# Here's a list of 100 HTML tags with examples to understand the concepts better:

1 <h1> to <h6>: Headings
2 <p>: Paragraph
3 <a>: Anchor (Link)
4 <img>: Image
5 <ul>: Unordered List
6 <ol>: Ordered List
7 <li>: List Item
8 <div>: Division or Section
9 <span>: Inline Section
10 <br>: Line Break
11 <hr>: Horizontal Rule
12 <strong>: Bold Text
13 <em>: Italicized Text
14 <table>: Table
15 <thead>: Table Header
16 <tbody>: Table Body
17 <tfoot>: Table Footer
18 <tr>: Table Row
19 <th>: Table Header Cell
20 <td>: Table Cell
21 <form>: Form
22 <input>: Input Field
23 <button>: Button
24 <textarea>: Text Area
25 <select>: Drop-down List
26 <option>: Option in Drop-down
27 <label>: Label for Form Elements
28 <fieldset>: Group Related Elements in a Form
29 <legend>: Caption for <fieldset>
30 <nav>: Navigation Links
31 <header>: Header Section
32 <footer>: Footer Section
33 <article>: Article Content
34 <section>: Section of a Document
35 <aside>: Content Aside from Main Content
36 <details>: Disclosure Widget
37 <summary>: Visible Summary of <details>

38 <mark>: Highlighted Text
39 <code>: Inline Code
40 <pre>: Preformatted Text
41 <blockquote>: Block Quotation
42 <cite>: Citation
43 <abbr>: Abbreviation
44 <address>: Contact Information
45 <b>: Bold Text (Stylistic)
46 <i>: Italic Text (Stylistic)
47 <small>: Smaller Text
48 <sub>: Subscript Text
49 <sup>: Superscript Text
50 <bdo>: Bi-directional Override
51 <audio>: Audio Content
52 <video>: Video Content
53 <source>: Media Source
54 <track>: Text Tracks for Media
55 <canvas>: Canvas for Graphics
56 <svg>: Scalable Vector Graphics
57 <iframe>: Inline Frame
58 <embed>: Embedded Content
59 <object>: Embedded Object
60 <param>: Parameters for <object>
61 <ins>: Inserted Text
62 <del>: Deleted Text
63 <figure>: Group of Media Content
64 <figcaption>: Caption for <figure>
65 <time>: Date/Time
66 <meter>: Scalar Measurement
67 <progress>: Progress Indicator
68 <output>: Output of Calculation
69 <dialog>: Dialog Box
70 <data>: Machine-readable Data
71 <var>: Variable
72 <samp>: Sample Output
73 <kbd>: Keyboard Input
74 <s>: Strikethrough Text
75 <u>: Underlined Text
76 <wbr>: Word Break Opportunity
77 <map>: Image Map
78 <area>: Area in Image Map
79 <link>: Relationship Metadata
80 <style>: CSS Styles

81. **<meta>: Metadata**
82. **<title>: Document Title**
83. **<base>: Base URL**
84. **<noscript>: Fallback Content for No JavaScript**
85. **<script>: JavaScript**
86. **<template>: HTML Template**
87. **<slot>: Placeholder in a Template**
88. **<shadow>: Shadow DOM**
89. **<content>: Deprecated - Shadow DOM**
90. **<picture>: Responsive Images**
91. **<datalist>: List of Predefined Options**
92. **<keygen>: Deprecated - Key Pair Generator**
93. **<bdi>: Bi-directional Isolation**
94. **<main>: Main Content**
95. **<rp>: Fallback for Non-supported Ruby**
96. **<rt>: Ruby Text**
97. **<ruby>: Ruby Annotation**
98. **<section>: Document Section**
99. **<template>: HTML Template**
100. **<track>: Text Tracks for Media**

Below are examples for some of these tags:

**1. <h1> to <h6>: Headings**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Heading Tags</title>
</head>
<body>
   <h1>This is an H1 heading</h1>
   <h2>This is an H2 heading</h2>
   <h3>This is an H3 heading</h3>
   <h4>This is an H4 heading</h4>
   <h5>This is an H5 heading</h5>
   <h6>This is an H6 heading</h6>
</body>
</html>
```

### 2. <p>: Paragraph

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Paragraph Tag</title>
</head>
<body>
   <p>This is a paragraph of text.</p>
</body>
</html>
```

### 3. <a>: Anchor (Link)

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Anchor Tag</title>
</head>
<body>
   <a href="https://www.example.com">Visit Example.com</a>
</body>
</html>
```

### 4. <img>: Image

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Image Tag</title>
</head>
<body>
   <img src="https://www.example.com/image.jpg" alt="Example
Image">
</body>
</html>
```

### 5. <ul>: Unordered List

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Unordered List</title>
</head>
<body>
    <ul>
        <li>List Item 1</li>
        <li>List Item 2</li>
        <li>List Item 3</li>
    </ul>
</body>
</html>
```

### 6. <ol>: Ordered List

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Ordered List</title>
</head>
<body>
    <ol>
        <li>List Item 1</li>
        <li>List Item 2</li>
        <li>List Item 3</li>
    </ol>
</body>
</html>
```

### 7. <li>: List Item

html

```
<!DOCTYPE html>
<html>
<head>
    <title>List Item</title>
</head>
<body>
    <ul>
        <li>List Item 1</li>
        <li>List Item 2</li>
        <li>List Item 3</li>
    </ul>
</body>
</html>
```

**8. <div>: Division or Section**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Division Tag</title>
</head>
<body>
    <div>This is a division.</div>
</body>
</html>
```

**9. <span>: Inline Section**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Span Tag</title>
</head>
<body>
    <p>This is a <span>span element</span> inside a paragraph.</p>
</body>
</html>
```

**10. <br>: Line Break**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Line Break</title>
</head>
<body>
    <p>This is a line.<br>This is a new line.</p>
</body>
```

```
</html>
```

### 11. <hr>: Horizontal Rule

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Horizontal Rule</title>
</head>
<body>
   <p>This is a paragraph.</p>
   <hr>
   <p>This is another paragraph after a horizontal rule.</p>
</body>
</html>
```

### 12. <strong>: Bold Text

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Strong Tag</title>
</head>
<body>
   <p>This is <strong>important</strong> text.</p>
</body>
</html>
```

### 13. <em>: Italicized Text

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Em Tag</title>
</head>
<body>
   <p>This is <em>emphasized</em> text.</p>
</body>
```

```
</html>
```

**14. <table>: Table**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Table Tag</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Heading 1</th>
            <th>Heading 2</th>
        </tr>
        <tr>
            <td>Data 1</td>
            <td>Data 2</td>
        </tr>
    </table>
</body>
</html>
```

**15. <thead>: Table Header**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Thead Tag</title>
</head>
<body>
    <table border="1">
        <thead>
            <tr>
                <th>Heading 1</th>
                <th>Heading 2</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Data 1</td>
                <td>Data 2</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

### 16. <tbody>: Table Body

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tbody Tag</title>
</head>
<body>
    <table border="1">
        <thead>
            <tr>
                <th>Heading 1</th>
                <th>Heading 2</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Data 1</td>
                <td>Data 2</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

### 17. <tfoot>: Table Footer

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tfoot Tag</title>
</head>
<body>
    <table border="1">
```

```
        <thead>
            <tr>
                <th>Heading 1</th>
                <th>Heading 2</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Data 1</td>
                <td>Data 2</td>
            </tr>
        </tbody>
        <tfoot>
            <tr>
                <td>Footer 1</td>
                <td>Footer 2</td>
            </tr>
        </tfoot>
    </table>
</body>
</html>
```

### 18. <tr>: Table Row

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Tr Tag</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Heading 1</th>
            <th>Heading 2</th>
        </tr>
        <tr>
            <td>Data 1</td>
            <td>Data 2</td>
        </tr>
    </table>
</body>
</html>
```

### 19. <th>: Table Header Cell

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Th Tag</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Heading 1</th>
            <th>Heading 2</th>
        </tr>
        <tr>
            <td>Data 1</td>
            <td>Data 2</td>
        </tr>
    </table>
</body>
</html>
```

### 20. <td>: Table Cell

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Td Tag</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Heading 1</th>
            <th>Heading 2</th>
        </tr>
        <tr>
            <td>Data 1</td>
            <td>Data 2</td>
        </tr>
    </table>
</body>
```

```
</html>
```

### 21. <form>: Form

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Form Tag</title>
</head>
<body>
   <form action="/submit" method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name"><br><br>
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

### 22. <input>: Input Field

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Input Tag</title>
</head>
<body>
   <form action="/submit" method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name"><br><br>
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

### 23. <button>: Button

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Button Tag</title>
```

```
</head>
<body>
    <button type="button" onclick="alert('Button Clicked!')">Click Me!</
button>
</body>
</html>
```

**24. <textarea>: Text Area**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Textarea Tag</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="message">Message:</label><br>
        <textarea id="message" name="message" rows="4" cols="50"></
textarea><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

**25. <select>: Drop-down List**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Select Tag</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="cars">Choose a car:</label>
        <select id="cars" name="cars">
            <option value="volvo">Volvo</option>
            <option value="saab">Saab</option>
            <option value="mercedes">Mercedes</option>
            <option value="audi">Audi</option>
        </select><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

### 26. <option>: Option in Drop-down

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Option Tag</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="cars">Choose a car:</label>
        <select id="cars" name="cars">
            <option value="volvo">Volvo</option>
            <option value="saab">Saab</option>
            <option value="mercedes">Mercedes</option>
            <option value="audi">Audi</option>
        </select><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

### 27. <label>: Label for Form Elements

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Label Tag</title>
</head>
<body>
    <form action="/submit" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

**28. <fieldset>: Group Related Elements in a Form**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Fieldset Tag</title>
</head>
<body>
  <form action="/submit" method="post">
    <fieldset>
      <legend>Personal Information</legend>
      <label for="name">Name:</label>
      <input type="text" id="name" name="name"><br><br>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email"><br><br>
    </fieldset>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

