked

We now have an HTML file with an embedded style sheet. But if our site grows we probably want many pages to share the same style. There is a better method than copying the style sheet into every page: if we put the style sheet in a separate file, all pages can point to it.

16. To make a separate stylesheet file, we need to create another empty text file. You can choose "New" from the File menu in the editor, to create an empty window. Then cut and paste everything that is inside the <style> element from the HTML file into the new window.

Don't copy the <style> and </style> tags themselves. They belong to HTML, not to CSS. In the new editor window, you should now have the complete style sheet:

```
body {  
  
    padding-left: 11em;  
  
    font-family: Georgia, "Times New Roman",  
        Times, serif;  
  
    color: purple;  
  
    background-color: #d8da3d }  
  
ul.navbar {  
  
    list-style-type: none;  
  
    padding: 0;  
  
    margin: 0;  
  
    position: absolute;  
  
    top: 2em;  
  
    left: 1em;  
  
    width: 9em }  
  
h1 {  
  
    font-family: Helvetica, Geneva, Arial,  
        SunSans-Regular, sans-serif }
```

```

ul.navbar li {
    background: white;
    margin: 0.5em 0;
    padding: 0.3em;
    border-right: 1em solid black }
ul.navbar a {
    text-decoration: none }
a:link {
    color: blue }
a:visited {
    color: purple }

```

17. Choose "Save As..." from the File menu, make sure that you are in the same directory/folder as the mypage.html file, and save the style sheet as "mystyle.css". Note: you need to make sure your text editor saves the file not as text but as Cascade Style Sheet (css).

18. Now go back to the window with the HTML code. Remove everything from the <style> tag up to and including the </style> tag and replace it with a <link> element, as follows:

```

<!DOCTYPE html>

<html>

<head>

    <title>My first styled page</title>

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

</head>

<body>

```

19. This will tell the browser that the style sheet is found in the file called "mystyle.css" and since no directory is mentioned, the browser will look in the same directory where it found the HTML file.

20. If you save the HTML file and reload it in the browser, you should see no change: the page is still styled the same way, but now the style comes from an external file. If you right click in the browser and choose View Source, you should see no CSS code, only the link to the

stylesheet.

21. Next we will create a "pseudo class" style rule, using the selector **a:hover**. Previously we created classes that would be used ANY TIME a selector was used (such as body text). However, many times we will want to apply a class to particular use of a selector.

We can define a pseudo class to be used anywhere by just labeling it dot + class name as below:

```
.awful { etc }
```

Or we can add it to a state of a selector:

```
a.hover.awful { etc }
```

22. Change an obvious property on the hover state like colour or background colour, and test your results.

You will sometimes see *dodgy* styles that use a different font size for the rollover effect. This is not a good idea because then your page "jumps around" as the text changes size. Perhaps try it and see how awful you can make it :)

An example of a HORRIBLE one might be:

```
a:hover.awful {  
    color: #FFF; background-color: #FF6; font-size: x-large; margin: 5em;  
    border: groove thick red; font-family: Comic Sans, Tahoma, sans-serif }
```

Note that you must use a dot (.) before your pseudo class name. This means that it only works on the hover state of an anchor tag.

See below that we have added the "awful" pseudo class to the navigation menu section of the CSS sheet underneath the styles for the anchor tag.

```
a:link {  
    color: blue }  
  
a:visited {  
    color: purple }  
  
a:hover.awful {  
    color: #FFF; background-color: #FF6; font-size: x-large;  
    margin: 5em;  
  
    border: groove thick red; font-family: Comic Sans, Tahoma,  
    sans-serif }
```


23. Once you have written your pseudo class, you can add it to the selector in the HTML page as seen below:

```
<!--Site navigation menu -->
<ul class= "navbar">

  <li><a href = "index.html" class= "awful"> Home page </a></li>

  <li><a href = "musings.html"> Musings </a></li>

  <li><a href = "town.html"> My town </a></li>

</ul>
```

24. On your html page, make a section named footer. We will use a <div> tag, to mark a section at the bottom of our page as the footer. Div tags are a block level element (meaning that they make a blank line before and after themselves. They are used to mark a section of a page for a certain purpose. If you just wanted to markup a smaller section of a paragraph or even one line you would use the tag.

Put the code below, right before the </body> tag on your html page.

```
<p>There should be more here, but I don't know</p>

<div id="footer">

  <p>&copy;Your name goes here, <a href= "http://www.jcu.edu.au"
target="_blank"> James Cook University</a></p>

</div>

</body>

</html>
```

25. We can style that div to give our footer a different look to the rest of the page and to place it correctly at the bottom of the window.
26. Although we could style the footer by creating a pseudo class like we did for the navbar, another option is override the ID selector in the div. Try the following code, by putting it in the bottom of your css file.

```

a:hover.awful {

    color: #FFF; background-color: #FF6; font-size: x-large; margin: 5em;

    border: groove thick red; font-family: Comic Sans, Tahoma, sans-serif }

#footer {

    left: 0;
    right: 0;
    bottom: 0;
    width: 100%;
    text-align: center;
    font-size: 0.7em;
    background-color: #2c2e38;
    color: #d7ceb2;
    font: "Helvetica", Arial, sans-serif;
    border-radius: 15px 15px 0px 0px;
    position: absolute;

}

#footer a { color: #FC3; }

```

27. Save your file, refresh it, and see how it looks. Play around with the properties to answer the following questions:

- Which property makes the footer always stick to the bottom of the page? What happens if you remove that property?
- How would you change the footer to have greater rounding of the corners? How would you make it rounded on all corners?