Step-by-Step Learning Activity

How CSS works with HTML

CSS is an acronym for Cascading Style Sheet. CSS is used to define how your web page will look. A CSS style declares a property and its value.

The three ways CSS styles can be added to a web page are:

1. **Inline style:** *Style can be attached to any HTML element/tag by the using the HTML attribute style=""*
2. **Internal style:** *Styles can be attached in the head of a HTML document using the style tag*
3. **External Style:** *Styles can be attached by linking to an external CSS document using the link element/tag.*

Styles cascade according to their selector (inline vs internal/external) and the last read style.

If there is more than one style declared for a tag the following takes precedence:

1. *Inline styles*
2. *External or internal stylesheets – whichever stylesheet is read last takes precedence*
3. *Browser default*

a) Open provided index.html document

Step 1: Download the Assets zipped folder and extract it in an appropriate place on your computer.

Step 2: Open the provided index.html file in Visual Studio Code

Basic CSS Syntax

Key takeaway: You must declare the property followed by a colon, then the value followed by a semicolon. For example, color:red;

Styling text – color property

Key takeaway: The color property styles the color of the text.

Styling text – font-family property

Key takeaway: The Font-family specifies the font for an element.

Styling text – font-size property

Key takeaway: The font-size property specifies how large the text appears.

Styling text – font-weight property

Key takeaway: The font-weight property specifies the weight of the font, from normal to bold.

Styling text – text-decoration property

Key takeaway: The text-decoration property specifies decorative additions to text such as the underline.

Styling lists – list-style-type property
Key takeaway: The list-style-type property specifies the kind of bullet or marker you want to use for unordered or ordered lists.

b) Add inline style
Key takeaway: Use the HTML style attribute to attach an inline style. Inline styles override internal and external styles.

Step 1: View the index.html in web browser.

Step 2: Add a style attribute to the h1 tag in the mydiv.

> <h1 style="">My Page Introduction</h1>

Step 3: Add the color property and a value to the style attribute.

> <h1 style="color: cornflowerblue;">My Page Introduction</h1>

Step 4: Save the index.html file.

Step 5: Preview in Live Server.

Step 6: Attach a font-family property and a value to the style attribute.

> <h1 style="color: cornflowerblue; font-family: Arial, 'sans-serif';">My Page Introduction</h1>

Step 7: Save the index.html file.

Step 8: Preview in Live Server.

Step 9: Add a style attribute to the p tag in the mydiv.

> <p style="">Vestibulum ut tempor…

Step 10: Add the color property and a value to the style attribute.

> <p style="color: grey;"> Vestibulum ut tempor…

Step 11: Add the font-family property and its value to the style attribute.

> <p style="color: grey; font-family:Times New Roman, 'serif';"> Vestibulum ut tempor…

Step 12: Add the font-size property and a value to the style attribute.

> <p style="color: grey; font-family:Times New Roman, 'serif'; font-size: 20px;"> Vestibulum ut tempor…

Step 13: Save the index.html file.

Step 14: Preview in Live Server.

Step 15: Add a style attribute to the link tag (a) in the paragraph.

> Sed egestas,

Step 16: Add the color property and a value to the style attribute.

> Sed egestas,

Step 17: Save the index.html file.

Step 18: Preview in Live Server.

Step 19: Switch back to HTML document to proceed to next segment.

CSS Syntax for internal and external stylesheets

Key takeaway: You must declare a selector and place the properties and values between curly braces for internal and external stylesheets. For example:

```
p {
    color:red;
    font-family:Times New Roman, 'serif';
}
```

Element Selectors

Key takeaway: Styling element selectors will define all instances of the HTML element unless overwritten by an alternate style.

```
p {
    color:red;
    font-family:Times New Roman, 'serif';
}
```

c)   Add an internal stylesheet

Key takeaway: Internal style sheets are styles that are declared in the head of a HTML document using the HTML style tag. For example:

```
<style>
    p {
        font-family: Arial, 'sans-serif';
    }
</style>
```

Step 1: Add an internal stylesheet to the head tag
```
<style>
</style>
```
Step 2: Add h2 element selector to target the h2 HTML element.
```
<style>
    h2 {
    }
</style>
```
Step 3: Add a font-family property and value inside the h2 selector
```
<style>
    h2 {
        font-family: "Arial Black", "sans-serif";
```

```
        }
    </style>
```

Step 4: Add a color property and value inside the h2 selector

```
<style>
    h2 {
        font-family: "Arial Black", "sans-serif";
        color:lightskyblue;
    }
</style>
```

Step 5: Save the index.html file.

