

Cascading Style Sheets

Kameswari Chebrolu
Department of CSE
IIT Bombay



<https://www.memedroid.com/memes/detail/4191421/HTML-vs-CSS?refGallery=tags&page=1&tag=css>

Cascading Style Sheets (CSS)

- HTML designed to define structure and semantics of a document, not so much presentation
 - Initially, developers used tags like ``, ``, and `<i>` to apply styles
 - Messy and hard-to-maintain code
 - Not easy to update/change visual design across pages without altering many files!
- Separate document content from presentation
 - led to development of CSS
 - Principle of separation of concerns

- Style sheet language used for describing the presentation of a document
 - Presentation: Layout, colors, fonts etc
 - spacing between elements, how elements are positioned and laid out
 - what background images or background colors are to be used
 - Can also help create animations, responsive web designs
 - Responsive: adapt to different screen sizes and devices
- Saves effort, can control the layout of multiple web pages all at once

Outline

- Including CSS in HTML
- Selectors and Declarations (building blocks)
- Color
- Text
- Styling Lists
- Styling Forms

CSS Building Blocks

- CSS associates style rules with HTML elements
- How to select elements? → **Selector**
- How the selected element should be styled (rule)? → **Declaration**
 - Declaration in turn made of **property:value**
 - Several properties can be specified in one declaration
 - Same declaration can apply to many selectors separated by commas

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;}
```



This rule indicates that all <h1>, <h2> and <h3> elements should be shown in the Arial typeface, in a yellow color

Including CSS

- Three main ways to include CSS in an HTML document
- Inline CSS: applied directly to an HTML element using the style attribute
 - Useful for applying styles to a single element without affecting the rest of the document
- Internal CSS (Embedded CSS): written within the <style> tag in the <head> section of the HTML document
 - Useful for styling a single HTML document without affecting other pages

- External CSS: CSS is placed in a separate .css file linked via a <link> tag
 - Ideal for styling multiple web pages with the same styles
 - Allows for easier maintenance and consistency
- One can combine all three methods. Precedence?
 - Inline CSS overrides internal and external CSS
 - Internal CSS overrides external CSS
 - External CSS is the least specific, but most versatile for site-wide styling

- You can also use `@import` to include an external stylesheet in your current stylesheet
 - E.g `@import url("styles.css")` or
 - Commonly used to import base styles, frameworks, or theme styles before the main CSS file's rules are applied
 - You can use either relative or absolute URLs in the `@import` rule

```
<html>
<head>
  <title>Combined CSS Example </title>
  <link rel="stylesheet" href="linking-styles.css" > <!-- External CSS
-->
  <style>
    p {
      color: green; /* Internal CSS */
    }
  </style>
</head>
<body>
  <h1 style="color: red;">Hello, World! </h1> <!-- Inline CSS -->
  <p>This paragraph is styled with both internal and external CSS. </p>
</body>
</html>
```

```
/* Importing an external stylesheet */
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap' );

/* Main styles */
body {
    font-family: 'Roboto', Arial, sans-serif; /* Using the imported font */
    background-color: #e4e9cb;
}

h1 {
    color: #333;
}

p {
    color: #4724d1;
    font-weight: bold;
}
```

(linking-styles.css file)

Hello, World!

This paragraph is styled with both internal and external CSS.

Selector

- CSS selectors are patterns used to select specific elements in an HTML document
 - Can select based on their name, attributes, classes, IDs, and more
- Many different types of CSS selectors:

Basic Selectors:

- Type Selector: Targets all elements of a specific type
 - E.g., <p>, <h1>, or <div> elements
- Class Selector: Targets elements with a specific class attribute
 - A class selector is prefixed with a dot “.”
- ID Selector: Targets a single element with a specific ID attribute
 - An ID selector is prefixed with a hash “#”
- Universal Selector: Targets all elements on the page
 - It is represented by an asterisk “*”

