



# CSS

# Cascading Style Sheets

---

**Class # 2**

# CSS Properties(Text Color)

The color property in CSS is used to specify the color of the text content within an element. It determines the foreground color or text color.

```
selector {  
  color: value;  
}
```

## Syntax

### Usage:

- **color: colorValue;** - Specifies the color using various formats such as color names (e.g., "red", "blue"), hexadecimal codes (e.g., "#FF0000"), RGB values (e.g., "rgb(255, 0, 0)"), or HSL values (e.g., "hsl(0, 100%, 50%)").

# CSS Properties(Text Color)

## Examples:

```
body {  
  color: red; /* Sets the text color to red */  
}
```

```
h1 {  
  color: #00FF00; /* Sets the text color to green using a hexadecimal code */  
}
```

```
p {  
  color: rgb(0, 0, 255); /* Sets the text color to blue using RGB values */  
}
```

# CSS Properties(Text Align)

The text-align property in CSS is used to specify the horizontal alignment of text within an element. It determines how the text is positioned within its container.

```
selector {  
  text-align: value;  
}
```

## Syntax

### Usage:

- ❑ **text-align: left;** - Aligns the text to the left edge of the container.
- ❑ **text-align: right;** - Aligns the text to the right edge of the container.
- ❑ **text-align: center;** - Centers the text horizontally within the container.
- ❑ **text-align: justify;** - Adjusts the spacing between words to justify the text within the container

# CSS Properties(Text Align)

## Examples:

```
body {  
  text-align: center; /* Centers the text horizontally within the body element */  
}
```

```
h1 {  
  text-align: left; /* Aligns the text to the left edge within <h1> elements */  
}
```

```
p {  
  text-align: right; /* Aligns the text to the right edge within <p> elements */  
}
```

# CSS Properties(Text Decoration)

The text-decoration property in CSS is used to add visual decorations to text within an element. It allows you to underline, strike through, or apply other text effects.

```
selector {  
  text-decoration: value;  
}
```

## Syntax

### Usage:

- ❑ **text-decoration: none;** - Removes any text decoration (default).
- ❑ **text-decoration: underline;** - Adds an underline to the text.
- ❑ **text-decoration: overline;** - Adds a line above the text.
- ❑ **text-decoration: line-through;** - Adds a line through the text, creating a strike-through effect.
- ❑ **text-decoration: blink;** - Makes the text blink or flash (not widely supported).

# CSS Properties(Text Decoration)

## Examples:

```
a {  
  text-decoration: underline; /* Adds an underline to all anchor links */  
}
```

```
h1 {  
  text-decoration: line-through; /* Adds a strike-through effect to all <h1> headings */  
}
```

```
p {  
  text-decoration: overline; /* Adds a line above all paragraphs */  
}
```

# CSS Properties(Text Transform)

The text-transform property in CSS is used to specify the capitalization or transformation of text within an element. It allows you to change the case of the text to uppercase, lowercase, or capitalize the first letter of each word.

```
selector {  
  text-transform: value;  
}
```

## Syntax

### Usage:

- ❑ **text-transform: none;** - No capitalization or transformation is applied (default).
- ❑ **text-transform: uppercase;** - Transforms the text to uppercase.
- ❑ **text-transform: lowercase;** - Transforms the text to lowercase.
- ❑ **text-transform: capitalize;** - Capitalizes the first letter of each word.



# CSS Properties(Text Transform)

## Examples:

```
h1 {  
  text-transform: uppercase; /* Transforms the text of all <h1> headings to uppercase */  
}
```

```
p {  
  text-transform: capitalize; /* Capitalizes the first letter of each word in all paragraphs */  
}
```

```
span {  
  text-transform: lowercase; /* Transforms the text of all <span> elements to lowercase */  
}
```

# CSS Properties(Text Indentation)

The text-indent property in CSS is used to specify the indentation of the first line of text within an element. It allows you to create a space or offset at the beginning of a paragraph or block of text.

```
selector {  
  text-indent: value;  
}
```

## Syntax

### Usage:

- **text-indent: lengthValue;** - Sets the indentation value using a specific length, such as pixels (px), em, rem, or percentages (%). A positive value creates an indentation, while a negative value creates a hanging indent.

# CSS Properties(Text Indentation)

## Examples:

```
p {  
  text-indent: 20px; /* Sets a 20-pixel indentation for all paragraphs */  
}
```

```
blockquote {  
  text-indent: -10px; /* Creates a hanging indent of 10 pixels for all blockquote elements */  
}
```

# CSS Properties(Letter Spacing)

The letter-spacing property in CSS is used to adjust the spacing between characters (letters) within a block of text. It allows you to increase or decrease the space between individual letters.

```
✓ selector {  
  letter-spacing: value;  
}
```

## Syntax

### Usage:

- ❑ **letter-spacing: normal;** - Default spacing between characters (no additional spacing).
- ❑ **letter-spacing: lengthValue;** - Sets the spacing value using a specific length, such as pixels (px), em, rem, or percentages (%). A positive value increases the space between letters, while a negative value decreases it.