Step 6: View in Liver Server.

Step 7: Add p element selector after the h2 selector style group to target the p HTML element.

```
<style>
    h2 {
        font-family: "Arial Black", "sans-serif";
        color:lightskyblue;
    }
    p {
    }
</style>
```

Step 8: Add a font-family property and value inside the p selector

```
    p {
        font-family: Arial, 'sans-serif';
    }
```

Step 9: Save the index.html file.

Step 10: View in Liver Server.

Step 11: Switch back to HTML document to proceed to next segment.

Styling links – a selector, pseudo selectors

Key takeaway: You can specify a different style for any state that the link is in by declaring the selector (a) and the state (hover) separated by a colon. For example

```
    a:hover {
      color: lightred;
    }
```

d)   Styling links – a:link pseudo selector

Step 1: Add a:link to the internal style sheet under the p tag

```
a:link {
}
```

Step 2: Set the color of the link normal state text to crimson

```
a:link {
    color:crimson;
}
```

Step 3: Get rid of the default underline decoration for links

```
a:link {
    color:crimson;
    text-decoration: none;
}
```

Step 4: Save the index.html file.

Step 5: Preview in Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

e)   Styling links – a:hover pseudo selector

Step 1: Add a:hover to the internal style sheet under the a:link

```
a:hover {
}
```

Step 2: Set the color of the link hover state text to light coral

```
a:hover {
    color: lightcoral;
}
```

Step 3: Save the index.html file.

Step 4: Preview in Live Server.

Step 5: Switch back to HTML document to proceed to next segment.

f)   Styling links – a:active pseudo selector

Step 1: Add a:active to the internal style sheet under the a:hover

```
a:active {
}
```

Step 2: Set the color of the link click or active state text to dark red

```
a:active {
    color:darkred;
```

}

Step 3: Save the index.html file.

Step 4: Preview in Live Server.

Step 5: Switch back to HTML document to proceed to next segment.

g) Styling links – a: visited pseudo selector

Step 1: Add a:visited pseudo selector to the internal style sheet under the a:active

a:visited {

}

Step 2: Set the color of the link normal state text to crimson

a:visited {

color:maroon;

}

Step 3: Save the index.html file.

Step 4: Preview in Live Server.

Step 5: Switch back to HTML document to proceed to next segment.

h) Styling lists – ul and ol selectors

Key takeaway: the ul and ol selectors are used to style list elements.

Step 1: Add ul selector to the internal style sheet under the a:visited

ul {

}

Step 2: Set the list-style-type property to square

ul {

list-style-type: square;

}

Step 3: Save the index.html file.

Step 4: Preview in Live Server.

Step 5: Why the list is not Arial?

Step 6: Switch back to HTML document to proceed to next segment.

External CSS stylesheets

Key takeaway: Use the external stylesheets to create consistent styles across your website.

i) Create an External CSS stylesheet

Key takeaway: Specify the @charset and save the text document as a .css file to create an external style sheet.

Step 1: Open the file called myStyles.css found under the older named "**css**"

Step 2: Set the charset to " UTF-8";

@charset "UTF-8";

Step 3: Declare the document to be CSS

@charset "UTF-8";

/* CSS Document */

Step 4: Save the myStyles.css file.

j) Add an External CSS stylesheet to an HTML document

Key takeaway: Use the link tag in the head of the HTML document to attach a CSS stylesheet to the web page.