**29. <legend>: Caption for <fieldset>**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Legend Tag</title>
</head>
<body>
  <form action="/submit" method="post">
    <fieldset>
      <legend>Personal Information</legend>
      <label for="name">Name:</label>
      <input type="text" id="name" name="name"><br><br>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email"><br><br>
    </fieldset>
    <input type="submit" value="Submit">
  </form>
</body>
```

```
</html>
```

**30. <nav>: Navigation Links**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Nav Tag</title>
</head>
<body>
   <nav>
      <a href="#home">Home</a> |
      <a href="#about">About</a> |
      <a href="#contact">Contact</a>
   </nav>
</body>
</html>
```

**31. <header>: Header Section**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Header Tag</title>
</head>
<body>
   <header>
      <h1>My Website</h1>
      <p>Welcome to my website!</p>
   </header>
</body>
</html>
```

**32. <footer>: Footer Section**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Footer Tag</title>
</head>
```

```
<body>
    <footer>
        <p>Footer content goes here.</p>
    </footer>
</body>
</html>
```

**33. <article>: Article Content**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Article Tag</title>
</head>
<body>
    <article>
        <h2>Article Title</h2>
        <p>Article content goes here.</p>
    </article>
</body>
</html>
```

**34. <section>: Section of a Document**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Section Tag</title>
</head>
<body>
    <section>
        <h2>Section Title</h2>
        <p>Section content goes here.</p>
    </section>
</body>
</html>
```

**35. <aside>: Content Aside from Main Content**

html

```
<!DOCTYPE html>
```

```
<html>
<head>
    <title>Aside Tag</title>
</head>
<body>
    <aside>
        <h2>Aside Title</h2>
        <p>Aside content goes here.</p>
    </aside>
</body>
</html>
```

**36. <details>: Disclosure Widget**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Details Tag</title>
</head>
<body>
    <details>
        <summary>Details Summary</summary>
        <p>Additional details go here.</p>
    </details>
</body>
</html>
```

**37. <summary>: Visible Summary of <details>**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Summary Tag</title>
</head>
<body>
    <details>
        <summary>Details Summary</summary>
        <p>Additional details go here.</p>
    </details>
</body>
</html>
```

### 38. <mark>: Highlighted Text

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Mark Tag</title>
</head>
<body>
   <p>This is <mark>highlighted</mark> text.</p>
</body>
</html>
```

### 39. <code>: Inline Code

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Code Tag</title>
</head>
<body>
   <p>This is an example of <code>inline code</code>.</p>
</body>
</html>
```

### 40. <pre>: Preformatted Text

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Pre Tag</title>
</head>
<body>
   <pre>
   This is preformatted text.
   It preserves whitespace and
   line breaks.
   </pre>
</body>
</html>
```

**41. <blockquote>: Block Quotation**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Blockquote Tag</title>
</head>
<body>
  <blockquote>
    <p>This is a blockquote.</p>
  </blockquote>
</body>
</html>
```

**42. <cite>: Citation**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Cite Tag</title>
</head>
<body>
  <p><cite>Author Name</cite></p>
</body>
</html>
```

**43. <abbr>: Abbreviation**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Abbr Tag</title>
</head>
<body>
  <p><abbr title="HyperText Markup Language">HTML</abbr> is a
markup language for creating web pages.</p>
</body>
</html>
```

**44. <address>: Contact Information**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Address Tag</title>
</head>
<body>
   <address>
      Written by <a href="mailto:webmaster@example.com">Jon Doe</a>.<br>
      Visit us at:<br>
      Example.com<br>
      Box 564, Disneyland<br>
      USA
   </address>
</body>
</html>
```

**45. <b>: Bold Text (Stylistic)**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>B Tag</title>
</head>
<body>
   <p>This is <b>bold</b> text.</p>
</body>
</html>
```

**46. <i>: Italic Text (Stylistic)**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>I Tag</title>
</head>
<body>
   <p>This is <i>italic</i> text.</p>
</body>
```

```
</html>
```

**47. <small>: Smaller Text**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Small Tag</title>
</head>
<body>
   <p>This is <small>smaller</small> text.</p>
</body>
</html>
```

**48. <sub>: Subscript Text**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Sub Tag</title>
</head>
<body>
   <p>This is <sub>subscript</sub> text.</p>
</body>
</html>
```

**49. <sup>: Superscript Text**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Sup Tag</title>
</head>
<body>
   <p>This is <sup>superscript</sup> text.</p>
</body>
</html>
```

**50. <bdo>: Bi-directional Override**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Bdo Tag</title>
</head>
<body>
  <p>Text in LTR: <bdo dir="ltr">This is left-to-right text.</bdo></p>
  <p>Text in RTL: <bdo dir="rtl">This is right-to-left text.</bdo></p>
</body>
</html>
```

### 51. <audio>: Audio Content

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Audio Tag</title>
</head>
<body>
  <audio controls>
    <source src="audiofile.mp3" type="audio/mp3">
    Your browser does not support the audio element.
  </audio>
</body>
</html>
```

### 52. <video>: Video Content

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Video Tag</title>
</head>
<body>
  <video width="320" height="240" controls>
    <source src="videofile.mp4" type="video/mp4">
    Your browser does not support the video tag.
  </video>
</body>
</html>
```

**53. <source>: Media Source**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Source Tag</title>
</head>
<body>
   <video width="320" height="240" controls>
      <source src="videofile.mp4" type="video/mp4">
      <source src="videofile.ogg" type="video/ogg">
      Your browser does not support the video tag.
   </video>
</body>
</html>
```

**54. <track>: Text Tracks for Media**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Track Tag</title>
</head>
<body>
   <video width="320" height="240" controls>
      <source src="videofile.mp4" type="video/mp4">
      <track src="subtitles_en.vtt" kind="subtitles" srclang="en"
label="English">
      Your browser does not support the video tag.
   </video>
</body>
</html>
```

**55. <canvas>: Canvas for Graphics**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Canvas Tag</title>
```

```
</head>
<body>
   <canvas id="myCanvas" width="200" height="100" style="border:1px
solid #000000;"></canvas>
   <script>
      var c = document.getElementById("myCanvas");
      var ctx = c.getContext("2d");
      ctx.moveTo(0, 0);
      ctx.lineTo(200, 100);
      ctx.stroke();
   </script>
</body>
</html>
```

### 56. <svg>: Scalable Vector Graphics

html

```
<!DOCTYPE html>
<html>
<head>
   <title>SVG Tag</title>
</head>
<body>
   <svg width="100" height="100">
      <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3"
fill="red" />
   </svg>
</body>
</html>
```

### 57. <iframe>: Inline Frame

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Iframe Tag</title>
</head>
<body>
   <iframe src="https://www.example.com" width="600" height="400"></
iframe>
</body>
</html>
```

### 58. <embed>: Embedded Content

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Embed Tag</title>
</head>
<body>
   <embed src="helloworld.swf" width="400" height="300">
</body>
</html>
```

### 59. <object>: External Resource

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Object Tag</title>
</head>
<body>
   <object data="helloworld.swf" width="400" height="300"></object>
</body>
</html>
```

### 60. <param>: Parameters for Object

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Param Tag</title>
</head>
<body>
   <object data="helloworld.swf" width="400" height="300">
      <param name="autoplay" value="true">
   </object>
</body>
</html>
```

### 61. <noscript>: Fallback for No Scripts

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Noscript Tag</title>
</head>
<body>
  <noscript>
    <p>JavaScript is not enabled in your browser.</p>
  </noscript>
</body>
</html>
```

### 62. <script>: Script

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Script Tag</title>
</head>
<body>
  <script>
    alert('Hello, World!');
  </script>
</body>
</html>
```

### 63. <del>: Deleted Text

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Del Tag</title>
</head>
<body>
  <p>This is <del>deleted</del> text.</p>
</body>
</html>
```

### 64. <ins>: Inserted Text

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Ins Tag</title>
</head>
<body>
   <p>This is <ins>inserted</ins> text.</p>
</body>
</html>
```

**65. <s>: Strikethrough Text**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>S Tag</title>
</head>
<body>
   <p>This is <s>strikethrough</s> text.</p>
</body>
</html>
```

**66. <dfn>: Defining Instance**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Dfn Tag</title>
</head>
<body>
   <p>The <dfn>HTML</dfn> specification is the standard for creating web pages.</p>
</body>
</html>
```

**67. <kbd>: Keyboard Input**

html

```
<!DOCTYPE html>
<html>
```

```
<head>
   <title>Kbd Tag</title>
</head>
<body>
   <p>Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy.</p>
</body>
</html>
```

### 68. <samp>: Sample Output

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Samp Tag</title>
</head>
<body>
   <p>The computer said <samp>Hello, World!</samp></p>
</body>
</html>
```

### 69. <var>: Variable

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Var Tag</title>
</head>
<body>
   <p>The area of a circle is <var>πr²</var>.</p>
</body>
</html>
```

### 70. <progress>: Progress Bar

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Progress Tag</title>
</head>
<body>
```

```
    <progress value="70" max="100"></progress>
</body>
</html>
```

**71. <meter>: Scalar Measurement**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Meter Tag</title>
</head>
<body>
    <p>Disk usage: <meter value="0.6">60%</meter></p>
</body>
</html>
```

**72. <time>: Date/Time**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Time Tag</title>
</head>
<body>
    <p>The meeting is scheduled for <time
datetime="2023-07-19T14:00">2:00 PM on July 19, 2023</time>.</p>
</body>
</html>
```

**73. <output>: Output Result**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Output Tag</title>
</head>
<body>
    <form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
        <input type="range" id="a" value="50"> +
        <input type="number" id="b" value="50"> =
        <output name="result" for="a b">100</output>
    </form>
</body>
</html>
```

**74. <figure>: Self-contained Content**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Figure Tag</title>
</head>
<body>
    <figure>
        <img src="image.jpg" alt="Example Image">
        <figcaption>An example image.</figcaption>
    </figure>
</body>
</html>
```

**75. <figcaption>: Caption for <figure>**

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Figcaption Tag</title>
</head>
<body>
    <figure>
        <img src="image.jpg" alt="Example Image">
        <figcaption>An example image.</figcaption>
    </figure>
</body>
</html>
```

