**Cascading Style Sheets (CSS)**

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language. CSS3 is collaboration of CSS2 specifications and new specifications, we can called this collaboration is **module**. Some of the modules are shown below −

* Selectors
* Box Model
* Backgrounds
* Image Values and Replaced Content
* Text Effects
* 2D Transformations
* 3D Transformations
* Animations
* Multiple Column Layout
* User Interface
* We have listed out all the CSS Measurement Units along with proper Examples −

|  |  |  |
| --- | --- | --- |
| **Unit** | **Description** | **Example** |
| % | Defines a measurement as a percentage relative to another value, typically an enclosing element. | p {font-size: 16pt; line-height: 125%;} |
| cm | Defines a measurement in centimeters. | div {margin-bottom: 2cm;} |
| em | A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt. | p {letter-spacing: 7em;} |
| ex | This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x. | p {font-size: 24pt; line-height: 3ex;} |
| in | Defines a measurement in inches. | p {word-spacing: .15in;} |
| mm | Defines a measurement in millimeters. | p {word-spacing: 15mm;} |
| pc | Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch. | p {font-size: 20pc;} |
| pt | Defines a measurement in points. A point is defined as 1/72nd of an inch. | body {font-size: 18pt;} |
| px | Defines a measurement in screen pixels. | p {padding: 25px;} |

You can specify your color values in various formats. Following table lists all the possible formats −

|  |  |  |
| --- | --- | --- |
| **Format** | **Syntax** | **Example** |
| Hex Code | #RRGGBB | p{color:#FF0000;} |
| Short Hex Code | #RGB | p{color:#6A7;} |
| RGB % | rgb(rrr%,ggg%,bbb%) | p{color:rgb(50%,50%,50%);} |
| RGB Absolute | rgb(rrr,ggg,bbb) | p{color:rgb(0,0,255);} |
| keyword | aqua, black, etc. | p{color:teal;} |

**Font Properties**

This chapter teaches you how to set fonts of a content, available in an HTML element. You can set following font properties of an element −

* The **font-family** property is used to change the face of a font.
* The **font-style** property is used to make a font italic or oblique.
* The **font-variant** property is used to create a small-caps effect.
* The **font-weight** property is used to increase or decrease how bold or light a font appears.
* The **font-size** property is used to increase or decrease the size of a font.
* The **font** property is used as shorthand to specify a number of other font properties.

**CSS Text**

This chapter teaches you how to manipulate text using CSS properties. You can set following text properties of an element −

* The **color** property is used to set the color of a text.
* The **direction** property is used to set the text direction.
* The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
* The **word-spacing** property is used to add or subtract space between the words of a sentence.
* The **text-indent** property is used to indent the text of a paragraph.
* The **text-align** property is used to align the text of a document.
* The **text-decoration** property is used to underline, overline, and strikethrough text.
* The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
* The **white-space** property is used to control the flow and formatting of text.
* The **text-shadow** property is used to set the text shadow around a text.

This tutorial will teach you how to set different properties of an HTML table using CSS. You can set following properties of a table −

* The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
* The **border-spacing** specifies the width that should appear between table cells.
* The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
* The **empty-cells** specifies whether the border should be shown if a cell is empty.
* The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

Now, we will see how to use these properties with examples.

The border-collapse Property

This property can have two values *collapse* and *separate*. The following example uses both the values −

<html>

<head>

<style type = "text/css">

table.one {border-collapse:collapse;}

table.two {border-collapse:separate;}

td.a {

border-style:dotted;

border-width:3px;

border-color:#000000;

padding: 10px;

}

td.b {

border-style:solid;

border-width:3px;

border-color:#333333;

padding:10px;

}

</style>

</head>

<body>

<table class = "one">

<caption>Collapse Border Example</caption>

<tr><td class = "a"> Cell A Collapse Example</td></tr>

<tr><td class = "b"> Cell B Collapse Example</td></tr>

</table>

<br />

<table class = "two">

<caption>Separate Border Example</caption>

<tr><td class = "a"> Cell A Separate Example</td></tr>

<tr><td class = "b"> Cell B Separate Example</td></tr>

</table>

</body>

</html>

**CSS Borders:**

The *border* properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change −

* The **border-color** specifies the color of a border.
* The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
* The **border-width** specifies the width of a border.

Now, we will see how to use these properties with examples.

## The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties −

* **border-bottom-color** changes the color of bottom border.
* **border-top-color** changes the color of top border.
* **border-left-color** changes the color of left border.
* **border-right-color** changes the color of right border.

**CSS Lists**

Lists are very helpful in conveying a set of either numbered or bullet points. This chapter teaches you how to control list type, position, style, etc., using CSS.

We have the following five CSS properties, which can be used to control lists −

* The **list-style-type** allows you to control the shape or appearance of the marker.
* The **list-style-position** specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
* The **list-style-image** specifies an image for the marker rather than a bullet point or number.
* The **list-style** serves as shorthand for the preceding properties.
* The **marker-offset** specifies the distance between a marker and the text in the list.

Now, we will see how to use these properties with examples.

The list-style-type Property

The *list-style-type* property allows you to control the shape or style of bullet point (also known as a marker) in the case of unordered lists and the style of numbering characters in ordered lists.

