

CSS Properties

Test Color:

- By using "color" property name we can provide color values to text content.
- Colors values are specified using predefined color names, or RGB, or RGBA, etc...

For example: `p { color : red; }`

- Here It displays all paragraphs with red color content.

CSS Background Color

- If we want to set the background color for HTML elements then we need to use background-color property name.

For example: `p { background-color : yellow; }`

CSS RGB Colors:

- An RGB color value represents RED, GREEN, and BLUE light sources.
- In CSS, a color can be specified as an RGB value, using this formula:
`rgb(red, green, blue)`
- Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.
- For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255) and the others are set to 0.
- To display black, set all color parameters to 0, like this: `rgb(0, 0, 0)`.
- To display white, set all color parameters to 255, like this: `rgb(255, 255, 255)`.

RGBA Value:

- RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.
- An RGBA color value is specified with: **`rgba(red, green, blue, alpha)`**
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):
`rgba(0, 0, 255, 0.5)` ----> it display blue color with 50% transparent

CSS Font Size :

- The font-size property sets the size of the text.

For example : `h1 { font-size : 30px }`

CSS Margins

- Margins are used to create space around elements, outside of any defined borders.
- With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).
- CSS has properties for specifying the margin for each side of an element:

margin-top

margin-right

margin-bottom

margin-left

- If the margin property has four values then use the margin shorthand property with four values:

For Example: `p { margin: 25px 50px 75px 100px ; }`

Here,

top margin is 25px

right margin is 50px

bottom margin is 75px

left margin is 100px

- If the margin property has three values then use the margin shorthand property with three values:

For example: `p { margin: 25px 50px 75px; }`

Here,

top margin is 25px

right and left margins are 50px

bottom margin is 75px

- If the margin property has two values then use the margin shorthand property with two values:

For example: `p { margin: 25px 50px; }`

Here,

top and bottom margins are 25px

right and left margins are 50px

- If the margin property has one value then use the margin shorthand property with one value:

For example: `p { margin: 25px; }`

Here,

All four side margins are 25px

CSS Padding:

- Padding is used to create space around an element's content, inside of any defined borders.
- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).
- CSS has properties for specifying the padding for each side of an element:

padding-top

padding-right

padding-bottom

padding-left

- The **padding** property is a shorthand property for the following individual padding properties like padding-top, padding-right , padding-bottom and padding-left.

Note: It is also same like margin property, sharing values to each side of an element.

CSS Borders:

- The CSS **border** properties allow you to specify the **border-width**, **border-style**, and **border-color** to an element.

- By using border property we can provide all the sides same width, style and color at a time.

For example: `h1 { border : 5px solid blue; }`

CSS border-style:

- The "**border-style**" property specifies what kind of border to display all the sides or each side of an element.

For example: `h1 { border-style : solid dotted double dashed; }`

Here,

top border-style is solid

right border-style is dotted

bottom border-style is double

left border-style is dashed

For example: `h1 { border-style : solid dotted double; }`

Here,

top border-style is solid

right border-style is dotted

bottom border-style is double

left border-style is dotted

For example: `h1 { border-style : solid dotted ; }`

Here,

top border-style is solid

right border-style is dotted

bottom border-style is solid

left border-style is dotted

For example: `h1 { border-style : solid ; }`

Here, top , right, bottom and left border-style is solid

- The following values are allowed to a border-style property:

dotted - Defines a dotted border

dashed - Defines a dashed border

solid - Defines a solid border

double - Defines a double border

none - Defines no border

CSS border-width:

- The "border-width" property specifies the width of the four borders.

For example: `h1{ border-width: 5px ; }`

CSS border-color:

- The "**border-color**" property is used to set the color of the four border.