# CSS Properties(Letter Spacing)

## Examples:

```
h1 {  
  letter-spacing: 2px; /* Increases the spacing between letters in all <h1> headings by 2 pixels */  
}
```

```
p {  
  letter-spacing: -1px; /* Decreases the spacing between letters in all paragraphs by 1 pixel */  
}
```

```
span {  
  letter-spacing: 0.5em; /* Increases the spacing between letters in all <span> elements by 0.5  
em */  
}
```

# CSS Properties(Word Spacing)

The word-spacing property in CSS is used to adjust the spacing between words within a block of text. It allows you to increase or decrease the space between individual words.

```
selector {  
  word-spacing: value;  
}
```

## Syntax

### Usage:

- ❑ **word-spacing: normal;** - Default spacing between words (no additional spacing).
- ❑ **word-spacing: lengthValue;** - Sets the spacing value using a specific length, such as pixels (px), em, rem, or percentages (%). A positive value increases the space between words, while a negative value decreases it.

# CSS Properties(Word Spacing)

## Examples:

```
p {  
  word-spacing: 2px; /* Increases the spacing between words in all paragraphs by 2 pixels */  
}
```

```
h1 {  
  word-spacing: -1px; /* Decreases the spacing between words in all <h1> headings by 1 pixel */  
}
```

```
span {  
  word-spacing: 0.5em; /* Increases the spacing between words in all <span> elements by 0.5 em  
  */  
}
```

# CSS Properties(Line Height)

The line-height property in CSS is used to control the height of each line of text within an element. It specifies the vertical space between lines, including the height of the text itself and any additional spacing.

```
selector {  
  line-height: value;  
}
```

## Syntax

### Usage:

- ❑ **line-height: normal;** - Uses the default line height (typically 1.2 times the font size).
- ❑ **line-height: numberValue;** - Sets the line height as a number value. For example, 1.5 means the line height is 1.5 times the font size.
- ❑ **line-height: lengthValue;** - Sets the line height using a specific length, such as pixels (px), em, rem, or percentages (%).



# CSS Properties(Line Height)

## Examples:

```
p {  
  line-height: 1.5; /* Sets the line height of all paragraphs to 1.5 times the font size */  
}  
  
h1 {  
  line-height: 30px; /* Sets the line height of all <h1> headings to 30 pixels */  
}  
  
div {  
  line-height: 120%; /* Sets the line height of all <div> elements to 120% of the font size */  
}
```

# CSS Properties(Text Shadow)

The text-shadow property in CSS is used to apply a shadow effect to text. It allows you to add a shadow behind the text, providing depth and visual emphasis.

```
selector {  
  text-shadow: h-shadow v-shadow blur-radius color;  
}
```

## Syntax

### Usage:

- ❑ **h-shadow** specifies the horizontal offset of the shadow (positive values move it to the right, negative values move it to the left).
- ❑ **v-shadow** specifies the vertical offset of the shadow (positive values move it down, negative values move it up).
- ❑ **blur-radius** specifies the blurring effect of the shadow. Higher values create a more blurred shadow.
- ❑ **color** specifies the color of the shadow.

# CSS Properties(Text Shadow)

## Examples:

```
h1 {  
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Adds a black shadow with a 2-pixel horizontal  
and vertical offset, 4-pixel blur radius, and 50% transparency to all <h1> headings */  
}
```

```
p {  
  text-shadow: -1px -1px 2px #FF0000; /* Adds a red shadow with a 1-pixel horizontal and  
vertical offset, 2-pixel blur radius to all paragraphs */  
}
```

```
span {  
  text-shadow: 0 0 3px rgba(0, 0, 255, 0.8); /* Adds a blue shadow with a 3-pixel blur radius and  
80% transparency to all <span> elements */  
}
```

# CSS Properties(Margin)

The margin property in CSS is used to control the spacing around an element. It defines the amount of space between the element's outer edge and adjacent elements.

```
selector {  
  margin: top right bottom left;  
}
```

## Syntax

### Usage:

- ❑ **margin: auto;** - Centers the element horizontally within its container.
- ❑ **margin: lengthValue;** - Sets the same margin value for all sides of the element.
- ❑ **margin: verticalValue horizontalValue;** - Sets the margin value for the vertical and horizontal sides of the element separately.
- ❑ **margin: topValue rightValue bottomValue leftValue;** - Sets the margin value individually for each side of the element.