```
<html>
<head>
  <title>Basic CSS Selectors Example</title>
  <link rel="stylesheet" href="selector-basic.css">
</head>
<html>
  <h1 class="title">Welcome to the Basic CSS Selectors Example</h1>

  <div class="content">
    <h2>Element Selector</h2>
    <p>This paragraph is styled using the element selector. All paragraph elements are affected.</p>

    <h2>Class Selector</h2>
    <p class="highlight">This paragraph is styled using a class selector (only elements with the
`highlight` class are affected).</p>

    <h2>ID Selector</h2>
    <p id="unique">This paragraph is styled using an ID selector (only the element with the ID
`unique` is affected).</p>

  </div>
</html>
```

```
/* Element Selector: Targets all <p> elements */
p {
    color: #555;
    font-size: 16px;
}

/* Class Selector: Targets elements with the class "highlight" */
.highlight {
    background-color: yellow;
    font-weight: bold;
}

/* ID Selector: Targets the element with the ID "unique" */
#unique {
    color: blue;
    font-size: 18px;
    text-decoration: underline;
}

/* Grouping Selector: Targets only h2 and p */
h1, p {
    font-style: italic;
}

/* Additional Styling for .content */
.content {
    background-color: #f0f0f0;
}
```


Welcome to the Basic CSS Selectors

Example

Element Selector

This paragraph is styled using the element selector. All paragraph elements are affected.

Class Selector

This paragraph is styled using a class selector (only elements with the `highlight` class are affected).

ID Selector

This paragraph is styled using an ID selector (only the element with the ID `unique` is affected).

Outline

- Including CSS in HTML
- Selectors and Declarations (building blocks)
- Color
- Text
- Styling Lists
- Styling Forms

Color

- Brings life to a page
- CSS provides various ways to define and manipulate colors

Properties

- color: Controls the text color of an element
- background-color: Controls the color of the element's background

CSS Color Representation

- Named Colors: CSS supports a list of predefined color names, such as red, blue, green, lightyellow, tomato etc
 - 147 in total
- Hexadecimal Colors: Defined using a hash (#) followed by a six-digit hexadecimal number
 - #ff6347 for Tomato
- RGB Colors: Defined using the rgb() function with red, green, and blue components
 - Values range from 0 to 255
 - rgb(255, 99, 71) for Tomato