Step 1: Return to the index.html file. In the head tag, add the link tag

<link>

Step 2: Specify the relationship to the web page

<link rel="stylesheet">

Step 3: Specify the type of file that you are linking to

<link rel="stylesheet" type="text/css">

Step 4: Specify the URL to the file you are linking to

<link rel="stylesheet" type="text/css" href="css/myStyles.css">

Step 5: Save the index.html file.

k) CSS comments

Key takeaway: Use CSS comments to organize and explain what your style sheet is doing. The syntax is / My Comment */*

Step 1: Add a comment to group general styles

/* General Styles */

Step 2: Add a comment to group table styles

/* Style the table */

Step 3: Add a comment to group form styles

/* Style the form */

Step 4: Add a comment to group footer styles

/* Style the footer */

Step 5: Save the myStyles.css file.

Basic CSS graphic properties – width and height
Key takeaway: Use the width and height to specify the width of an object or container.
Basic CSS graphic properties – background-color
Key takeaway: The background-color property…
Basic CSS graphic properties – border
Key takeaway: Use the border property to specify the width, style and color of a container or table.
Basic CSS graphic properties – border-radius
Key takeaway: Use the border-radius to specify the curve of the corners of a container or button.
Basic CSS graphic properties – outline
Key takeaway: Use the outline property to highlight or make an element standout.
Basic CSS graphic properties – border-collapse
Key takeaway: Use the border-collapse property to collapse table borders into a single border.
Basic CSS spacing properties – margin
Key takeaway: The margin property specifies the space outside an element.
Basic CSS spacing properties – padding
Key takeaway: The padding property specifies the space inside an element
Basic CSS spacing properties – text-align
Key takeaway: The text-align property specifies how text is aligned in a container.
l)   Styling tables – the table selector
Key takeaway: Use the table selector to style general appearance of the entire table.

Step 1: In the CSS document add the table selector below the style the tables comment
    table {
    }
Step 2: Set the font-family for the entire table
    table {

```
        font-family: Arial, 'sans-serif';
    }
```

Step 3: Collapse the table border to be a single border

```
    table {
        font-family: Arial, 'sans-serif';
        border-collapse: collapse;
    }
```

Step 4: Set the width to be 50% of the viewable area

```
    table {
        font-family: Arial, 'sans-serif';
        border-collapse: collapse;
        width: 50%;
    }
```

Step 5: Save the myStyles.css file.

Step 6: Preview your index.html file in the Live Server.

Step 7: Switch back to HTML document to proceed to next segment.

m)   Styling tables – th selector

Key takeaway: *Use the th selector to style the table heading.*

Step 1: In the CSS document add the th selector below the table selector

```
    th {
    }
```

Step 2: Set the text colour to be white

```
    th {
        color: white;
    }
```

Step 3: Set the background color to be blue

```
    th {
        color: white;
        background-color: cornflowerblue;
    }
```

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

n)   Styling tables – td selector

Key takeaway: *Use the td selector to style the table columns.*

Step 1: In the CSS document add the td selector below the th selector

```
td {
}
```

Step 2: Set the border to be solid, 1 px wide, and lightgrey

```
td {
    border: 1px solid lightgrey;
}
```

Step 3: Center the text in the column

```
td {
    border: 1px solid lightgrey;
    text-align: center;
}
```

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

Styling tables – tr selector

Key takeaway: *Use the tr selector to style rows of a table.*

o) Styling tables – :nth-child()

Key takeaway: *Create alternating styles for rows and columns using :nth-child(odd or even).*

Step 1: In the CSS document add the tr selector after the td selector

```
tr{
}
```

Step 2: Add the pseudo selector to the td selector

```
tr:nth-child() {
}
```

Step 3: Specify that only odd rows should be styled

```
tr:nth-child(odd) {
}
```

Step 4: Set the odd rows background colour to be light grey

```
tr:nth-child(odd) {
    background-color: whitesmoke;
}
```

Step 5: Save the myStyles.css file.

10

Step 6: Preview your index.html file in the Live Server.

Step 7: Switch back to HTML document to proceed to next segment.

p) Styling tables – :hover pseudo selector

Key takeaway: Use the :hover state selector with td or tr tags to create alternate styles for when the mouse hovers over the row or column.