**76. <dialog>: Dialog Box**

html

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Dialog Tag</title>
</head>
<body>
    <dialog id="myDialog">
        <p>This is a dialog box.</p>
        <button
onclick="document.getElementById('myDialog').close()">Close</button>
    </dialog>
    <button
onclick="document.getElementById('myDialog').showModal()">Open
Dialog</button>
</body>
</html>
```

### 77. <summary>: Summary of <details>

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Summary Tag</title>
</head>
<body>
    <details>
        <summary>Details Summary</summary>
        <p>Additional details go here.</p>
    </details>
</body>
</html>
```

### 78. <main>: Main Content

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Main Tag</title>
</head>
<body>
    <main>
        <h1>Main Content</h1>
        <p>This is the main content of the document.</p>
    </main>
```

```
</body>
</html>
```

**79. <template>: HTML Template**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Template Tag</title>
</head>
<body>
   <template id="myTemplate">
      <p>This is a template content.</p>
   </template>
   <script>
      var template = document.getElementById('myTemplate');
      document.body.appendChild(template.content.cloneNode(true));
   </script>
</body>
</html>
```

**80. <slot>: Placeholder for Web Components**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Slot Tag</title>
</head>
<body>
   <template id="myTemplate">
      <slot name="mySlot">Default content</slot>
   </template>
   <script>
      var template = document.getElementById('myTemplate');
      document.body.appendChild(template.content.cloneNode(true));
   </script>
</body>
</html>
```

**81. <picture>: Responsive Images**

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Picture Tag</title>
</head>
<body>
  <picture>
    <source srcset="image-400.jpg" media="(max-width: 400px)">
    <source srcset="image-800.jpg" media="(max-width: 800px)">
    <img src="image.jpg" alt="Example Image">
  </picture>
</body>
</html>
```

### 82. <wbr>: Word Break Opportunity

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Wbr Tag</title>
</head>
<body>
  <p>Supercalifragilistic<wbr>expialidocious</p>
</body>
</html>
```

### 83. <data>: Machine-readable Data

html

```
<!DOCTYPE html>
<html>
<head>
  <title>Data Tag</title>
</head>
<body>
  <p>Product ID: <data value="12345">ABC123</data></p>
</body>
</html>
```

### 84. <bdi>: Bi-directional Isolation

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Bdi Tag</title>
</head>
<body>
   <p>User <bdi>John</bdi> posted a new comment.</p>
</body>
</html>
```

**85. <ruby>: Ruby Annotation**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Ruby Tag</title>
</head>
<body>
   <p>漢 <ruby>字 <rt>Kanji</rt></ruby> are used in Japanese.</p>
</body>
</html>
```

**86. <rp>: Fallback for Ruby Annotation**

html

```
<!DOCTYPE html>
<html>
<head>
   <title>Rp Tag</title>
</head>
<body>
   <p>漢 <ruby>字 <rp>(</rp><rt>Kanji</rt><rp>)</rp></ruby> are used in Japanese.</p>
</body>
</html>
```

# Here’s a list of 100 CSS properties, along with brief explanations and

# examples:

**color**

*Sets the color of the text.*

css

```
p {

  color: blue;
}
```

**background-color**

*Sets the background color of an element.*

css

```
div {

  background-color: yellow;
}
```

**font-size**

*Sets the size of the font.*

css

```
h1 {

  font-size: 24px;
}
```

**font-family**

*Sets the font of the text.*

css

```
body {

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

### font-weight

*Sets the weight (thickness) of the font.*

css

```
strong {

  font-weight: bold;
}
```

### font-style

*Sets the style of the font (italic, oblique).*

css

```
em {

  font-style: italic;
}
```

### text-align

*Aligns the text inside an element.*

css

```
p {

  text-align: center;
}
```

### text-decoration

*Sets decoration on text (underline, overline, line-through, none).*

css

```
a {

  text-decoration: none;
}
```

**text-transform**

*Transforms the case of the text.*

css

```
h2 {

  text-transform: uppercase;
}
```

**letter-spacing**

*Sets the space between characters.*

css

```
h3 {

  letter-spacing: 2px;
}
```

**line-height**

*Sets the distance between lines of text.*

css

```
p {

  line-height: 1.5;
}
```

**margin**

*Sets the margin space around an element.*

css

```
div {

  margin: 10px;
}
```

**padding**
*Sets the padding space inside an element.*
css

```
div {

  padding: 15px;
}
```

**border**
*Sets the border around an element.*
css

```
div {

  border: 1px solid black;
}
```

**border-radius**
*Sets the radius for the element's corners.*
css

```
div {

  border-radius: 10px;
}
```

**width**
*Sets the width of an element.*
css

```
img {

  width: 100px;
}
```

**height**

*Sets the height of an element.*

css

```
img {

  height: 100px;
}
```

**display**

*Sets the display behavior of an element (block, inline, none, etc.).*

css

```
span {

  display: block;
}
```

**position**

*Sets the positioning method of an element (static, relative, absolute, fixed, sticky).*

css

```
div {

  position: absolute;
}
```

**top**

*Sets the top position of a positioned element.*

css

```
div {

  position: absolute;
  top: 20px;
}
```

**right**
*Sets the right position of a positioned element.*
css

```
div {

  position: absolute;
  right: 20px;
}
```

**bottom**
*Sets the bottom position of a positioned element.*
css

```
div {

  position: absolute;
  bottom: 20px;
}
```

**left**
*Sets the left position of a positioned element.*
css

```
div {

  position: absolute;
  left: 20px;
}
```

**float**
*Sets the float behavior of an element (left, right, none).*
css

```
img {

  float: left;
}
```

**clear**
*Sets the clearing of floats (left, right, both, none).*
css

```
div {

  clear: both;
}
```

**overflow**
*Sets what to do when content overflows an element's box (visible, hidden, scroll, auto).*
css

```
div {

  overflow: hidden;
}
```

**visibility**
*Sets the visibility of an element (visible, hidden, collapse).*
css

```
div {

  visibility: hidden;
}
```

**z-index**
*Sets the stack order of positioned elements.*
css

```
div {

  position: absolute;
  z-index: 10;
}
```

**opacity**
*Sets the opacity level of an element.*
css

```
div {

  opacity: 0.5;
}
```

**background-image**
*Sets an image as the background of an element.*
css

```
div {

  background-image: url('image.jpg');
}
```

**background-size**
*Sets the size of the background image.*
css

```
div {

  background-image: url('image.jpg');
  background-size: cover;
}
```

**background-repeat**

*Sets the repetition of the background image.*

css

```
div {

  background-image: url('image.jpg');
  background-repeat: no-repeat;
}
```

**background-position**

*Sets the position of the background image.*

css

```
div {

  background-image: url('image.jpg');
  background-position: center;
}
```

**border-color**

*Sets the color of the border.*

css

```
div {

  border: 1px solid;
  border-color: red;
}
```

**border-style**

*Sets the style of the border (solid, dashed, dotted, etc.).*

css

```
div {

  border: 2px;
  border-style: dashed;
}
```

**border-width**

*Sets the width of the border.*

css

```
div {

  border-width: 5px;
}
```

**border-top**

*Sets the border on the top side of an element.*

css

```
div {

  border-top: 2px solid black;
}
```

**border-right**

*Sets the border on the right side of an element.*

css

```
div {

  border-right: 2px solid black;
}
```

**border-bottom**

*Sets the border on the bottom side of an element.*

css

```
div {

  border-bottom: 2px solid black;
}
```

**border-left**

*Sets the border on the left side of an element.*

css

```
div {

  border-left: 2px solid black;
}
```

**border-collapse**

*Sets whether table borders are collapsed into a single border or separated.*

css

```
table {

  border-collapse: collapse;
}
```

**border-spacing**

*Sets the distance between the borders of adjacent cells in a table.*

css

```
table {

  border-spacing: 10px;
}
```

**box-shadow**

*Sets the shadow of an element.*

css

```
div {

  box-shadow: 5px 5px 10px gray;
}
```

### text-shadow

*Sets the shadow of the text.*

css

```
h1 {

  text-shadow: 2px 2px 4px gray;
}
```

### transform

*Applies a 2D or 3D transformation to an element.*

css

```
div {

  transform: rotate(45deg);
}
```

### transition

*Defines the transition effects when changing property values.*

css

```
div {

  transition: all 0.5s ease;
}
```

### animation

*Defines the animation properties for an element.*

css

```
@keyframes example {

  from { background-color: red; }
  to { background-color: blue; }
}

div {
  animation: example 5s infinite;
}
```

**overflow-x**

*Sets the horizontal overflow behavior.*

css

```
div {

  overflow-x: auto;
}
```

**overflow-y**

*Sets the vertical overflow behavior.*

css

```
div {

  overflow-y: hidden;
}
```

**white-space**

*Sets how whitespace inside an element is handled.*

css

```
p {

  white-space: nowrap;
}
```

**text-align-last**

*Sets the alignment of the last line of a block-level element.*

css

```
p {

  text-align-last: right;
}
```

**direction**

*Sets the text direction (ltr, rtl).*

css

```
p {

  direction: rtl;
}
```

**writing-mode**

*Sets the direction in which text and blocks flow.*

css

```
p {

  writing-mode: vertical-rl;
}
```

**text-indent**

*Sets the indentation of the first line of text.*

css

```
p {

  text-indent: 20px;
}
```

**list-style**

*Sets the list style type, position, and image.*

css

```
ul {

  list-style: square inside;
}
```

**list-style-type**

*Sets the type of marker in a list.*

css

```
ul {

  list-style-type: circle;
}
```

**list-style-position**

*Sets the position of the list markers.*

css

```
ul {

  list-style-position: inside;
}
```

**list-style-image**

*Sets an image as a list item marker.*

css

```
ul {

  list-style-image: url('marker.png');
}
```

**visibility**

*Sets whether an element is visible or hidden.*

css

```
div {

  visibility: visible;
}
```

### text-overflow

*Specifies how overflowed text is signaled (clip, ellipsis).*

css

```
p {

  text-overflow: ellipsis;
}
```

### letter-spacing

*Sets the spacing between characters.*

css

```
p {

  letter-spacing: 1px;
}
```

### word-spacing

*Sets the spacing between words.*

css

```
p {

  word-spacing: 5px;
}
```

### border-image

*Sets an image as the border of an element.*

css

```
div {

  border-image: url('border.png') 30 round;
}
```

**border-image-source**

*Sets the source of the border image.*

css

```
div {

  border-image-source: url('border.png');
}
```

**border-image-slice**

*Defines how to slice the border image.*

css

```
div {

  border-image-slice: 20;
}
```

**border-image-width**

*Sets the width of the border image.*

css

```
div {

  border-image-width: 5px;
}
```

**border-image-repeat**

*Sets how the border image is repeated.*

```css
div {

  border-image-repeat: round;
}
```