```
<html>
<head>
  <title>CSS Color Formats</title>
</head>
<body>
  <!-- Text Colors -->
  <div>
    Text color using a named color: <span style="color: coral;">coral</span>
  </div>
  <div>
    Text color using a hex color: <span style="color: #4682b4;">steelblue</span>
  </div>
  <div>
    Text color using an RGB color: <span style="color: rgb(255, 99, 71);">tomato</span>
  </div>
  <!-- Background Colors -->
  <div style="background-color: lightpink; color: black;" >
    Background color using a named color: lightpink
  </div>
  <div style="background-color: #ff6347; color: white;" >
    Background color using a hex color: #ff6347 (tomato)
  </div>
  <div style="background-color: rgb(100, 149, 237); color: white;" >
    Background color using an RGB color: rgb(100, 149, 237) (cornflowerblue)
  </div>
</body>
</html>
```

Text color using a named color: coral

Text color using a hex color: steelblue

Text color using an RGB color: tomato

Background color using a named color: lightpink

Background color using a hex color: #ff6347 (tomato)

Background color using an RGB color: rgb(100, 149, 237) (cornflowerblue)

Text

CSS provides a wide range of properties to control text formatting

Font Related Properties

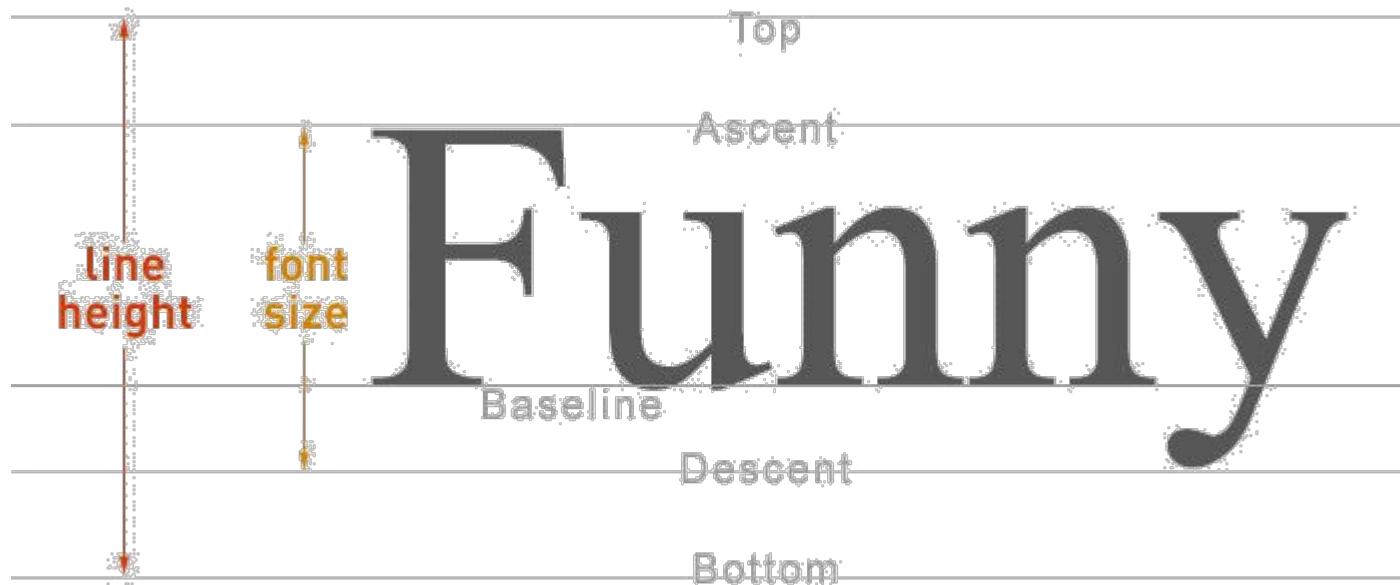
- font-family: Specifies the typeface to be used for text
 - Value is the name of the typeface
 - Note: People viewing the page need to have this typeface installed on computer
 - Can include multiple fonts as a fallback mechanism
 - Can also end with a generic font name for the type of font (e.g. sans-serif)

- font-size: Sets the size of the font
 - Can be defined in various units such as pixels (px), ems (em), or percentages (%)
 - Default size of text in browsers is 16px
 - Percentages: a size of 75% would be the equivalent of 12px
 - em is equivalent to the width of a letter m

Font Size Mapping Table

| Text Type | Pixels (px) | Percentage (%) | Em (em) |
|------------|-------------|----------------|---------|
| Body Text | 16px | 100% | 1em |
| Small Text | 14px | 87.5% | 0.875em |
| H1 | 32px | 200% | 2em |
| H2 | 24px | 150% | 1.5em |
| H3 | 20px | 125% | 1.25em |
| H4 | 18px | 112.5% | 1.125em |
| H5 | 16px | 100% | 1em |
| H6 | 14px | 87.5% | 0.875em |
| Caption | 12px | 75% | 0.75em |
| Subscript | 10px | 62.5% | 0.625em |

- font-weight: Controls the boldness of the text
 - Common values include normal, bold, bolder, lighter, or numerical values like 400 (normal) and 700 (bold)
- font-style: Specifies the style of the font, such as normal, italic, or oblique
- Line-height: Defines the height of a line of text
 - Influences spacing between lines
 - Best given as a unitless number that acts as a multiplier of the font size
 - line-height: 1.5 → if font size is 16px, the line height would be 24px