Step 1: In the CSS document add another tr selector to below the other tr

```
tr {
}
```

Step 2: Add the :hover pseudo selector to the tr selector

```
tr:hover {
}
```

Step 3: Set the background colour property to be light grey

```
tr:hover {
    background-color: lightgrey;
}
```

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

q) Styling tables – grouping selectors together

Key takeaway: Use a comma to group selectors together that have the same properties and values.

Step 1: In the CSS document add another th selector to below the tr:hover

```
th {
}
```

Step 2: Add a comma and then the td selector

```
th, td {
}
```

Step 3: Set the margin (spacing outside the table) to be 0px

```
th, td {
    margin: 0px;
}
```

Step 4: Set the padding (spacing inside the table) to be 10px

```
th, td {
    margin: 0px;
    padding: 10px;
```

}

Step 5: Save the myStyles.css file.

Step 6: Preview your index.html file in the Live Server.

Step 7: Switch back to HTML document to proceed to next segment.

r)   Styling forms

Key takeaway: *Use the form selector to style the container of the form elements.*

Step 1: In the CSS document add the form selector below the style the form comments

<mark>form {</mark>

<mark>}</mark>

Step 2: Set the background to be whitesmoke

form {

<mark>background-color: whitesmoke;</mark>

}

Step 3: Set the padding (inside spacing) to 20px for the top and bottom and 50 px for right and left (order rotates clockwise from the top)

form {

background-color: whitesmoke;

<mark>padding: 20px 50px;</mark>

}

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

s)   Styling forms – styling elements that are inside another element

Key takeaway: *To create an alternate style for an element that is within another element use the a space between the elements.*

Step 1: In the CSS document add another form selector following the previous one

<mark>form {</mark>

<mark>}</mark>

Step 2: Add a space followed by a p selector

form <mark>p</mark> {

}

Step 3: Make the font-size bigger at 16px

form p {

<div style="border: 1px solid black">

```
            font-size: 16px;
      }
```

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

</div>

t) Styling forms – input type selectors

Key takeaway: To style a specific input type use the syntax input[type=value]

Step 1: In the CSS document add the input[type=text] selector below the form selector

```
input[type=text] {
}
```

Step 2: Set the font-size to 12

```
input[type=text] {
      font-size: 12px;
}
```

Step 3: Set the padding (inside spacing) for all directions to 5

```
input[type=text] {
      font-size: 12px;
      padding: 5px;
}
```

Step 4: Set the top and bottom margin (outside spacing) to 12px and the right and left padding to 0px

```
input[type=text] {
      font-size: 12px;
      padding: 5px;
      margin: 12px 0px;
}
```

Step 5: Set the border to 1px wide, solid and light grey

```
input[type=text] {
      font-size: 12px;
      padding: 5px;
      margin: 12px 0px;
      border: 1px solid lightgrey;
}
```

Step 6: Set the border radius to have a 6px curve

```
input[type=text] {
      font-size: 12px;
      padding: 5px;
```

```
        margin: 12px 0px;

        border: 1px solid lightgrey;

        border-radius: 6px;

    }
```

Step 7: Save the myStyles.css file.

Step 8: Preview your index.html file in the Live Server.

Step 9: Switch back to HTML document to proceed to next segment.

u) Grouping selectors with the same values together

Key takeaway: Use a comma to group selectors together that have the same properties and values.

Step 1: In the CSS document group the input[type=text] and textarea selectors together separated by a comma, as they have the same properties and values

```
    input[type=text], textarea {

        font-size: 12px;

        padding: 5px;

        margin: 12px 0px;

        border: 1px solid lightgrey;

        border-radius: 6px;

    }
```

Step 2: Save the myStyles.css file.

Step 3: Preview your index.html file in the Live Server.

Step 4: In the CSS document group the input[type=text], textarea and select selectors together separated by a comma, as they have the same properties and values

```
    input[type=text], textarea, select {

        font-size: 12px;

        padding: 5px;

        margin: 12px 0px;

        border: 1px solid lightgrey;

        border-radius: 6px;

    }
```

Step 5: Save the myStyles.css file.

Step 6: Preview your index.html file in the Live Server.