**clip**
*Defines a clipping region for an element.*

```css
div {

  clip: rect(10px, 100px, 100px, 10px);
}
```

**clip-path**
*Defines a clipping path for an element.*

```css
div {

  clip-path: circle(50%);
}
```

**mask**
*Sets an image or element to use as a mask.*

```css
div {

  mask: url('mask.svg');
}
```

**mask-image**
*Sets an image to use as a mask.*

css

```
div {

  mask-image: url('mask.svg');
}
```

**mask-size**
*Sets the size of the mask image.*
css

```
div {

  mask-size: cover;
}
```

**mask-repeat**
*Sets the repeat behavior of the mask image.*
css

```
div {

  mask-repeat: no-repeat;
}
```

**mask-position**
*Sets the position of the mask image.*
css

```
div {

  mask-position: center;
}
```

**visibility**
*Sets the visibility of an element (visible, hidden, collapse).*

css

```
div {

  visibility: hidden;
}
```

**opacity**
*Sets the opacity of an element.*
css

```
div {

  opacity: 0.5;
}
```

**resize**
*Sets whether an element can be resized.*
css

```
textarea {

  resize: both;
}
```

**box-sizing**
*Sets how the width and height of an element are calculated.*
css

```
div {

  box-sizing: border-box;
}
```

**overflow-wrap**
*Sets how the browser should handle text overflow.*

css

```
p {

  overflow-wrap: break-word;
}
```

### hyphens

*Sets how hyphenation is applied to text.*

css

```
p {

  hyphens: auto;
}
```

### text-align

*Sets the horizontal alignment of text.*

css

```
p {

  text-align: justify;
}
```

### text-align-last

*Sets the alignment of the last line of a block element.*

css

```
p {

  text-align-last: right;
}
```

### text-transform

*Transforms the case of text.*

css

```
p {

  text-transform: capitalize;
}
```

**text-indent**
*Sets the indentation of the first line of text.*
css

```
p {

  text-indent: 20px;
}
```

**font-variant**
*Sets the font variant (e.g., small-caps).*
css

```
p {

  font-variant: small-caps;
}
```

**font-feature-settings**
*Controls the use of OpenType features.*
css

```
p {

  font-feature-settings: "smcp";
}
```

**font-kerning**
*Sets the kerning of the font.*

css

```
p {

  font-kerning: normal;
}
```

**font-variation-settings**
*Sets font variations for variable fonts.*
css

```
p {

  font-variation-settings: "wght" 700;
}
```

**letter-spacing**
*Sets the spacing between characters.*
css

```
p {

  letter-spacing: 2px;
}
```

**line-height**
*Sets the height of a line box.*
css

```
p {

  line-height: 1.5;
}
```

**word-break**
*Sets how words should break when reaching the end of a line.*

css

```
p {

  word-break: break-all;
}
```

**word-wrap**

*Sets whether long words should be broken and wrap onto the next line.*

css

```
p {

  word-wrap: break-word;
}
```

**text-justify**

*Sets the justification of text.*

css

```
p {

  text-justify: inter-word;
}
```

**text-outline**

*Sets the outline of text.*

css

```
h1 {

  text-shadow: 1px 1px 2px black;
}
```

**text-rendering**

*Optimizes text rendering.*

css

```
p {

  text-rendering: optimizeLegibility;
}
```

**transform-origin**
*Sets the origin for transformations.*
css

```
div {

  transform-origin: center;
}
```

**transform-style**
*Defines how nested elements are rendered in 3D space.*
css

```
div {

  transform-style: preserve-3d;
}
```

**perspective**
*Defines the perspective from which 3D elements are viewed.*
css

```
div {

  perspective: 1000px;
}
```

**perspective-origin**
*Sets the origin point for perspective.*

css

```
div {

  perspective-origin: center;
}
```

**backface-visibility**
*Defines whether the back face of an element is visible when facing the user.*
css div { backface-visibility: hidden; }

Here’s a list of 100 CSS properties, along with brief explanations and examples:

**color**
*Sets the color of the text.*
css

```
p {

  color: blue;
}
```

**background-color**
*Sets the background color of an element.*
css

```
div {

  background-color: yellow;
}
```

**font-size**
*Sets the size of the font.*
css

```
h1 {
  font-size: 24px;
}
```

**font-family**
*Sets the font of the text.*
css

```
body {

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

**font-weight**
*Sets the weight (thickness) of the font.*
css

```
strong {

  font-weight: bold;
}
```

**font-style**
*Sets the style of the font (italic, oblique).*
css

```
em {

  font-style: italic;
}
```

**text-align**
*Aligns the text inside an element.*
css

```
p {
  text-align: center;
}
```

### text-decoration

*Sets decoration on text (underline, overline, line-through, none).*

css

```
a {

  text-decoration: none;
}
```

### text-transform

*Transforms the case of the text.*

css

```
h2 {

  text-transform: uppercase;
}
```

### letter-spacing

*Sets the space between characters.*

css

```
h3 {

  letter-spacing: 2px;
}
```

### line-height

*Sets the distance between lines of text.*

css

```
p {
  line-height: 1.5;
}
```

**margin**

*Sets the margin space around an element.*

css

```
div {

  margin: 10px;
}
```

**padding**

*Sets the padding space inside an element.*

css

```
div {

  padding: 15px;
}
```

**border**

*Sets the border around an element.*

css

```
div {

  border: 1px solid black;
}
```

**border-radius**

*Sets the radius for the element's corners.*

css

```
div {
  border-radius: 10px;
}
```

**width**
*Sets the width of an element.*
css

```
img {

  width: 100px;
}
```

**height**
*Sets the height of an element.*
css

```
img {

  height: 100px;
}
```

**display**
*Sets the display behavior of an element (block, inline, none, etc.).*
css

```
span {

  display: block;
}
```

**position**
*Sets the positioning method of an element (static, relative, absolute, fixed, sticky).*
css

```
div {

  position: absolute;
}
```

### top

*Sets the top position of a positioned element.*

css

```
div {

  position: absolute;
  top: 20px;
}
```

### right

*Sets the right position of a positioned element.*

css

```
div {

  position: absolute;
  right: 20px;
}
```

### bottom

*Sets the bottom position of a positioned element.*

css

```
div {

  position: absolute;
  bottom: 20px;
}
```

### left

*Sets the left position of a positioned element.*

css

```
div {

  position: absolute;
  left: 20px;
}
```

### float

*Sets the float behavior of an element (left, right, none).*

css

```
img {

  float: left;
}
```

### clear

*Sets the clearing of floats (left, right, both, none).*

css

```
div {

  clear: both;
}
```

### overflow

*Sets what to do when content overflows an element's box (visible, hidden, scroll, auto).*

css

```
div {

  overflow: hidden;
}
```

**visibility**

*Sets the visibility of an element (visible, hidden, collapse).*

css

```
div {

  visibility: hidden;
}
```

**z-index**

*Sets the stack order of positioned elements.*

css

```
div {

  position: absolute;
  z-index: 10;
}
```

**opacity**

*Sets the opacity level of an element.*

css

```
div {

  opacity: 0.5;
}
```

**background-image**

*Sets an image as the background of an element.*

css

```
div {

  background-image: url('image.jpg');
}
```

**background-size**

*Sets the size of the background image.*

css

```
div {

  background-image: url('image.jpg');
  background-size: cover;
}
```

**background-repeat**

*Sets the repetition of the background image.*

css

```
div {

  background-image: url('image.jpg');
  background-repeat: no-repeat;
}
```

**background-position**

*Sets the position of the background image.*

css

```
div {

  background-image: url('image.jpg');
  background-position: center;
}
```

**border-color**

*Sets the color of the border.*

css

```
div {
  border: 1px solid;
  border-color: red;
}
```

### border-style

*Sets the style of the border (solid, dashed, dotted, etc.).*

css

```
div {

  border: 2px;
  border-style: dashed;
}
```

### border-width

*Sets the width of the border.*

css

```
div {

  border-width: 5px;
}
```

### border-top

*Sets the border on the top side of an element.*

css

```
div {

  border-top: 2px solid black;
}
```

### border-right

*Sets the border on the right side of an element.*

css

```
div {

  border-right: 2px solid black;
}
```

**border-bottom**
*Sets the border on the bottom side of an element.*
css

```
div {

  border-bottom: 2px solid black;
}
```

**border-left**
*Sets the border on the left side of an element.*
css

```
div {

  border-left: 2px solid black;
}
```

**border-collapse**
*Sets whether table borders are collapsed into a single border or separated.*
css

```
table {

  border-collapse: collapse;
}
```

**border-spacing**
*Sets the distance between the borders of adjacent cells in a table.*
css

```
table {

  border-spacing: 10px;
}
```

**box-shadow**
*Sets the shadow of an element.*
css

```
div {

  box-shadow: 5px 5px 10px gray;
}
```

**text-shadow**
*Sets the shadow of the text.*
css

```
h1 {

  text-shadow: 2px 2px 4px gray;
}
```

**transform**
*Applies a 2D or 3D transformation to an element.*
css

```
div {

  transform: rotate(45deg);
}
```

**transition**
*Defines the transition effects when changing property values.*
css

```
div {

  transition: all 0.5s ease;
}
```

**animation**
*Defines the animation properties for an element.*
css

```
@keyframes example {

  from { background-color: red; }
  to { background-color: blue; }
}

div {
  animation: example 5s infinite;
}
```

**overflow-x**
*Sets the horizontal overflow behavior.*
css