```
<html>
<head>
  <title>Font Styling with Units Example</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      font-size: 16px; /* Base font size */
      line-height: 1.6; /* Unitless value, scaling with font size */
    }

    h1 {
      font-family: 'Georgia', serif;
      font-size: 2em; /* Font size relative to parent element's font size */
      font-weight: bold;
      font-style: oblique; /* Oblique text style */
      line-height: 120%; /* Percentage of the font size */
    }
  </style>

```

```
p {
    font-family: 'Verdana', sans-serif;
    font-size: 0.875em; /* Font size relative to parent element's font size */
    font-weight: normal;
    line-height: 1.8em; /* Line height in em units */
}

.highlight {
    font-weight: bold;
    font-style: italic;
}

</style>
</head>
<body>
    <h1>Font Styling with CSS</h1>
    <p>This is a paragraph demonstrating different font properties in CSS with various units.</p>
    <p class="highlight">This is a highlighted paragraph with bold and italic styling.</p>
    <a href="https://www.example.com" target="_blank">Example Website</a>
</body>
</html>
```

Font Styling with CSS

This is a paragraph demonstrating different font properties in CSS with various units.

This is a highlighted paragraph with bold and italic styling.

[Example Website](#)

Text Alignment and Decoration

- `text-align`: Aligns the text horizontally
 - Possible values include left, right, center, and justify
 - When you have several paragraphs of text, it is best to use left

- **text-decoration:** Adds decorative lines to text
 - Possible values are underline, overline, line-through, or none
 - Blink (flashes text on/off) was deprecated in CSS3
- **text-transform:** Controls the capitalization of text
 - Possible values include none, capitalize, uppercase, and lowercase
 - Capitalize causes the first letter of each word to appear capitalize

- **text-indent:** Indents the first line of a text block by a specified amount
 - Amount to indent can be specified in a number of ways but is usually given in pixels or ems
 - It can take a negative value as well
- **letter-spacing:** Adjusts the space between characters in a text
 - Useful when heading or sentence is all in uppercase
 - Value is specified in ems or px, added on top of default spacing
- **word-spacing:** Controls the space between words

- Text-shadow (CSS3): Applies a shadow to the text
 - It accepts values for horizontal offset, vertical offset, blur radius, and color
 - h-offset: Positive values move the shadow to the right, and negative values move it to the left
 - v-offset: Positive values move the shadow down, and negative values move it up
 - blur-radius: A higher value creates a more blurred and softer shadow
 - If omitted, the default is 0, meaning the shadow is sharp
 - color: color of the shadow
 - Can be any valid color value (e.g., named colors, HEX, RGB, RGBA, HSL, HSLA)

```
<html>
<head>
  <title>Text Styling Example</title>
  <style>
    .text-example {
      text-align: left; /* Aligns text to the left */
      text-decoration: underline; /* Underlines the text */
      text-transform: uppercase; /* Transforms text to uppercase */
      text-indent: 20px; /* Indents the first line of text by 20px */
      letter-spacing: 2px; /* Sets spacing between letters to 2px */
      word-spacing: 5px; /* Sets spacing between words to 5px */
      text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Adds a shadow to the text */
      color: #333; /* Sets text color to a dark gray */
      font-family: Arial, sans-serif; /* Sets font family */
    }
  </style>
</head>
<body>
  <div class="text-example">
    This is an example of text styling with CSS. Notice the alignment, decoration, transformation,
    indentation, spacing, and shadow applied to this text.
  </div>
</body>
</html>
```

THIS IS AN EXAMPLE OF TEXT STYLING WITH CSS.
NOTICE THE ALIGNMENT, DECORATION,
TRANSFORMATION, INDENTATION, SPACING, AND
SHADOW APPLIED TO THIS TEXT.

Styling Lists

- List-style property is used to style lists
 - How list items are marked or numbered?
- list-style shorthand property can set:
 - list-style-type: Defines the type of marker (bullet or number) used for the list items
 - list-style-position: Defines the position of the marker relative to the list item content
 - list-style-image: Defines an image to use as the marker

- list-style-type Values:
 - Unordered Lists ():
 - disc: Solid circle (default for unordered lists)
 - circle: Hollow circle
 - square: Solid square
 - Ordered Lists ():
 - decimal: Numbers (default for ordered lists)
 - decimal-leading-zero: Numbers with leading zeros (e.g., 01, 02)
 - lower-roman: Lowercase Roman numerals (i, ii, iii)
 - upper-roman: Uppercase Roman numerals (I, II, III)
 - lower-alpha: Lowercase letters (a, b, c)
 - upper-alpha: Uppercase letters (A, B, C)
 - none: No numbering

- list-style-position Values:
 - inside: marker is inside the list item box
 - Text will wrap around the marker.
 - outside: marker is outside the list item box
 - Text will not wrap around the marker (default value).