Step 7: Switch back to HTML document to proceed to next segment.

v) Styling forms – select selector

Key takeaway: Use the select selector to style a select dropdown menu.

Step 1: In the CSS document add the select selector below the previous grouping

select {

}

Step 2: Set the height to 30px

select {

height: 30px;

}

Step 3: Set the width to 300px

select {

height: 30px;

width: 300px;

}

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

w) Styling forms – textarea selector

Key takeaway: Use the textarea selector to style input for multiple lines of text.

Step 1: In the CSS document add the textarea selector below the select selector

textarea {

}

Step 2: Set the height to 150px

textarea {

height: 150px;

}

Step 3: Set the width to 100%

textarea {

height: 150px;

width: 100%;

}

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server.

Step 6: Switch back to HTML document to proceed to next segment.

x) Styling forms – :focus pseudo selector

Key takeaway: Use the :focus pseudo selector to create an alternate style for when the user is interacting with a specific input in your form. This helps a user keep track of where they are in the form.

Step 1: In the CSS document add the :focus pseudo selector below the textarea selector

input[type=text]:focus, textarea:focus, select:focus {

}

Step 2: Set the border to be 2px wide, solid and blue

input[type=text]:focus, textarea:focus, select:focus {

border: 2px solid cornflowerblue;

}

Step 3: In the CSS document set the outline to none

input[type=text]:focus, textarea:focus, select:focus {

border: 2px solid cornflowerblue;

outline: none;

}

Step 4: Save the myStyles.css file.

Step 5: Preview your index.html file in the Live Server. Select the input areas to see what happens when they are selected.

Step 6: In the CSS document group the input[type=text]:focus and the textarea:focus together, separated by a comma, as they have the same properties and values

input[type=text]:focus, textarea:focus {

border: 2px solid cornflowerblue;

outline: none;

}

Step 7: Add the select:focus to the group, separated by a comma, as they have the same properties and values

input[type=text]:focus, textarea:focus, select:focus {

border: 2px solid cornflowerblue;

outline: none;

}

Step 8: Save the myStyles.css file.

Step 9: Preview your index.html file in the Live Server. Select the input areas to see what happens when they are selected.

Step 10: Switch back to HTML document to proceed to next segment.

y) Styling forms – input[type=submit] selector

Key takeaway: Use the input[type=submit] selector to style a submit button.

Step 1: In the CSS document add the input[type=submit] below the previous selector group

```
input[type=submit] {

}
```

Step 2: Set the text color to white

```
input[type=submit] {
    color: white;
}
```

Step 3: Set the font-size to 12px

```
input[type=submit] {
    color: white;
    font-size: 12px;
}
```

Step 4: Set the padding to 10px

```
input[type=submit] {
    color: white;
    font-size: 12px;
    padding: 10px;
}
```

Step 5: Set the margin to 12px

```
input[type=submit] {
    color: white;
    font-size: 12px;
    padding: 10px;
    margin: 12px;
}
```

Step 6: Set the border-radius to 10px

```
input[type=submit] {
    color: white;
    font-size: 12px;
    padding: 10px;
    margin: 12px;
    border-radius: 10px;
}
```

Step 7: Set the set the background-color to blue

```
input[type=submit] {
    color: white;
    font-size: 12px;
    padding: 10px;
    margin: 12px;
```

```
        border-radius: 10px;
        background-color: cornflowerblue;
    }
```

Step 8: Set the border to none

```
input[type=submit] {
        color: white;
        font-size: 12px;
        padding: 10px;
        margin: 12px;
        border-radius: 10px;
        background-color: cornflowerblue;
        border: none;
    }
```

Step 9: Save the myStyles.css file.

Step 10: Preview your index.html file in the Live Server.

Step 11: Switch back to HTML document to proceed to next segment.

z)   Styling forms – button selector

Key takeaway: *Use the button selector to style button elements.*

Step 1: In the CSS document group the button selector with the input[type=submit] selector separated by a comma as they have the same values

```
input[type=submit], button {
        color: white;
        font-size: 12px;
        padding: 10px;
        margin: 12px;
        border-radius: 10px;
        background-color: cornflowerblue;
        border: none;
    }
```

Step 2: Save the myStyles.css file.