```
div {

  overflow-x: auto;
}
```

**overflow-y**
*Sets the vertical overflow behavior.*
css

```
div {

  overflow-y: hidden;
}
```

**white-space**

*Sets how whitespace inside an element is handled.*

css

```
p {

  white-space: nowrap;
}
```

**text-align-last**

*Sets the alignment of the last line of a block-level element.*

css

```
p {

  text-align-last: right;
}
```

**direction**

*Sets the text direction (ltr, rtl).*

css

```
p {

  direction: rtl;
}
```

**writing-mode**

*Sets the direction in which text and blocks flow.*

css

```
p {

  writing-mode: vertical-rl;
}
```

### text-indent

*Sets the indentation of the first line of text.*

css

```
p {

  text-indent: 20px;
}
```

### list-style

*Sets the list style type, position, and image.*

css

```
ul {

  list-style: square inside;
}
```

### list-style-type

*Sets the type of marker in a list.*

css

```
ul {

  list-style-type: circle;
}
```

### list-style-position

*Sets the position of the list markers.*

css

```
ul {

  list-style-position: inside;
}
```

**list-style-image**

*Sets an image as a list item marker.*

```css
ul {

  list-style-image: url('marker.png');
}
```

**visibility**

*Sets whether an element is visible or hidden.*

```css
div {

  visibility: visible;
}
```

**text-overflow**

*Specifies how overflowed text is signaled (clip, ellipsis).*

```css
p {

  text-overflow: ellipsis;
}
```

**letter-spacing**

*Sets the spacing between characters.*

```css
p {

  letter-spacing: 1px;
}
```

**word-spacing**
*Sets the spacing between words.*
css

```
p {

  word-spacing: 5px;
}
```

**border-image**
*Sets an image as the border of an element.*
css

```
div {

  border-image: url('border.png') 30 round;
}
```

**border-image-source**
*Sets the source of the border image.*
css

```
div {

  border-image-source: url('border.png');
}
```

**border-image-slice**
*Defines how to slice the border image.*
css

```
div {

  border-image-slice: 20;
}
```

### border-image-width

*Sets the width of the border image.*

css

```
div {

  border-image-width: 5px;
}
```

### border-image-repeat

*Sets how the border image is repeated.*

css

```
div {

  border-image-repeat: round;
}
```

### clip

*Defines a clipping region for an element.*

css

```
div {

  clip: rect(10px, 100px, 100px, 10px);
}
```

### clip-path

*Defines a clipping path for an element.*

css

```
div {

  clip-path: circle(50%);
}
```

**mask**

*Sets an image or element to use as a mask.*

css

```
div {

  mask: url('mask.svg');
}
```

**mask-image**

*Sets an image to use as a mask.*

css

```
div {

  mask-image: url('mask.svg');
}
```

**mask-size**

*Sets the size of the mask image.*

css

```
div {

  mask-size: cover;
}
```

**mask-repeat**

*Sets the repeat behavior of the mask image.*

css

```
div {

  mask-repeat: no-repeat;
}
```

**mask-position**

*Sets the position of the mask image.*

css

```
div {

  mask-position: center;
}
```

**visibility**

*Sets the visibility of an element (visible, hidden, collapse).*

css

```
div {

  visibility: hidden;
}
```

**opacity**

*Sets the opacity of an element.*

css

```
div {

  opacity: 0.5;
}
```

**resize**

*Sets whether an element can be resized.*

css

```
textarea {

  resize: both;
}
```

**box-sizing**
*Sets how the width and height of an element are calculated.*
css

```
div {

  box-sizing: border-box;
}
```

**overflow-wrap**
*Sets how the browser should handle text overflow.*
css

```
p {

  overflow-wrap: break-word;
}
```

**hyphens**
*Sets how hyphenation is applied to text.*
css

```
p {

  hyphens: auto;
}
```

**text-align**
*Sets the horizontal alignment of text.*
css

```
p {

  text-align: justify;
}
```

**text-align-last**

*Sets the alignment of the last line of a block element.*

css

```
p {

  text-align-last: right;
}
```

**text-transform**

*Transforms the case of text.*

css

```
p {

  text-transform: capitalize;
}
```

**text-indent**

*Sets the indentation of the first line of text.*

css

```
p {

  text-indent: 20px;
}
```

**font-variant**

*Sets the font variant (e.g., small-caps).*

css

```
p {

  font-variant: small-caps;
}
```

**font-feature-settings**

*Controls the use of OpenType features.*

css

```
p {

  font-feature-settings: "smcp";
}
```

**font-kerning**

*Sets the kerning of the font.*

css

```
p {

  font-kerning: normal;
}
```

**font-variation-settings**

*Sets font variations for variable fonts.*

css

```
p {

  font-variation-settings: "wght" 700;
}
```

**letter-spacing**

*Sets the spacing between characters.*

css

```
p {

  letter-spacing: 2px;
}
```

**line-height**
*Sets the height of a line box.*
css

```
p {

  line-height: 1.5;
}
```

**word-break**
*Sets how words should break when reaching the end of a line.*
css

```
p {

  word-break: break-all;
}
```

**word-wrap**
*Sets whether long words should be broken and wrap onto the next line.*
css

```
p {

  word-wrap: break-word;
}
```

**text-justify**
*Sets the justification of text.*
css

```
p {

  text-justify: inter-word;
}
```

**text-outline**

*Sets the outline of text.*

css

```
h1 {

  text-shadow: 1px 1px 2px black;
}
```

**text-rendering**

*Optimizes text rendering.*

css

```
p {

  text-rendering: optimizeLegibility;
}
```

**transform-origin**

*Sets the origin for transformations.*

css

```
div {

  transform-origin: center;
}
```

**transform-style**

*Defines how nested elements are rendered in 3D space.*

css

```
div {

  transform-style: preserve-3d;
}
```

**perspective**

*Defines the perspective from which 3D elements are viewed.*

css

```
div {

  perspective: 1000px;
}
```

**perspective-origin**

*Sets the origin point for perspective.*

css

