

CSS Tutorial

Introduction

Cascading Style Sheets (CSS) describe how HTML elements are displayed on the screen. While HTML structures webpage content, CSS styles and enhances its appearance. CSS is whitespace insensitive.

How to Include CSS

There are two primary ways to add CSS to your webpage:

1. **Inline Style:** Within the HTML <head> using a <style> tag.
2. **External Stylesheet:** Using a separate .css file linked within the HTML <head>.

Basic CSS Syntax

CSS rules follow this general format:

```
selector {  
  property1: value1;  
  property2: value2;  
  /* more properties */  
}
```

Example CSS Rule

Styling a paragraph (<p>):

```
p {  
  color: blue;  
  font-family: Arial;  
}
```

Adding CSS to HTML

Inline Style

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p { color: blue; font-family: Arial; }
  </style>
</head>
<body>
  <p>Welcome to CS373E!</p>
</body>
</html>
```

CSS Notes

- The type="text/css" attribute can be added to the <style> tag, but it is **optional** in modern HTML.
- Comments in CSS are written inside /* */.
Example:
/* This is a comment */
- **Color values** in CSS can be expressed in different formats:
 - **By name:** e.g., blue
 - **By RGB value:**
Format: rgb(red, green, blue)
 - Each parameter (red, green, and blue) is a number between 0 and 255 representing color intensity.
 - Example: rgb(0, 0, 255) is equivalent to the color blue.
 - If using RGB, you can replace a color name with its RGB equivalent.
For example, instead of:
color: blue;
you can write:
color: rgb(0, 0, 255);

External Stylesheet

Create a file named style.css:

```
/* style.css */
```

```
p { color: blue; font-family: Arial; }
```

Link this stylesheet in HTML:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<p>Welcome to CS373E!</p>
```

```
</body>
```

```
</html>
```

Note that **rel** defines the relationship between the current document and the linked document or resource, **href** specifies the URL of the page the link points to, and **type** indicates the media type of the linked document or resource.

Common CSS Properties and Examples

Styling the Background

```
/* style.css */
```

```
body {
```

```
    background-image: url(UTCS.png);
```

```
    /* Sets background image. By default, it repeats both vertically and horizontally */
```

```
    background-repeat: no-repeat;
```

```
    /* Prevents repeating the image */
```

```
    background-position: 0px 0px;
```

```
    /* Positions the background image: 50px from left, 50px from top */
```

```
    background-size: cover;
```

```
    /* Stretches the image to cover the entire page */
```

```
}
```

`/* Units like px, %, em, etc., can be used to define dimensions and spacing */`

Styling a Paragraph

`/* style.css */`

```
p {  
  color: blue;           /* Sets text color */  
  font-family: Arial;    /* Sets font family */  
  font-style: italic;    /* Sets font style to italic */  
  text-indent: 25px;     /* Indents the first line of the paragraph */  
  text-align: left;      /* Specifies the text alignment (left) */  
}
```

Styling Headings

`/* style.css */`

```
h1 {  
  background-color: yellow; /* Sets the background color */  
  border: 3px dashed red;   /* Sets border: width = 3px, style = dashed, color = red */  
}
```

CSS Box Model



The CSS Box Model includes margin, border, padding, and content:

- **Padding:** Space inside border.
- **Border:** Outline around padding and content.
- **Margin:** Space outside border.

```
/* style.css */
```

```
h1 {  
  background-color: yellow;      /* Sets the background color of the heading */  
  border: 3px dashed red;        /* Adds a red dashed border with 3px width */  
  padding: 20px 30px 20px 30px; /* Adds space inside the border: top, right, bottom, left */  
  margin: 50px 50px 50px 50px; /* Adds space outside the border: top, right, bottom, left */  
}
```

List Styling

```
/* style.css */
```

```
ul {  
  color: blue; /* Sets the text color of unordered list items */  
}
```

```
ol {  
  color: green; /* Sets the text color of ordered list items */  
}
```

```
ol li {  
  border: 3px dashed red; /* Adds a red dashed border with 3px width around each list item in an ordered list */  
}
```

Image Styling

```
/* style.css */
```

```
img {  
  width: 250px; /* Sets the image width to 250 pixels */  
  height: 150px; /* Sets the image height to 150 pixels */  
}
```

Hyperlink Styling

```
/* style.css */
a:link {
    color: green;           /* Sets the color of unvisited links */
    text-decoration: none;  /* Removes the underline from unvisited links */
}

a:visited {
    color: brown;          /* Sets the color of visited links */
}

a:hover {
    background-color: blue; /* Sets background color when mouse hovers over the link */
    color: white;           /* Changes link text color on hover */
    text-decoration: underline; /* Underlines the link text on hover */
    font-weight: bold;      /* Makes the link text bold on hover */
}

a:active {
    background-color: orange; /* Sets background color when the link is being clicked */
}
```

ID Styling Example

```
/* style.css */
#firsth1 {
    background-color: orange; /* Sets the background color of the element with
id="firsth1" */
    border: 3px solid brown;  /* Adds a solid brown border with 3px width */
    padding: 20px 30px;      /* Adds padding: 20px top/bottom, 30px left/right */
    margin: 50px 50px 50px 0; /* Adds margin: top = 50px, right = 50px, bottom = 50px,
left = 0 */
}
```

<!-- HTML -->

<h1 id="firsth1">Heading Elements</h1> <!-- Applies styles using the ID selector -->

Class Styling Example

```

/* style.css */
.firsth1 {
  background-color: orange;      /* Sets the background color of any element with
class="firsth1" */
  border: 3px solid brown;      /* Adds a solid brown border with 3px width */
  padding: 20px 30px;          /* Adds padding: 20px top/bottom, 30px left/right */
  margin: 50px 50px 50px 0;    /* Adds margin: top = 50px, right = 50px, bottom = 50px,
left = 0 */
}
<!-- HTML -->
<h1 class="firsth1">Heading Elements</h1> <!-- Applies styles using the class selector -->

```

Positioning Elements in CSS

CSS allows you to control how elements are positioned using the position property. Here's a breakdown of the different values and how they behave:

1. Static (default)

- This is the default positioning for all HTML elements.
- Elements are placed according to the normal document flow: top to bottom, left to right.
- The top, left, right, and bottom properties do not affect statically positioned elements.

2. Relative

- Similar to static positioning, but now the element responds to top, left, right, and bottom.
- The element is positioned relative to its normal location in the document flow.
- Other elements will not move to accommodate the shifted position — they behave as if the element is still in its original place.

3. Absolute

- Removes the element completely from the document flow — it won't affect or be affected by the layout of surrounding elements.
- Positioned relative to the nearest ancestor that has a non-static position (e.g., relative, absolute, or fixed).
- If no such positioned ancestor exists, it is positioned relative to the <html> (the document itself).

4. Fixed

- Similar to absolute, but always positioned relative to the viewport, not any parent element.
- The element stays in place even when the page is scrolled.

- Commonly used for headers, footers, or floating action buttons.

5. Sticky

- Behaves like relative until a specified scroll threshold is reached.
- Once the threshold is passed, the element "sticks" in place and behaves like fixed.
- Useful for sticky navigation bars or headers.

Common Typography Properties

- line-height: Spacing between lines.
- font-weight: Thickness of font (e.g., bold, 500).
- font-style: Font styling (italic).
- text-align: Alignment (center, left, right, justify).
- text-indent: Indent first line of text.

References

- [W3Schools CSS Intro](#)
- [CSS Positioning Tutorial](#)