Step 3: Preview your index.html file in the Live Server.

Step 4: In the CSS document add the button selector below the previous group

```
button {
}
```

Step 5: Set the set the background-color to crimson

```
button {
```

```
        background-color: crimson;
    }
```

Step 6: Save the myStyles.css file.

Step 7: Preview your index.html file in the Live Server.

Step 8: Switch back to HTML document to proceed to next segment.

aa) Styling the footer

Key takeaway: Use the footer selector to style the general appearance of the footer element and elements that are inside the footer.

Step 1: In the CSS document add the footer selector below the style the footer comment

```
    footer {

    }
```

Step 2: Set the set margin-top to be 40px

```
    footer {
        margin-top: 40px;
    }
```

Step 3: Set the set the padding to 10px

```
    footer {
        margin-top: 40px;
        padding: 10px;
    }
```

Step 4: Set the set the background-color to blue

```
    footer {
        margin-top: 40px;
        padding: 10px;
        background-color:cornflowerblue;
    }
```

Step 5: Set the set the text color to white

```
    footer {
        margin-top: 40px;
        padding: 10px;
        background-color:cornflowerblue;
        color: white;
    }
```

Step 6: Save the myStyles.css file.

Step 7: Preview your index.html file in the Live Server.

Step 8: Switch back to HTML document to proceed to next segment.

ab) Styling the footer – styling alternate link styles

Key takeaway: Use a space between selectors to specify styles that are only rendered when the second element is inside the first.

Step 1: In the CSS document add the footer selector previous footer selector

```
footer {
}
```

Step 2: Add the a:link selector after the footer, separated by a space

```
footer a:link {
    color: lightcyan;
}
```

Step 3: Set the text color to light blue

```
footer a:link {
    color: lightcyan;
}
```

Step 4: In the CSS document add the footer selector previous footer selector

```
footer {
}
```

Step 5: Add the a:active and a:visited selector after the footer, separated by a space

```
footer a:active, a:visited {
}
```

Step 6: Set the set the text color to cyan

```
footer a:active, a:visited {
    color: cyan;
}
```

Step 7: In the CSS document add the footer selector previous footer selector

```
footer {
}
```

Step 8: Add the a:hover after the footer

```
footer a:hover {
}
```

Step 9: Set the set the text color to cyan

```
footer a:hover {
    color: paleturquoise;
}
```

Step 10: Save the myStyles.css file.

Step 11: Preview your index.html file in the Live Server.

Step 12: Why the style did not change?

Step 13: Switch back to HTML document to proceed to next segment.

ac) Cascading Styles

Key takeaway: You can attach styles to a HTML document three ways: inline styling, internal styling and attaching styles via an external stylesheet.

Inline styles take precedence over internal and external styles. Between internal or external styles, precedence is given to whichever is read last by the browser.

Step 1: In the HTML document comment out the link styles in the internal style sheet

```
/*a:link {
    color:crimson;
    text-decoration: none;
}
a:hover {
    color: lightcoral;
}
a:active {
    color:darkred;
}
a:visited {
    color:maroon;
}*/
```

Step 2: Save the index.html file

Step 3: Copy the internal link styles

Step 4: Paste the styles into the CSS document below the p selector in the general styles section

```
a:link {
    color:crimson;
    text-decoration: none;
}
a:hover {
    color: lightcoral;
}
a:active {
    color:darkred;
}
a:visited {
    color:maroon;
```

}

Step 5: Save the myStyles.css file.

Step 6: Preview your index.html file in the Live Server.

Step 7: Why did the style change?

Step 8: Switch back to HTML document to proceed to next segment.

CSS code validator

Key takeaway: Use the free WC3 validator https://jigsaw.w3.org/css-validator/ to find bugs and ensure there are no errors in your code.

Step 1: In the CSS document select all the CSS code

Step 2: Go to https://jigsaw.w3.org/css-validator/ and past the code into the validate by direct input tab

Step 3: If there are no errors the message *"Congratulations! No Error Found"* will appear. If there are errors the file will list the errors with details as to what is wrong.