```
div {

  perspective-origin: center;
}
```

**backface-visibility**

*Defines whether the back face of an element is visible when facing the user.*

css div { backface-visibility: hidden; }

**align-items**

*Aligns items along the cross axis in a flex container.*

css .container { display: flex; align-items: center; }

**align-content**

*Aligns flex container lines within the flex container.*

css .container { display: flex; align-content: space-between; }

**align-self**

*Aligns a single flex item along the cross axis.*

css .item { align-self: flex-end; }

**justify-content**

*Aligns items along the main axis in a flex container.*

css .container { display: flex; justify-content: space-around; }

**flex-direction**
*Defines the direction flex items are placed in the flex container.*
css .container { display: flex; flex-direction: column; }

**flex-wrap**
*Defines whether flex items should wrap onto multiple lines.*
css .container { display: flex; flex-wrap: wrap; }

**flex-flow**
*Shorthand for flex-direction and flex-wrap.*
css .container { display: flex; flex-flow: row wrap; }

**flex**
*Defines the flex-grow, flex-shrink, and flex-basis properties in one declaration.*
css .item { flex: 1 1 auto; }

**flex-grow**
*Defines the ability for a flex item to grow relative to the rest of the flex items.*
css .item { flex-grow: 2; }

**flex-shrink**
*Defines the ability for a flex item to shrink relative to the rest of the flex items.*
css .item { flex-shrink: 1; }

**flex-basis**
*Defines the default size of an element before the remaining space is distributed.*
css .item { flex-basis: 200px; }

**grid-template-columns**
*Defines the number and size of columns in a grid layout.*
css .container { display: grid; grid-template-columns: 1fr 2fr; }

**grid-template-rows**
*Defines the number and size of rows in a grid layout.*
css .container { display: grid; grid-template-rows: 100px auto; }

**grid-column**
*Specifies how many columns an element should span in a grid layout.*
css .item { grid-column: span 2; }

**grid-row**
*Specifies how many rows an element should span in a grid layout.*

css .item { grid-row: span 3; }

**grid-area**
*Defines a grid item’s size and location within a grid layout.*
css .item { grid-area: 1 / 1 / 2 / 3; }

**grid-template-areas**
*Defines named grid areas for a grid layout.*
css .container { display: grid; grid-template-areas: "header header" "sidebar content"; }

**grid-auto-columns**
*Defines the size of columns in an implicit grid.*
css .container { display: grid; grid-auto-columns: 100px; }

**grid-auto-rows**
*Defines the size of rows in an implicit grid.*
css .container { display: grid; grid-auto-rows: 100px; }

**grid-auto-flow**
*Controls how auto-placed items are flowed into the grid.*
css .container { display: grid; grid-auto-flow: column; }

**justify-items**
*Aligns grid items along the inline axis within their grid area.*
css .container { display: grid; justify-items: center; }

**align-items**
*Aligns grid items along the block axis within their grid area.*
css .container { display: grid; align-items: start; }

**justify-content**
*Aligns the grid container’s items along the inline axis.*
css .container { display: grid; justify-content: space-between; }

**align-content**
*Aligns the grid container’s items along the block axis.*
css .container { display: grid; align-content: space-around; }

**justify-self**
*Aligns a grid item along the inline axis within its grid area.*
css .item { justify-self: end; }

**object-fit**
*Specifies how an element's content should fit within its box.*

css img { object-fit: cover; }

**object-position**
*Specifies the position of an element's content within its box.*
css img { object-position: center top; }

**mask-type**
*Defines whether a mask is applied as a luminance or alpha mask.*
css .masked-element { mask-type: alpha; }

**will-change**
*Hints to the browser which properties will change, helping optimize performance.*
css div { will-change: transform; }

**contain**
*Defines the size and layout containment for an element.*
css .container { contain: layout style; }

**contain-intrinsic-size**
*Defines an intrinsic size for an element with containment.*
css .element { contain-intrinsic-size: 100px 100px; }

**writing-mode**
*Sets the direction in which text and blocks flow.*
css p { writing-mode: vertical-rl; }

**text-shadow**
*Applies shadow effects to text.*
css h1 { text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }

**text-transform**
*Controls the capitalization of text.*
css p { text-transform: uppercase; }

**text-decoration**
*Sets the text decorations (e.g., underline, overline).*
css a { text-decoration: underline; }

**text-decoration-line**
*Specifies the text decoration lines (e.g., underline, line-through).*
css p { text-decoration-line: line-through; }

**text-decoration-style**
*Sets the style of text decoration lines (e.g., solid, dashed).*

css p { text-decoration-style: dashed; }

**text-decoration-color**
*Sets the color of the text decoration line.*
css p { text-decoration-color: red; }

**border-radius**
*Defines the radius of the element's corners.*
css div { border-radius: 10px; }

**border-top-left-radius**
*Defines the radius for the top-left corner of an element.*
css div { border-top-left-radius: 20px; }

**border-top-right-radius**
*Defines the radius for the top-right corner of an element.*
css div { border-top-right-radius: 20px; }

**border-bottom-left-radius**
*Defines the radius for the bottom-left corner of an element.*
css div { border-bottom-left-radius: 20px; }

**border-bottom-right-radius**
*Defines the radius for the bottom-right corner of an element.*
css div { border-bottom-right-radius: 20px; }

**border-collapse**
*Specifies whether table borders should collapse into a single border.*
css table { border-collapse: collapse; }

**border-spacing**
*Defines the distance between table borders.*
css table { border-spacing: 10px; }

**caption-side**
*Specifies the position of a table caption.*
css table { caption-side: bottom; }

**empty-cells**
*Controls the appearance of empty cells in a table.*
css table { empty-cells: show; }

**table-layout**
*Specifies the algorithm used to lay out table cells, rows, and columns.*
css table { table-layout: fixed; }

**list-style-type**
*Sets the type of list item marker.*
css ul { list-style-type: square; }

**list-style-position**
*Sets the position of the list markers.*
css ul { list-style-position: inside; }

**list-style-image**
*Sets an image as a list item marker.*
css ul { list-style-image: url('marker.png'); }

**box-shadow**
*Applies shadow effects around an element's frame.*
css div { box-shadow: 5px 5px 10px rgba(0, 0, 0, 0.3); }

**filter**
*Applies graphical effects such as blur or color shifting to an element.*
css img { filter: grayscale(100%); }

**backdrop-filter**
*Applies graphical effects to the area behind an element.*
css .overlay { backdrop-filter: blur(5px); }

**transform**
*Applies a 2D or 3D transformation to an element.*
css div { transform: rotate(45deg); }

**transform-style**
*Defines how nested elements are rendered in 3D space.*
css .container { transform-style: preserve-3d; }

**perspective**
*Defines the perspective from which 3D elements are viewed.*
css .container { perspective: 1000px; }

**perspective-origin**
*Sets the origin point for perspective.*
css .container { perspective-origin: center; }

**backface-visibility**
*Defines whether the back face of an element is visible when facing the user.*
css .flip { backface-visibility: hidden; }

**animation**

*Shorthand for setting multiple animation properties.*
css .animate { animation: slide 2s ease-in-out infinite; }

**animation-name**
*Defines the name of the keyframes to be used for the animation.*
css .animate { animation-name: slide; }

**animation-duration**
*Defines the duration of the animation.*
css .animate { animation-duration: 2s; }

**animation-timing-function**
*Defines the speed curve of the animation.*
css .animate { animation-timing-function: ease-in; }

**animation-delay**
*Defines a delay before the animation starts.*
css .animate { animation-delay: 1s; }

**animation-iteration-count**
*Defines the number of times an animation cycle should be played.*
css .animate { animation-iteration-count: infinite; }

**animation-direction**
*Defines whether an animation should play forwards, backwards, or alternate.*
css .animate { animation-direction: alternate; }

**animation-fill-mode**
*Defines what values are applied to the element when the animation is not playing.*
css .animate { animation-fill-mode: forwards; }

**animation-play-state**
*Defines whether the animation is running or paused.*
css .animate { animation-play-state: paused; }

**@keyframes**
*Defines the keyframes for animations.*
css @keyframes slide { 0% { transform: translateX(0); } 100% { transform: translateX(100px); } }

**grid**
*Defines a grid container.*
css .container { display: grid; }

**flex**
*Defines a flex container.*
css .container { display: flex; }

**inline-flex**
*Defines an inline-level flex container.*
css .container { display: inline-flex; }

**inline-grid**
*Defines an inline-level grid container.*
css .container { display: inline-grid; }

**block**
*Defines an element as a block-level element.*
css div { display: block; }

**inline**
*Defines an element as an inline element.*
css span { display: inline; }

**inline-block**
*Defines an element as an inline-level block container.*
css .box { display: inline-block; }

**none**
*Removes the element from the document flow.*
css .hidden { display: none; }

**visibility**
*Defines whether an element is visible or hidden (but still takes up space).*
css .hidden { visibility: hidden; }

**overflow-x**
*Controls the overflow of content along the horizontal axis.*
css .container { overflow-x: auto; }

**overflow-y**
*Controls the overflow of content along the vertical axis.*
css .container { overflow-y: scroll; }

**overflow**
*Shorthand for setting overflow-x and overflow-y.*
css .container { overflow: hidden; }

**resize**

*Defines if and how an element can be resized by the user.*
css .resizable { resize: both; overflow: auto; }

**contain**
*Defines how an element is contained within its context.*
css .container { contain: layout style; }

**contain-intrinsic-size**
*Defines the intrinsic size of a container with containment.*
css .container { contain-intrinsic-size: 200px 100px; }

**pointer-events**
*Controls whether or not an element can be the target of mouse events.*
css .no-pointer-events { pointer-events: none; }

**appearance**
*Defines the default styling of user interface controls.*
css input[type="button"] { appearance: none; }

**opacity**
*Defines the transparency level of an element.*
css .transparent { opacity: 0.5; }

**clip**
*Defines a clipping region for an element.*
css .clipped { clip: rect(0, 100px, 100px, 0); }

**clip-path**
*Defines a clipping path for an element.*
css .clipped { clip-path: circle(50%); }

**mask**
*Defines a mask for an element.*
css .masked { mask: url('mask.svg'); }

**mask-type**
*Defines whether a mask is applied as a luminance or alpha mask.*
css .masked { mask-type: alpha; }

**mask-image**
*Defines an image as a mask for an element.*
css .masked { mask-image: url('mask.png'); }

**mask-position**
*Defines the position of the mask image.*

css .masked { mask-position: center; }

**mask-size**
*Defines the size of the mask image.*
css .masked { mask-size: cover; }

**mask-repeat**
*Defines if and how the mask image should be repeated.*
css .masked { mask-repeat: no-repeat; }

**mask-origin**
*Defines the positioning area of the mask image.*
css .masked { mask-origin: border-box; }

**mask-composite**
*Defines how multiple masks are combined.*
css .masked { mask-composite: add; }

**mask-border**
*Defines an image mask for an element's border.*
css .bordered { mask-border: url('border-mask.svg'); }

**border-image**
*Shorthand for setting all the border-image properties.*
css .bordered { border-image: url('border.png') 30 round; }

**border-image-source**
*Defines the source of the border image.*
css .bordered { border-image-source: url('border.png'); }