```
<html>
<head>
  <title>List Style Properties</title>
  <style>

    /* Different list-style-types */
    ul.circle {
      list-style-type: circle;
    }

    ul.square {
      list-style-type: square;
    }
  </style>
</head>
<body>
  <ul>
    <li>List Style Properties</li>
  </ul>
</body>
</html>
```

```
ul.upper-roman {
    list-style-type: upper-roman;
}

ul.lower-alpha {
    list-style-type: lower-alpha;
}

/* Position properties */
ul.inside {
    list-style-position: inside;
}

ul.outside {
    list-style-position: outside;
}
</style>
</head>
<body>

<h1>List Style Properties</h1>

<h2>List Style Type</h2>
<ul class="circle">
    <li>Circle bullet 1</li>
    <li>Circle bullet 2</li>
</ul>
```

```
</ul>
```

```
<ul class="square">
```

```
  <li>Square bullet 1</li>
```

```
  <li>Square bullet 2</li>
```

```
</ul>
```

```
<ul class="upper-roman">
```

```
  <li>Upper Roman numeral 1</li>
```

```
  <li>Upper Roman numeral 2</li>
```

```
</ul>
```

```
<ul class="lower-alpha">
```

```
  <li>Lower Alpha bullet a</li>
```

```
  <li>Lower Alpha bullet b</li>
```

```
</ul>
```

<h2>List Style Position</h2>

<p>Outside (default):</p>

<ul class="outside">

Outside position: This is a long text that will wrap to the next line. Notice how the bullet point remains outside the text block, and the text aligns itself after the bullet point.

Outside position: Another long text to show how the text aligns when the list-style-position is set to outside. The bullet is outside the block of text, and the text starts on a new line after the bullet.

<p>Inside:</p>

<ul class="inside">

Inside position: This is a long text that will wrap to the next line. Notice how the bullet point is inside the text block, and the text aligns itself right next to the bullet point.

Inside position: Another long text to show how the text aligns when the list-style-position is set to inside. The bullet is placed inside the text block, and the text continues right after the bullet.

</body>

</html>

List Style Properties

List Style Type

- Circle bullet 1
- Circle bullet 2

- Square bullet 1
- Square bullet 2

- I. Upper Roman numeral 1
- II. Upper Roman numeral 2

- a. Lower Alpha bullet a
- b. Lower Alpha bullet b

List Style Position

Outside (default):

- Outside position: This is a long text that will wrap to the next line. Notice how the bullet point remains outside the text block, and the text aligns itself after the bullet point.
- Outside position: Another long text to show how the text aligns when the list-style-position is set to outside. The bullet is outside the block of text, and the text starts on a new line after the bullet.

Inside:

- Inside position: This is a long text that will wrap to the next line. Notice how the bullet point is inside the text block, and the text aligns itself right next to the bullet point.
- Inside position: Another long text to show how the text aligns when the list-style-position is set to inside. The bullet is placed inside the text block, and the text continues right after the bullet.

Styling Forms

- Can be styled based on already covered concepts
 - Font-size, color, background-color, border etc

- `<fieldset>` and `<legend>` elements are used together to group and label a set of form controls
 - make forms more organized and easier to understand
- `<fieldset>` Element: groups related elements within a form
 - Often used to wrap input fields, checkboxes, radio buttons, or any other form controls that belong together logically
 - By default, browsers render `<fieldset>` with a border around it
- `<legend>` Element: provides a caption or title for the `<fieldset>`
 - Placed at the top of the `<fieldset>` by default
 - `<legend>` text is typically rendered slightly larger and bolder than normal text
 - Positioned within the border of the `<fieldset>`
 - Can customize its appearance with CSS