- The color can be set by:

name - specify a color name, like "red"

HEX - specify a HEX value, like "#ff0000"

RGB - specify a RGB value, like "rgb(255,0,0)"

- **Note: If border-color is not set, it inherits the color of the element.**

For example: `h1 { border-style: solid; border-color: red; }`

- If you want set each side separate value then we can do like this,

For example:

```
h1 {  
  border-top-style: solid;  
  border-bottom-style: solid;  
  border-left-style: solid;  
  border-right-style: solid;  
  
  border-left-color: red;  
  border-bottom-width : 5px;  
}
```

CSS Rounded Borders

- The **border-radius** property is used to add rounded borders to an element:

For example: `h1 { border: 2px solid red; border-radius: 5px; }`

CSS Height and Width

- The CSS height and width properties are used to set the height and width of an element.
- The CSS max-width property is used to set the maximum width of an element.

For example:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      div {  
        height: 150px;  
        width: 100%;  
        border: 1px solid #4CAF50;  
        background-color: powderblue;  
      }  
    </style>  
  </head>  
  <body>  
    <h2>CSS height and width properties</h2>  
    <div> This div element has a height of 50 pixels and a width of 100%.  
  </div> <  
  
</body>  
</html>
```

CSS font-family Property

- In CSS, we use the font-family property to specify the font of a text.
- It means content style changing.

For example: `h1 { font-family : Arial, Helvetica, sans-serif; }`

CSS font-size Property

- The **font-size** property sets the size of the text.

For example : `h1 { font-size: 40px; }`

CSS outline Property

- An outline is a line drawn outside the element's border.
- By using **outline** property we can provide all the sides same width, style and color at a time.

For example: `h1 { outline : 5px solid blue; }`

- We can also use **outline-width** , **outline-style** and **outline-color** properties

CSS text-align Property:

- The **text-align** property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centered, or justified.
- The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

For example:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 { text-align: center; }
      h2 { text-align: left; }
      h3 { text-align: right; }
    </style>
  </head>
  <body>
    <h1>Heading 1 (center)</h1>
    <h2>Heading 2 (left)</h2>
    <h3>Heading 3 (right)</h3>
    <p>The three headings above are aligned center, left and right.</p>
  </body>
</html>
```

- When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):

CSS text-align-last Property

- The text-align-last property specifies how to align the last line of a text.

For example:

- Align the last line of text in three <p> elements:

```
p.a { text-align-last: right; }
p.b { text-align-last: center; }
p.c { text-align-last: justify; }
```

CSS Text Direction

- The direction and unicode-bidi properties can be used to change the text direction of an element:
For example: `h1 { direction: rtl; unicode-bidi: bidi-override; }`

CSS Text Decoration

- By using text-decoration property we can provide some decoration style to our text.
For example : `h1{ text-decoration : underline solid blue; }`
- We can also use text-decoration properties separately by using below properties.
text-decoration-line
text-decoration-color
text-decoration-style
text-decoration-thickness

CSS display Property

- The display property is the most important CSS property for controlling layout.
- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is **block** or **inline**.
- We can override the default display Value aslo.
- As mentioned, every element has a default display value. However, you can override this.
- Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

For example:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      li {display: inline; }
    </style>
  </head>
  <body>
    <p>Display a list of links as a horizontal menu:</p>
    <ul>
      <li><a href="/html/default.asp" target="_blank">HTML</a></li>
      <li><a href="/css/default.asp" target="_blank">CSS</a></li>
      <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
    </ul>
  </body>
</html>
```

CSS overflow Property:

- The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.
- The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.
- The overflow property has the following values:
- visible - Default. The overflow is not clipped. The content renders outside the element's box
- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
- auto - Similar to scroll, but it adds scrollbars only when necessary
- Note: The overflow property only works for block elements with a specified height.

For example:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        background-color: coral;
        width: 200px;
        height: 65px;
        border: 1px solid;
        overflow: visible;
      }
    </style>
  </head>
  <body>
    <h2>Overflow: visible</h2>

    <p>By default, the overflow is visible, meaning that it is not clipped
    and it renders outside the element's box:</p>

    <div>You can use the overflow property when you want to have better
    control of the layout. The overflow property specifies what happens if
    content overflows an element's box.</div>
  </body>
</html>
```