**border-image-slice**
*Defines how to slice the border image.*
css .bordered { border-image-slice: 30%; }

**border-image-width**
*Defines the width of the border image.*
css .bordered { border-image-width: 5px; }

**border-image-outset**
*Defines how much the border image area extends beyond the border box.*
css .bordered { border-image-outset: 10px; }

**border-image-repeat**
*Defines how the border image should be repeated.*

css .bordered { border-image-repeat: stretch; }

**contain**
*Defines the containment for layout, style, and paint.*
css .container { contain: content; }

**scroll-snap-type**
*Defines how scrolling behaves within a scroll container.*
css .container { scroll-snap-type: y mandatory; }

**scroll-snap-align**
*Defines where the scroll container should snap to.*
css .snap-item { scroll-snap-align: start; }

**scroll-snap-stop**
*Defines whether a scroll container should snap to the nearest snap position.*
css .snap-item { scroll-snap-stop: always; }

**scroll-snap-points-x**
*Defines the snap positions along the x-axis (deprecated in favor of scroll-snap-type).*
css .container { scroll-snap-points-x: repeat(100px); }

**scroll-snap-points-y**
*Defines the snap positions along the y-axis (deprecated in favor of scroll-snap-type).*
css .container { scroll-snap-points-y: repeat(100px); }

**scroll-behavior**
*Defines the scroll behavior for the container.*
css .container { scroll-behavior: smooth; }

**tab-size**
*Defines the width of tab characters.*
css pre { tab-size: 4; }

**hyphens**
*Controls hyphenation of text.*
css p { hyphens: auto; }

**text-align-last**
*Defines the alignment of the last line of text.*
css p { text-align-last: right; }

**text-justify**

*Defines the justification of text.*
css p { text-justify: inter-word; }

**word-break**
*Defines how words should break when reaching the end of a line.*
css p { word-break: break-all; }

**overflow-anchor**
*Controls scroll anchoring behavior.*
css .container { overflow-anchor: none; }

**scrollbar-width**
*Defines the width of the scrollbar.*
css .container { scrollbar-width: thin; }

**scrollbar-color**
*Defines the color of the scrollbar.*
css .container { scrollbar-color: blue gray; }

**mask-image**
*Defines an image as a mask for an element.*
css .masked { mask-image: url('mask.png'); }

**mask-position**
*Defines the position of the mask image.*
css .masked { mask-position: center; }

**mask-size**
*Defines the size of the mask image.*
css .masked { mask-size: cover; }

**mask-repeat**
*Defines if and how the mask image should be repeated.*
css .masked { mask-repeat: no-repeat; }

**mask-origin**
*Defines the positioning area of the mask image.*
css .masked { mask-origin: border-box; }

**mask-composite**
*Defines how multiple masks are combined.*
css .masked { mask-composite: add; }

**mask-border**
*Defines an image mask for an element's border.*

css .bordered { mask-border: url('border-mask.svg'); }

**aspect-ratio**
*Defines the aspect ratio of an element.*
css .box { aspect-ratio: 16 / 9; }

**contain-intrinsic-size**
*Defines the intrinsic size of an element when it has containment.*
css .container { contain-intrinsic-size: 200px 100px; }

**filter**
*Applies graphical effects such as blur or color shifting to an element.*
css .filtered { filter: blur(5px); }

**backdrop-filter**
*Applies graphical effects to the area behind an element.*
css .overlay { backdrop-filter: blur(5px); }

**accent-color**
*Defines the color of UI controls and accents.*
css button { accent-color: #ff5722; }

**font-feature-settings**
*Controls the advanced typographic features of a font.*
css p { font-feature-settings: "liga" 1, "calt" 1; }

**font-kerning**
*Controls the use of kerning pairs in fonts.*
css p { font-kerning: normal; }

**font-variant**
*Controls the use of different font variants.*
css p { font-variant: small-caps; }

**font-variant-caps**
*Controls the use of capital letters.*
css p { font-variant-caps: all-small-caps; }

**font-variant-east-asian**
*Controls the typographic features for East Asian languages.*
css p { font-variant-east-asian: ruby; }

**font-variant-numeric**
*Controls the numeric typographic features.*
css p { font-variant-numeric: lining-nums; }

**font-variant-alternates**
*Controls the use of alternate glyphs.*
css p { font-variant-alternates: stylistic(); }

**font-variant-ligatures**
*Controls the use of ligatures in fonts.*
css p { font-variant-ligatures: common-ligatures; }

**grid-template-areas**
*Defines named grid areas in a grid layout.*
css .container { grid-template-areas: "header header" "sidebar content"; }

**grid-template-columns**
*Defines the columns in a grid layout.*
css .container { grid-template-columns: 1fr 2fr; }

**grid-template-rows**
*Defines the rows in a grid layout.*
css .container { grid-template-rows: auto 1fr; }

**grid-auto-rows**
*Defines the size of rows in an implicit grid.*
css .container { grid-auto-rows: minmax(100px, auto); }

**grid-auto-columns**
*Defines the size of columns in an implicit grid.*
css .container { grid-auto-columns: 100px; }

**grid-auto-flow**
*Controls the placement of grid items in the grid.*
css .container { grid-auto-flow: row dense; }

**grid-column**
*Shorthand for defining grid column start and end.*
css .item { grid-column: 1 / 3; }

**grid-row**
*Shorthand for defining grid row start and end.*
css .item { grid-row: 2 / 4; }

**grid-column-start**
*Defines the start line of a grid item.*
css .item { grid-column-start: 1; }

**grid-column-end**

*Defines the end line of a grid item.*
css .item { grid-column-end: 3; }

**grid-row-start**
*Defines the start line of a grid item.*
css .item { grid-row-start: 2; }

**grid-row-end**
*Defines the end line of a grid item.*
css .item { grid-row-end: 4; }

**grid-area**
*Defines a grid item's size and location in the grid.*
css .item { grid-area: header; }

**place-items**
*Shorthand for setting both align-items and justify-items.*
css .container { place-items: center; }

**place-content**
*Shorthand for setting both align-content and justify-content.*
css .container { place-content: center; }

**place-self**
*Shorthand for setting both align-self and justify-self.*
css .item { place-self: center; }

**repeat**
*Defines how many times a pattern should be repeated.*
css .background { background-repeat: repeat-x; }

**cross-fade**
*Creates a smooth transition between two images.*
css .fade { background-image: cross-fade(url('image1.jpg'), url('image2.jpg'), 0.5); }

**filter**
*Applies graphical effects such as blur or brightness to an element.*
css .blurred { filter: blur(4px); }

**will-change**
*Hints which properties will change, enabling optimizations.*
css .animated { will-change: transform; }

**text-shadow**
*Applies shadow effects to text.*

css h1 { text-shadow: 2px 2px 4px rgba(0,0,0,0.5); }

**text-overflow**
*Defines how overflowed content should be signaled.*
css .ellipsis { text-overflow: ellipsis; overflow: hidden; white-space: nowrap; }

**line-clamp**
*Clamps the content of a block to a maximum number of lines.*
css .clamped { display: -webkit-box; -webkit-line-clamp: 2; -webkit-box-orient: vertical; overflow: hidden; }

**text-decoration-color**
*Defines the color of text decorations.*
css .decorated { text-decoration-color: red; }

**text-decoration-style**
*Defines the style of text decorations.*
css .decorated { text-decoration-style: dashed; }

**text-decoration-thickness**
*Defines the thickness of text decorations.*
css .decorated { text-decoration-thickness: 2px; }

**text-transform**
*Controls the capitalization of text.*
css .uppercase { text-transform: uppercase; }

**text-align**
*Aligns text content.*
css .centered { text-align: center; }

**letter-spacing**
*Defines the spacing between letters.*
css .spaced { letter-spacing: 2px; }

**word-spacing**
*Defines the spacing between words.*
css .spaced { word-spacing: 4px; }

**white-space**
*Defines how white space inside an element is handled.*
css .nowrap { white-space: nowrap; }

**word-wrap**
*Defines whether to break words when reaching the end of a line.*

css .break-word { word-wrap: break-word; }

**resize**
*Defines if and how an element can be resized by the user.*
css .resizable { resize: both; overflow: auto; }

**transition**
*Defines the transition effects for properties.*
css .box { transition: all 0.3s ease; }

**animation**
*Shorthand for setting all animation properties.*
css .animated { animation: slide 2s ease-in-out; }

**animation-name**
*Defines the name of the keyframes animation.*
css .animated { animation-name: slide; }

**animation-duration**
*Defines the duration of the animation.*
css .animated { animation-duration: 2s; }

**animation-timing-function**
*Defines the speed curve of the animation.*
css .animated { animation-timing-function: ease-in-out; }

**animation-delay**
*Defines a delay before the animation starts.*
css .animated { animation-delay: 1s; }

**animation-iteration-count**
*Defines the number of times an animation should repeat.*
css .animated { animation-iteration-count: infinite; }

**animation-direction**
*Defines the direction of the animation.*
css .animated { animation-direction: alternate; }

**animation-fill-mode**
*Defines how styles are applied before and after the animation.*
css .animated { animation-fill-mode: forwards; }

**animation-play-state**
*Defines whether the animation is running or paused.*
css .animated { animation-play-state: paused; }

**transform**
*Applies 2D or 3D transformations to an element.*
css .transformed { transform: rotate(45deg); }

**transform-origin**
*Defines the origin point of transformations.*
css .transformed { transform-origin: center center; }

**transform-style**
*Defines how nested elements are rendered in 3D space.*
css .container { transform-style: preserve-3d; }

**perspective**
*Defines the perspective view for a 3D element.*
css .container { perspective: 1000px; }

**perspective-origin**
*Defines the origin point of the perspective view.*
css .container { perspective-origin: 50% 50%; }

**backface-visibility**
*Defines whether the back face of an element is visible when facing the viewer.*
css .flipped { backface-visibility: hidden; }

**clip-path**
*Defines a clipping path to create various shapes and effects.*
css .clipped { clip-path: circle(50% at 50% 50%); }

**mask-type**
*Defines the type of mask used for an element.*
css .masked { mask-type: alpha; }

**object-fit**
*Defines how the content of a replaced element should be resized to fit its container.*
css .image { object-fit: cover; }

**object-position**
*Defines the position of the content inside a replaced element.*
css .image { object-position: center; }

**contain**
*Defines containment for layout, style, and paint.*
css .contained { contain: layout style paint; }

**size**
*Defines the size of the page for printing.*
css @page { size: A4; }

**page-break-before**
*Defines if a page break should occur before an element when printing.*
css .break { page-break-before: always; }

**page-break-after**
*Defines if a page break should occur after an element when printing.*
css .break { page-break-after: always; }

**page-break-inside**
*Defines if a page break should occur inside an element when printing.*
css .no-break { page-break-inside: avoid; }

**writing-mode**
*Defines the direction in which text is laid out.*
css .vertical { writing-mode: vertical-rl; }

**direction**
*Defines the direction of text.*
css .rtl { direction: rtl; }

**text-align**
*Defines the horizontal alignment of text.*
css .centered { text-align: center; }

**text-indent**
*Defines the indentation of the first line of text.*
css .indented { text-indent: 20px; }

**text-transform**
*Defines the capitalization of text.*
css .uppercase { text-transform: uppercase; }

**text-align-last**
*Defines the alignment of the last line of text in a block.*
css p { text-align-last: right; }

**letter-spacing**
*Defines the spacing between letters.*
css .spaced { letter-spacing: 2px; }

**word-spacing**
*Defines the spacing between words.*

css .spaced { word-spacing: 4px; }

**line-height**
*Defines the line height of text.*
css p { line-height: 1.5; }