# CSS Properties(Margin)

## Examples:

```
p {  
  margin: 20px; /* Sets a 20-pixel margin on all sides of paragraphs */  
}
```

```
div {  
  margin: 10px 20px; /* Sets a 10-pixel margin on the top and bottom, and a 20-pixel margin on  
the left and right of all <div> elements */  
}
```

```
h1 {  
  margin: 0 auto; /* Centers the <h1> element horizontally within its container */  
}
```

# CSS Properties(Padding)

The padding property in CSS is used to control the spacing between the content of an element and its border. It defines the amount of space between the element's content and its inner edges.

```
selector {  
  padding: top right bottom left;  
}
```

## Syntax

### Usage:

- ❑ **padding: lengthValue;** - Sets the same padding value for all sides of the element.
- ❑ **padding: verticalValue horizontalValue;** - Sets the padding value for the vertical and horizontal sides of the element separately.
- ❑ **padding: topValue rightValue bottomValue leftValue;** - Sets the padding value individually for each side of the element.

# CSS Properties(Padding)

## Examples:

```
p {  
  padding: 10px; /* Sets a 10-pixel padding on all sides of paragraphs */  
}
```

```
div {  
  padding: 20px 10px; /* Sets a 20-pixel padding on the top and bottom, and a 10-pixel padding  
on the left and right of all <div> elements */  
}
```

```
h1 {  
  padding: 0; /* Removes padding from all sides of <h1> elements */  
}
```

# CSS ShortHand Properties

Shorthand properties in CSS allow you to set multiple related properties using a single line of code.

- **margin:** Sets the margin values for all four sides of an element in a clockwise order: top, right, bottom, and left.
  - **Example:** `margin: 10px 20px 10px 20px;` (top: 10px, right: 20px, bottom: 10px, left: 20px)
- **padding:** Sets the padding values for all four sides of an element in a clockwise order: top, right, bottom, and left.
  - **Example:** `padding: 10px 20px 10px 20px;` (top: 10px, right: 20px, bottom: 10px, left: 20px)
- **border:** Sets the properties for all sides of an element's border, including the width, style, and color.
  - **Example:** `border: 1px solid #000;` (width: 1px, style: solid, color: #000)



# CSS ShortHand Properties

Shorthand properties in CSS allow you to set multiple related properties using a single line of code.

- **background:** Sets multiple background properties, including background color, image, position, repeat, and attachment.
  - **Example:** background: #f00 url('image.jpg') center top no-repeat fixed;  
(color: #f00, image: 'image.jpg', position: center top, repeat: no-repeat, attachment: fixed)
- **font:** Sets multiple font properties, including font size, font family, font weight, font style, and line height.
  - **Example:** font: 16px/1.5 Arial, sans-serif; (size: 16px, line height: 1.5, family: Arial, sans-serif)

# Types of Units in CSS

Two types of units in CSS:

- Absolute Units
- Relative Units

# Absolute Units

Absolute units in CSS are used to specify measurements that are fixed and do not change with the context or size of the viewport. Here are some commonly used absolute units in CSS:

- ❑ **px (pixels):** A pixel is the smallest unit of measurement in CSS and represents a single dot on a screen. It is commonly used to set precise dimensions, such as width, height, padding, and margins.
- ❑ **pt (points):** A point is a unit of measurement commonly used in print media. In CSS, 1 point is equal to 1/72 of an inch. It is often used to set font sizes for print stylesheets

# Absolute Units

- **in (inches):** An inch is a unit of measurement commonly used in print media and physical documents. In CSS, it represents a physical inch on the screen. It is less commonly used in web design due to the variability of screen sizes.
- **cm (centimeters):** A centimeter is a unit of measurement commonly used in print media and physical documents. In CSS, it represents a physical centimeter on the screen. Like inches, it is less commonly used in web design.
- **mm (millimeters):** A millimeter is a unit of measurement commonly used in print media and physical documents. In CSS, it represents a physical millimeter on the screen. It is less commonly used in web design.

# Relative Units

Relative units in CSS are used to specify measurements that are relative to other elements or properties, making them more adaptable and responsive to different screen sizes and contexts. Here are some commonly used relative units in CSS:

- ❑ **% (percentage):** The percentage unit represents a proportion relative to the parent element or the containing block. It is often used to set widths, heights, margins, and paddings. For example, `width: 50%;` sets the width of an element to 50% of its parent's width.
- ❑ **em:** The em unit is relative to the font size of the element itself. One em is equal to the computed font size of the element. It is commonly used to set font sizes and spacing. For example, `font-size: 1.2em;` sets the font size to 1.2 times the parent element's font size.

# Relative Units

- ❑ **rem:** Similar to em, the rem unit is relative to the root element's font size (usually the <html> element). It ensures consistent scaling across the entire document, irrespective of nested elements. It is often used for font sizes and spacing.
- ❑ **vw (viewport width):** The vw unit represents a percentage of the viewport's width. It allows for responsive sizing based on the width of the viewport. For example, width: 50vw; sets the width of an element to 50% of the viewport's width.
- ❑ **vh (viewport height):** Similar to vw, the vh unit represents a percentage of the viewport's height. It enables responsive sizing based on the height of the viewport. For example, height: 75vh; sets the height of an element to 75% of the viewport's height.