



MARK AS COMPLETE

VIEW CALENDAR 

CSS Properties

CSS Properties

There are a number of CSS properties that you can use to style our webpage. Here we'll discuss some of the CSS properties such as Border, Padding, Margin, display, position, color, and text-align.

CSS Border Property

The CSS border property allows to style the border area of a box. The properties and corresponding values with examples covered under the CSS border are tabulated below:

Property	values	Usage	Example
border-width	medium, thin, thick, length	Used to define the border area of a box	<code>div { border-width: medium 10px thick 15px; }</code>
border-style	none, hidden, dashed, dotted, double, groove, inset, outset, ridge and solid	sets the style of a box's border	<code>p { border-style: dotted; }</code>
border-color	hex-value for colors	specify the color of a box's border	<code>p { border-style: solid; border-color: #ff0000; }</code>

CSS Padding Property

The CSS padding property allow you to set the padding area for an element that separates its border from its content. The padding property can take one, two, three, or four values separated by white spaces as listed in the below table. Depending on the list of property values, the HTML element has the padding area on the top, bottom, right, and left.

Examples	Explanation
<code>p { padding: 70px; }</code>	Sets the padding for an <code><p></code> element to 70 pixels for all four sides
<code>p { padding: 35px 70px; }</code>	Sets the padding for an <code><p></code> element to 35 pixels for top and bottom and 70 pixels for right and left sides.
<code>p { padding: 35px 70px 40px; }</code>	Sets the padding for an <code><p></code> element to 35 pixels for the top, 70 pixels for the left and right side and 40 pixels for the bottom.
<code>p { padding: 35px 70px 40px 80px; }</code>	Sets the padding for an <code><p></code> element to 35 pixels for the top, 70 pixels for the right side, 40 pixels for the bottom and 80 pixels for the left side.

The padding property is a shorthand property for the padding-top, padding-right, padding-bottom, and padding-left properties. The below examples set padding on a specific side for the HTML element.

Examples:

```
h1 {
  padding-bottom: 10px;
}
p {
  padding-top: 20px;
  padding-left: 50px;
}
```

CSS Margin Property

The CSS margin property is similar to the CSS border property, but it sets the margins around the sides of an element's box instead of the border. It also takes one, two, three, or four values separated by white spaces. The shorthand properties are margin-top, margin-right, margin-bottom, and margin-left to set a margin on respective sides.

Example:

```
p {
  margin-left: 10px;
  margin-right: 30px;
}
h1 {
  margin: 25px 50px;
}
```

CSS Display Property

The display property controls the display behaviour of an element. The CSS display property sets whether an element is treated as a block or inline elements and the layout used for its children, such as flow layout, flex or grid.

There are two types of HTML elements: **inline-level elements** and **block-level elements**. The differences between these elements affect how you use the box model. Both inline and block-level elements appear within the body of an HTML page. But, inline-level elements are used to create a short structure that can have data and other inline elements. Inline level elements include ``, `<big>`, `<i>`, `<small>`, `<tt>`, `<abbr>`, `<acronym>`, `<code>`, ``, etc.

Block-level elements used to create larger structures than inline elements also it starts on new lines by default whereas inline-level elements not. Block elements include `<p>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, ``, ``, `<pre>`, `<address>`, `<blockquote>`, `<dl>`, `<div>`, `<fieldset>`, `<form>`, `<hr>`, `<table>`, etc .

The Syntax for the display property is `selector {display: value}` . The property values and description with examples are tabulated below:

Property value	Description	Example
block	behaves likes block-level elements	<code>a {display: block;}</code>
inline	behaves like inline-level elements	<code>ul li { display: inline; }</code>
none	elements doesn't generate boxes	<code>h1 { display: none;}</code>

CSS Position Property

The position property defines how an element will be arranged on a page. The Syntax for the position property is `selector {position: value}` . The property values are static, relative, absolute, fixed, or inherit.

static - The element's box is arranged automatically consistent with the normal flow.

relative - The element's box position is relative to its normal flow position. You can adjust the normal flow position by using the top, bottom, left and right properties.

absolute - The element's box arranged to an absolute position with reference to its containing block. Its containing block is that the nearest ancestor element that has its position property set to relative, absolute, or fixed. The top, right, bottom, and left properties are used to set the offset of the element's box with reference to its containing block.

fixed - The element's box position is offset from its browser window by using the top, right, bottom, and left properties. The element's box won't move when the browser window is scrolled.

inherit - The inherit keyword is employed to specify that the value for this property should be taken from the parent element. If inherit is used with the root element, then the initial value for this property is going to be used.

Example:

```
a {position: static;}
div {position: relative; top: 20px; left: 50px;}
h1 {position: absolute; top: 30px; left: 20px;}
div {position: fixed; top: 325px; left: 60px;}
```

CSS Color property

The color property is used to specify the foreground color of text. The color properties are set using 5 different color notation types which is listed below:

```
a {color: red;}
div {color: #3c5;}
h1 {color: #ffa500;}
div {color: rgb(100,20,255);}
#id1 {color: rgb(30%,50%,70%);}
```

CSS text-align property

The text-align property is used to align the content inside the element. The text inside the element can be aligned in 4 ways - left, right, center and justify.

Example: The text-align properties are set to left, right, justify, and center.

```
div {text-align:left;}
h1 {text-align: right;}
p {text-align: justify;}
div {text-align: center;}
```

◀ Previous
Inline, Internal, ...

Next ▶
Element Selectors