**text-decoration**
*Defines decorations added to text.*
css .decorated { text-decoration: underline; }

**text-shadow**
*Defines shadows added to text.*
css .shadowed { text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); }

**text-overflow**
*Defines how overflowed text should be signaled.*
css .truncated { text-overflow: ellipsis; }

**overflow-wrap**
*Defines how text should wrap when reaching the end of a line.*
css .wrap { overflow-wrap: break-word; }

**font-family**
*Defines the font family for text.*
css .custom-font { font-family: 'Arial', sans-serif; }

**font-size**
*Defines the size of the font.*
css .large { font-size: 20px; }

**font-weight**
*Defines the weight (thickness) of the font.*
css .bold { font-weight: bold; }

**font-style**
*Defines the style of the font.*
css .italic { font-style: italic; }

**font-variant**
*Defines the font variant.*
css .small-caps { font-variant: small-caps; }

**font-variant-caps**
*Defines capitalization of text in a font.*
css .small-caps { font-variant-caps: small-caps; }

**font-variant-numeric**
*Defines numeric style features of a font.*
css .lining-nums { font-variant-numeric: lining-nums; }

**font-variant-east-asian**
*Defines East Asian typographic features.*
css .ruby { font-variant-east-asian: ruby; }

**font-variant-alternates**
*Defines alternate font features.*
css .alternate { font-variant-alternates: stylistic(); }

**font-variant-ligatures**
*Defines ligature usage in fonts.*
css .ligatures { font-variant-ligatures: common-ligatures; }

**letter-spacing**
*Defines the spacing between letters.*
css .spaced { letter-spacing: 1px; }

**text-transform**
*Defines capitalization of text.*
css .uppercase { text-transform: uppercase; }

**line-height**
*Defines the height of a line box.*
css .spaced { line-height: 1.6; }

**white-space**
*Defines how whitespace is handled.*
css .nowrap { white-space: nowrap; }

**overflow-wrap**
*Defines how text should wrap.*
css .wrap { overflow-wrap: break-word; }

**visibility**
*Defines the visibility of an element.*
css .hidden { visibility: hidden; }

**opacity**
*Defines the opacity of an element.*
css .transparent { opacity: 0.5; }

**z-index**
*Defines the stack order of an element.*

css .overlay { z-index: 10; }

**display**
*Defines the display behavior of an element.*
css .block { display: block; }

**position**
*Defines the positioning method used for an element.*
css .absolute { position: absolute; }

**top**
*Defines the top offset of a positioned element.*
css .top { top: 10px; }

**right**
*Defines the right offset of a positioned element.*
css .right { right: 10px; }

**bottom**
*Defines the bottom offset of a positioned element.*
css .bottom { bottom: 10px; }

**left**
*Defines the left offset of a positioned element.*
css .left { left: 10px; }

**clip**
*Defines a clipping region for an absolutely positioned element.*
css .clipped { clip: rect(10px, 50px, 50px, 10px); }

**overflow**
*Defines how overflowed content should be handled.*
css .scroll { overflow: auto; }

**visibility**
*Defines the visibility of an element.*
css .hidden { visibility: hidden; }

**pointer-events**
*Defines how an element responds to pointer events.*
css .no-pointer { pointer-events: none; }

**contain**
*Defines the containment of an element.*
css .contained { contain: layout; }

**transform**

*Defines a transformation to be applied to an element.*

css .transformed { transform: rotate(45deg); }

**filter**

*Defines graphical effects such as blur or brightness.*

css .filtered { filter: brightness(0.5); }

**backdrop-filter**

*Defines graphical effects applied to the area behind an element.*

css .backdrop { backdrop-filter: blur(5px); }

**background-clip**

*Defines the painting area of the background.*

css .clip { background-clip: padding-box; }

**background-origin**

*Defines the origin of the background.*

css .origin { background-origin: border-box; }

**background-size**

*Defines the size of the background image.*

css .background { background-size: cover; }

**background-repeat**

*Defines how background images should be repeated.*

css .repeat { background-repeat: no-repeat; }

**background-position**

*Defines the position of a background image.*

css .position { background-position: center; }

**background-attachment**

*Defines how the background image is fixed or scrolls with the page.*

css .fixed { background-attachment: fixed; }

**background-color**

*Defines the background color of an element.*

css .color { background-color: #f00; }

**background-image**

*Defines the background image of an element.*

css .image { background-image: url('image.jpg'); }

**background**

*Shorthand for setting all background properties.*

css .background { background: #f00 url('image.jpg') no-repeat center fixed; }

**border-radius**
*Defines the radius of the element's corners.*
css .rounded { border-radius: 10px; }

**border-width**
*Defines the width of the border.*
css .bordered { border-width: 2px; }

**border-style**
*Defines the style of the border.*
css .bordered { border-style: solid; }

**border-color**
*Defines the color of the border.*
css .bordered { border-color: #f00; }

**border**
*Shorthand for setting all border properties.*
css .bordered { border: 2px solid #f00; }

**border-top**
*Shorthand for setting the top border properties.*
css .border-top { border-top: 2px solid #f00; }

**border-right**
*Shorthand for setting the right border properties.*
css .border-right { border-right: 2px solid #f00; }

**border-bottom**
*Shorthand for setting the bottom border properties.*
css .border-bottom { border-bottom: 2px solid #f00; }

**border-left**
*Shorthand for setting the left border properties.*
css .border-left { border-left: 2px solid #f00; }

**margin**
*Defines the space outside the element.*
css .margin { margin: 20px; }

**margin-top**
*Defines the top margin of an element.*

css .margin-top { margin-top: 20px; }

**margin-right**
*Defines the right margin of an element.*
css .margin-right { margin-right: 20px; }

**margin-bottom**
*Defines the bottom margin of an element.*
css .margin-bottom { margin-bottom: 20px; }

**margin-left**
*Defines the left margin of an element.*
css .margin-left { margin-left: 20px; }

**padding**
*Defines the space inside the element.*
css .padding { padding: 20px; }

**padding-top**
*Defines the top padding of an element.*
css .padding-top { padding-top: 20px; }

**padding-right**
*Defines the right padding of an element.*
css .padding-right { padding-right: 20px; }

**padding-bottom**
*Defines the bottom padding of an element.*
css .padding-bottom { padding-bottom: 20px; }

**padding-left**
*Defines the left padding of an element.*
css .padding-left { padding-left: 20px; }

**float**
*Defines the float position of an element.*
css .float-left { float: left; }

**clear**
*Defines what elements can float beside the cleared element.*
css .clear { clear: both; }

**list-style**
*Defines the style of list items.*
css ul { list-style: circle; }

**list-style-type**
*Defines the type of list item marker.*
css ul { list-style-type: square; }

**list-style-position**
*Defines the position of the list item marker.*
css ul { list-style-position: inside; }

**list-style-image**
*Defines an image as the list item marker.*
css ul { list-style-image: url('marker.png'); }

**visibility**
*Defines whether an element is visible or not.*
css .visible { visibility: visible; }

**outline**
*Shorthand for setting outline properties.*
css .outlined { outline: 2px solid red; }

**outline-width**
*Defines the width of the outline.*
css .outlined { outline-width: 2px; }

**outline-style**
*Defines the style of the outline.*
css .outlined { outline-style: solid; }

**outline-color**
*Defines the color of the outline.*
css .outlined { outline-color: red; }

**outline-offset**
*Defines the space between the outline and the border edge.*
css .outlined { outline-offset: 4px; }

**overflow-x**
*Defines how to handle overflow on the x-axis.*
css .overflow-x { overflow-x: scroll; }

**overflow-y**
*Defines how to handle overflow on the y-axis.*
css .overflow-y { overflow-y: hidden; }

**position**
*Defines the positioning method for an element.*

css .relative { position: relative; }

**top**
*Defines the top offset of a positioned element.*
css .top { top: 10px; }

**right**
*Defines the right offset of a positioned element.*
css .right { right: 10px; }

**bottom**
*Defines the bottom offset of a positioned element.*
css .bottom { bottom: 10px; }

**left**
*Defines the left offset of a positioned element.*
css .left { left: 10px; }

**transform**
*Defines a transformation to be applied to an element.*
css .scale { transform: scale(1.5); }

**transform-origin**
*Defines the origin point for the transformation.*
css .origin { transform-origin: center center; }

**transition**
*Defines the transition effects for properties.*
css .transition { transition: all 0.3s ease; }

**animation**
*Defines the animation to be applied to an element.*
css .animate { animation: bounce 2s infinite; }

**animation-name**
*Defines the name of the animation.*
css .animate { animation-name: bounce; }

**animation-duration**
*Defines the duration of the animation.*
css .animate { animation-duration: 2s; }

**animation-timing-function**
*Defines the timing function of the animation.*
css .animate { animation-timing-function: ease-in; }

**animation-delay**
*Defines the delay before the animation starts.*
css .animate { animation-delay: 1s; }

**animation-iteration-count**
*Defines the number of times the animation should repeat.*
css .animate { animation-iteration-count: infinite; }

**animation-direction**
*Defines the direction of the animation.*
css .animate { animation-direction: alternate; }

**animation-fill-mode**
*Defines how the animation styles are applied before and after the animation.*
css .animate { animation-fill-mode: forwards; }

**animation-play-state**
*Defines whether the animation is running or paused.*
css .animate { animation-play-state: paused; }

**@keyframes**
*Defines the keyframes for an animation.*
css @keyframes bounce { 0% { transform: translateY(0); } 50% { transform: translateY(-30px); } 100% { transform: translateY(0); } }

**@media**
*Defines responsive design rules.*
css @media (max-width: 600px) { .responsive { font-size: 14px; } }

**@supports**
*Defines conditional CSS rules based on feature support.*
css @supports (display: grid) { .grid { display: grid; } }

**@import**
*Defines a rule for importing external style sheets.*
css @import url('styles.css');

**@font-face**
*Defines custom fonts.*
css @font-face { font-family: 'MyFont'; src: url('myfont.woff2') format('woff2'); }

**@page**
*Defines styles for printed pages.*

css @page { margin: 1in; }

**@viewport**
*Defines the viewport properties for responsive design.*
css @viewport { width: device-width; zoom: 1; }

**@keyframes**
*Defines the keyframes for animations.*
css @keyframes slideIn { from { transform: translateX(-100%); } to { transform: translateX(0); } }

**@counter-style**
*Defines a custom counter style.*
css @counter-style custom { system: cyclic; symbols: '◉' '◊' '◯'; }

**@document**
*Defines styles for specific documents.*
css @document url('https://example.com') { body { color: blue; } }

**@color**
*Defines a custom color scheme.*
css @color scheme { color: #ff0; }

**@scroll-timeline**
*Defines a scroll-based timeline for animations.*
css @scroll-timeline { scroll-source: scroll; time-range: 1s; }

**@container**
*Defines container queries for responsive design.*
css @container (min-width: 500px) { .container { width: 100%; } }

**@scrollbar**
*Defines custom scrollbar styles.*
css @scrollbar { width: 16px; background: #f0f0f0; }

**@media**
*Defines responsive media queries.*
css @media (min-width: 768px) { .container { width: 750px; } }

**@font-face**
*Defines a custom font face.*
css @font-face { font-family: 'MyCustomFont'; src: url('mycustomfont.woff2') format('woff2'); }

**@supports**

*Defines conditional styles based on feature support.*

```
css @supports (display: grid) { .grid-container { display: grid; grid-template-columns: 1fr 1fr; } }
```