Hands-on Lab: CSS Basics - Styling Your Web Page

Estimated time: 30 minutes

In this lab, you will take the role of a web developer who has been asked by a primary school to develop a web page about the solar system.

Objectives

After completing this lab, you will be able to:

1. Understand different ways to apply CSS to HTML
2. Create a basic web page
3. Specify the font family and font size for text
4. Use colors to change the appearance of elements
5. Create borders
6. Practice Exercises: Apply CSS and add CSS Styling to an HTML page through

1. Understanding different ways to apply CSS to HTML

In this exercise, you will learn about the three primary methods of applying CSS to HTML: **inline CSS**, **internal CSS**, and **external CSS**.
You will also practice implementing each method with examples to understand their differences and use cases.

Types of CSS and Their Descriptions

1. **Inline CSS**
   - Applied directly within an HTML element's `style` attribute
   - Ideal for quick styling of a single element
   - **Example use case**: Highlighting a specific word or line with unique styling
   - **Pros**: Simple to use for small changes
   - **Cons**: Difficult to maintain for larger projects; violates separation of content and style
2. **Internal CSS**
   - Defined within the `<style>` tag in the `<head>` section of the HTML document
   - Best for styling specific to a single HTML page
   - **Example use case**: Designing a webpage where styles aren't shared across multiple pages
   - **Pros**: Allows centralized styling for one document
   - **Cons**: Not reusable across multiple files
3. **External CSS**
   - Written in a separate `.css` file and linked to the HTML using a `<link>` tag
   - Preferred for websites with multiple pages requiring consistent styling
   - **Example use case**: Large websites where styles are shared across pages
   - **Pros**: Promotes reusability and cleaner code; separation of concerns
   - **Cons**: Requires an extra HTTP request to load the CSS file

2. Create a web page

In this exercise, you will create a simple HTML web page about the solar system.

On the window to the right, click **File > New File**.

A **New File** window opens

Enter `solarsystem.html` as the file name and click **OK**.

Now we'll add some HTML to the file and then apply CSS styles.

Copy and paste the following code into the file you created and save the page:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Solar System</title>
    </head>
    <body>
        <h1>Solar System</h1>
        <img src="https://upload.wikimedia.org/wikipedia/commons/c/cb/Planets2013.svg" alt="Solar System image" height= "250px" width="700px">
        <p>The solar system consists of the sun and everything that orbits the sun. This includes the eight planets and their moons, the dwarf planet
```

```
            </p>
            <p> All the planets and dwarf planets, the rocky asteroids, and the icy bodies in the Kuiper belt move around the Sun in elliptical orbits in
            </p>
            <p><strong><u> Planets in the solar system: </u></strong> </p>
            <table>
                <tr>
                    <th>No. </th>
                    <th>Planet Name </th>
                    <th>Distance from Sun (in million km)</th>
                </tr>
                <tr>
                    <td>1 </td> <td>Mercury</td> <td>57.91 m km</td>
                </tr>
                <tr>
                    <td>2 </td> <td>Venus</td> <td>108.2 m km</td>
                </tr>
                <tr>
                    <td>3 </td> <td>Earth</td> <td>149.6 m km</td>
                </tr>
                <tr>
                    <td>4 </td> <td>Mars</td> <td>227.9 m km</td>
                </tr>
                <tr>
                    <td>5 </td> <td>Jupiter</td> <td>778.5 m km</td>
                </tr>
                <tr>
                    <td>6 </td> <td>Saturn</td> <td>1.434 b km</td>
                </tr>
                <tr>
                    <td>7 </td> <td>Uranus</td> <td>2.871 b km</td>
                </tr>
                <tr>
                    <td>8</td> <td>Neptune</td> <td>4.495 billion km</td>
                </tr>
            </table>
        </body>
    </html>
```

Preview your HTML File

To preview your file, you can use the built-in Live Server extension by following the instructions below.

1. Open the file explorer and navigate to your file

2. Right click your file & click "Open with Live Server™"

3. This should show a notification mentioning that the server has started on port 5500

4. Click the Skills Network button on the left, it will open the "Skills Network Toolbox". Then click the Other and then Launch Application. From there you enter the port no. as 5500 and launch.

5. Click the file name to view its preview

6. Your page should look like this:

3. Specify a font family and font size for text

CSS helps define style, layout, colors, and fonts for your pages. In this exercise, you will set the font family. We'll also use CSS to change the font characteristics for some elements.

We'll use the **font-family** and **font-size** properties to change the appearance of the heading.

Note: In this lab, you will apply styles using **inline CSS format**. You will add the **style** attribute to the element to change the style.

Change the heading font

Modify the `<h1>` tag so that it looks like this:

```
<h1 style="font-family: Cursive">Solar System</h1>
```

Here, `Cursive` is a generic font family. You could also specify a font family such as `Arial` or a group of font family names separated by a comma, such as `font-family: Arial, serif`.

When you have updated the tag, save the file and reload your web page. It should now look like this, assuming you have specified the `Arial, serif` font family:

Change the font size

Add the **font-size** property to the `<h1>` style so that your code looks like this:

```
<h1 style="font-family: Cursive; font-size: 70px"> Solar System </h1>
```

When you have made the change, save your file. Your page should now look like this:

4. Change the color of an element

You will now set the background color of the page and update the color of an HTML element. To change the background color, you use the CSS **background-color** property. To change an element's color, you use the **color** property.

Change the background color

To change the background color of the body of the web page, update the `<body>` tag as in the following code:

```
<body style="background-color:wheat">
```

Change the font color

You specify a color for an element using a predefined color name, or by using RGB, HEX, HSL, RGBA, or HSLA values.

The RGB system enables you to specify a color as a combination of the primary colors red, green, and blue. RGB colors range from *rgb(0,0,0)*, which is black, to *rgb(255,255,255)* which is white.

Change the color of your heading using the RGB system, as follows:

```
<h1 style="font-family: Cursive; font-size:70px; color:rgb(139,0,0)">Solar System</h1>
```

When you have updated the code, save the file. The web page should now look like this:

5. Create borders

Tables created in HTML do not display a border by default. Using CSS, we can apply borders to tables, `<div>` elements, and other tags.

Add a table border

We'll add a border to the table in your web page. The border will be solid, black, and two pixels wide.

Update the `<table>` tag with the following CSS code:

```
<table style="border: 2px solid black">
```

When you have made the changes, save the file. The table in the page should now look like this:

Add a border to table headings and table cells

CSS gives you the flexibility to add a style to multiple instances of the same HTML tag. For example, instead of specifying the style for each individual `<th>` and `<td>` tag, we can add a style to all of those tags using **internal CSS**. We'll add a `<style>` tag within the `<head>` section of our page to specify how we want all `<th>` and `<td>` tags to display.

To apply an internal CSS style to your page, replace the `<head>` section with the code below:

```
<head>
    <title>Solar System</title>
    <style>
        table,th,td {
            border: 2px solid black;
        }
    </style>
</head>
```

When you have made the changes, save the file. The page should now look like this:

Highlight the table header

Finally, let's make the table header stand out by specifying a background color for the entire table row. By adding the style to the row, you do not need to update each individual `<th>` tag within the row.

Add a style to the first `<tr>` tag in the table (the header row), as follows:

```
 <tr style="background-color:yellow">
    <th>No. </th>
    <th>Planet Name </th>
    <th>Distance from Sun </th>
</tr>
```

When you have made the changes, save the file. The page should now look like this:

Practice Exercise 1: Applying CSS

Part 1: Inline CSS

1. Open your HTML editor and create a new file named `inline-css.html`

2. Use the `style` attribute within an HTML element to apply inline CSS
3. Apply styles to an `<h1>` element to make the text **blue** and center-aligned
4. Add a `<p>` element with a **font size of 18px** and **green text color**
5. Save the file and view it in your browser

▶ Click here to view the solution code

Part 2: Internal CSS

1. Create a new file named `internal-css.html`
2. Add a `<style>` tag within the `<head>` section of the HTML document
3. Inside the `<style>` tag:
   - Style an `<h2>` element with a **font-family of Verdana, sans-serif** and a **solid border**
   - Style the `<ol>` and `<li>` elements so the list items have a **font size of 18px** and **blue text color**
4. In the `<body>` section:
   - Add an `<h2>` element with heading text
   - Add an ordered list `<ol>` with three items `<li>`
5. Save the file and view it in your browser

▶ Click here to view the solution code

Part 3: External CSS

1. Create a new file named `external-css.html`
2. Create a separate CSS file named `styles.css`
3. In the `styles.css` file:
   - Style the `<h3>` element with **dark green text** and **left alignment**
   - Style the `<table>` element with a **black border** and a **light blue background color**
4. Link the `styles.css` file to `external-css.html` using the `<link>` tag
5. In the `<body>` section of `external-css.html`:
   - Add an `<h3>` element with heading text
   - Add a `<table>` element with two rows and two columns
6. Save both files and view the `external-css.html` file in your browser

▶ Click here to view the solution code

Practice Exercise 2: Add CSS styling to an HTML page through links

You should have completed this lab on using HTML5 strcutural elements for untis conversion. If not, you can do it now.

In this lab, you will add CSS styling to improve its UI.

1. Click on the button below to open `index.html`. If you don't have the file in the lab environment, you will get a prompt asking if you want to create a new one. You can choose to do that and paste the following code in it and save the file.

Open **index.html** in IDE

▶ Click to see the content of index.html

2. Click the button below to create a new file named style.css

Open **style.css** in IDE

3. Now that you have created the html file and the style sheet, link style.css to index.html.

▶ Click here for a hint
▶ Click here for the solution

- Now that you have linked the style sheet, you can start applying the styles to the html page.

4. Set the margin of all the elements to **10px** on all sides and the background-color to **black**

5. Style the `Home` section (having the app name & the nav bar) to have:

- A margin of **-1** to the top.left & right
- Bottom padding of 1 or 2 cm
- Background color to be a lighter shade of **blue magenta**

6. Style the `Header` section as follows:

- Font-size as 30 pixels
- Colour to be light purple
- Floating to the left
- Padding of 2 cms to the left

7. The `Navigation Bar Container` as:

- Floating to the right
- Padding of 1 cm to the right

8. The `Navigation Bar Buttons` can have `static` styling as:

- A light gray color
- Background color as dark purple
- Margin & padding of 10 pixels each. A border radius of 1mm
- Font size as 20 pixels
- Optional text decoration

9. The `Navigation Bar Buttons` can have the below `on-hover` style:

- Color as white
- Bolder font weight
- Background colour as purple
- Optional text decoration

▶ Click to see the content of style.css

10. The 3 conversion sections of the calculator will now be styled:

a. All the sections should have the below styling:

- Items are aligned vertically and positioned at the center.
- Grid-based layout

b. The figure tag can be styled as follows:

- Float to the left
- Alignment (justification) can be automatic
- Width as 200 pixels

c. The image tag can have a width of 200 pixels

d. The figcaption tag should:

- Have a black color
- Font size as 17 pixels
- Text aligned to the centre

e. The conversion buttons can have static styling as:

- Font size of 20px
- A transparent border with border-radius as 40px and the content fitting the page width
- Background-color as light green
- Left and right padding as 15px each

f. The conversion button should have `on-hover` styling of changing the mouse appearance from an arrow to a **pointer**

▶ Click to see the content of style.css

11. The calculator panel will now be styled:

a. The elements in bold `b` class will have:

- Border top style as solid & border top width as 20px, border radius as 10px
- Background-color as white, Width as 600px, height as 400px
- Top and bottom margins as 25 pixels
- Flexible box display

b. The `temparature` class can have the top border color as **green**

c. The `weight` class can have top border colour as **coral**

d. The `distance` class can have top border colour as **cyan**

e. The legend tag can have font-size as 30px with a **bolder** font-weight

f. The panel can have an overall left margin of 50px

▶ Click to see the content of style.css

12. The floating home button will now be styled.

a. The button icon can:

- Have width and height as 40px each
- Items aligned and justified to the center
- Border radius as 100% and background colour as **cyan**
- Flexible box display
- The icon should be fixed and not move with scrolling. It can be at the extreme left and 40 pixels from the bottom of the page

b. The image for the icon button can have a width of 30 px

13. The footer tag can have:

- The background colour as **light pink** & top margin as 20px

14. Save the changes made so far

▶ Click to see the completed code for style.css

15. Set the values in index.html as below:

a. Set the value of the class attribute of the `div` inside inside the nav tag to **topdiv**

b. Set the value of the class attribute of the buttons inside the nav bar to **topmenu**

c. Set the value of the class attribute of the section with id `temperatue` to **b temperature**. You are setting one generic style `b` and a specific style `temperature`

d. Set the value of the class attribute of the section with id `weight` to **b weight**. You are setting one generic style `b` and a specific style `weight`

e. Set the value of the class attribute of the section with id `distance` to **b distance**. You are setting one generic style `b` and a specific style `distance`

f. Set the value of the class attribute of the div with id `go-home` to **iconbutton**

16. Once you have added all the styling to the index.html, save your changes.

▶ Click here to see the completed code for index.html

17. Save index.html file and then view it with Live Server. The page would render as given below.

Note: Kindly ensure that you complete steps 15 and 16 to update the index.html file — this is crucial for the styling to get applied.

Note: You will be using the same index.html and style.css in the upcoming week's hands-on lab: Validating a JavaScript form. Therefore, please ensure to download both files by right-clicking and selecting the download option as highlighted in the below screenshot.

Summary

Congratulations!!! You have now learned how to update the styles for a web page using both inline and internal CSS. Additionally, you learned about the types of CSS: inline, internal, and external, and applied these CSS styling concepts. You have also completed two practice exercises where you applied CSS styling in different ways and enhanced the `Hands-on Lab: Unit Conversion using HTML5 Structural Elements` by applying CSS styling to beautify it.

Don't stop there – we encourage you to create a web page lab about your country and try applying and updating styles using CSS.

Author(s)

Ramesh Sannareddy