```
<html>
<head>
  <title>Styled Form Example</title>
  <style>
    body {
      font-family: Arial, sans-serif; /* Font for the page */
      background-color: #f4f4f4; /* Page background color */
      padding: 20px; /* Padding around the page */
    }

    form {
      background-color: #ffffff; /* Form background color */
      border: 2px solid #007BFF; /* Form border */
      border-radius: 8px; /* Rounded corners for the form */
      padding: 20px; /* Padding inside the form */
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Optional shadow for better visibility */
      max-width: 500px; /* Max width for the form */
      margin: auto; /* Center the form */
    }

    fieldset {
      border: 1px solid #007BFF; /* Border color of fieldset */
      border-radius: 6px; /* Rounded corners for fieldset */
      padding: 10px; /* Padding inside fieldset */
      margin-bottom: 20px; /* Margin below fieldset */
      background-color: #f9f9f9; /* Background color of fieldset */
    }
  </style>
</head>
<body>
```

```
legend {
    font-size: 1.2em; /* Font size for the legend */
    color: #007BFF; /* Text color for the legend */
    font-weight: bold; /* Font weight for the legend */
}

label {
    display: block; /* Block display for labels */
    margin-bottom: 8px; /* Margin below labels */
    font-size: 1em; /* Font size for labels */
    color: #333; /* Text color for labels */
}

input[type="text"], input[type="email"], textarea {
    width: 100%; /* Full width for text inputs and textarea */
    padding: 8px; /* Padding inside inputs and textarea */
    border: 1px solid #ccc; /* Border color for inputs and textarea */
    border-radius: 4px; /* Rounded corners for inputs and textarea */
    box-sizing: border-box; /* Include padding and border in the element's total width and height */
    font-size: 1em; /* Font size for inputs and textarea */
    color: #333; /* Text color for inputs and textarea */
    background-color: #ffffff; /* Background color for inputs and textarea */
}

input[type="text"]:focus, input[type="email"]:focus, textarea:focus {
    border-color: #007BFF; /* Border color when focused */
    box-shadow: 0 0 5px rgba(0, 123, 255, 0.5); /* Shadow effect when focused */
    outline: none; /* Remove default outline */
}
```

```
button {  
  
    background-color: #007BFF; /* Button background color */  
    color: #ffffff; /* Button text color */  
    border: none; /* Remove default border */  
    padding: 10px 15px; /* Padding inside button */  
    border-radius: 4px; /* Rounded corners for button */  
    font-size: 1em; /* Font size for button */  
    cursor: pointer; /* Pointer cursor on hover */  
}  
  
button:hover {  
    background-color: #0056b3; /* Darker background color on hover */  
    transform: scale(1.05); /* Slightly enlarge button on hover */  
}  
</style>  
</head>
```

```
<body>
  <form>
    <fieldset>
      <legend>Contact Information</legend>
      <span>Name:</span>
      <input type="text" id="name" name="name" placeholder="Enter your name" required>
      <span>Email:</span>
      <input type="email" id="email" name="email" placeholder="Enter your email" required>
      <span>Message:</span>
      <textarea id="message" name="message" rows="5" placeholder="Enter your message"
required></textarea>
      <button type="submit">Submit</button>
    </fieldset>
  </form>
</body>

</html>
```

Contact Information

Name:

Email:

Message:

Summary

- CSS is also a vast field, just covered a small subset
- Reference:
<https://www.w3schools.com/css/default.asp>