Here are the values which can be used for an unordered list −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **none**  NA |
| 2 | **disc (default)**  A filled-in circle |
| 3 | **circle**  An empty circle |
| 4 | **square**  A filled-in square |

Here are the values, which can be used for an ordered list −

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Example** |
| decimal | Number | 1,2,3,4,5 |
| decimal-leading-zero | 0 before the number | 01, 02, 03, 04, 05 |
| lower-alpha | Lowercase alphanumeric characters | a, b, c, d, e |
| upper-alpha | Uppercase alphanumeric characters | A, B, C, D, E |
| lower-roman | Lowercase Roman numerals | i, ii, iii, iv, v |
| upper-roman | Uppercase Roman numerals | I, II, III, IV, V |
| lower-greek | The marker is lower-greek | alpha, beta, gamma |
| lower-latin | The marker is lower-latin | a, b, c, d, e |
| upper-latin | The marker is upper-latin | A, B, C, D, E |
| hebrew | The marker is traditional Hebrew numbering |  |
| armenian | The marker is traditional Armenian numbering |  |
| georgian | The marker is traditional Georgian numbering |  |
| cjk-ideographic | The marker is plain ideographic numbers |  |
| hiragana | The marker is hiragana | a, i, u, e, o, ka, ki |
| katakana | The marker is katakana | A, I, U, E, O, KA, KI |
| hiragana-iroha | The marker is hiragana-iroha | i, ro, ha, ni, ho, he, to |
| katakana-iroha | The marker is katakana-iroha | I, RO, HA, NI, HO, HE, TO |

The *padding* property allows you to specify how much space should appear between the content of an element and its border −

The value of this attribute should be either a length, a percentage, or the word *inherit*. If the value is *inherit*, it will have the same padding as its parent element. If a percentage is used, the percentage is of the containing box.

The following CSS properties can be used to control lists. You can also set different values for the padding on each side of the box using the following properties −

* The **padding-bottom** specifies the bottom padding of an element.
* The **padding-top** specifies the top padding of an element.
* The **padding-left** specifies the left padding of an element.
* The **padding-right** specifies the right padding of an element.
* The **padding** serves as shorthand for the preceding properties.

Now, we will see how to use these properties with examples.

The padding-bottom Property

The *padding-bottom* property sets the bottom padding (space) of an element. This can take a value in terms of length of %.

Here is an example −

<html>

<head>

</head>

<body>

<p style = "padding-bottom: 15px; border:1px solid black;">

This is a paragraph with a specified bottom padding

</p>

<p style = "padding-bottom: 5%; border:1px solid black;">

This is another paragraph with a specified bottom padding in percent

</p>

</body>

</html>

The *cursor* property of CSS allows you to specify the type of cursor that should be displayed to the user.

One good usage of this property is in using images for submit buttons on forms. By default, when a cursor hovers over a link, the cursor changes from a pointer to a hand. However, it does not change form for a submit button on a form. Therefore, whenever someone hovers over an image that is a submit button, it provides a visual clue that the image is clickable.

The following table shows the possible values for the cursor property −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **auto**  Shape of the cursor depends on the context area it is over. For example an I over text, a hand over a link, and so on... |
| 2 | **crosshair**  A crosshair or plus sign |
| 3 | **default**  An arrow |
| 4 | **pointer**  A pointing hand (in IE 4 this value is hand) |
| 5 | **move**  The I bar |
| 6 | **e-resize**  The cursor indicates that an edge of a box is to be moved right (east) |
| 7 | **ne-resize**  The cursor indicates that an edge of a box is to be moved up and right (north/east) |
| 8 | **nw-resize**  The cursor indicates that an edge of a box is to be moved up and left (north/west) |
| 9 | **n-resize**  The cursor indicates that an edge of a box is to be moved up (north) |
| 10 | **se-resize**  The cursor indicates that an edge of a box is to be moved down and right (south/east) |
| 11 | **sw-resize**  The cursor indicates that an edge of a box is to be moved down and left (south/west) |
| 12 | **s-resize**  The cursor indicates that an edge of a box is to be moved down (south) |
| 13 | **w-resize**  The cursor indicates that an edge of a box is to be moved left (west) |
| 14 | **text**  The I bar |
| 15 | **wait**  An hour glass |
| 16 | **help**  A question mark or balloon, ideal for use over help buttons |
| 17 | **<url>**  The source of a cursor image file |

**NOTE** − You should try to use only these values to add helpful information for users, and in places, they would expect to see that cursor. For example, using the crosshair when someone hovers over a link can confuse visitors.

Here is an example −

<html>

<head>

</head>

<body>

<p>Move the mouse over the words to see the cursor change:</p>

<div style = "cursor:auto">Auto</div>

<div style = "cursor:crosshair">Crosshair</div>

<div style = "cursor:default">Default</div>

<div style = "cursor:pointer">Pointer</div>

<div style = "cursor:move">Move</div>

<div style = "cursor:e-resize">e-resize</div>

<div style = "cursor:ne-resize">ne-resize</div>

<div style = "cursor:nw-resize">nw-resize</div>

<div style = "cursor:n-resize">n-resize</div>

<div style = "cursor:se-resize">se-resize</div>

<div style = "cursor:sw-resize">sw-resize</div>

<div style = "cursor:s-resize">s-resize</div>

<div style = "cursor:w-resize">w-resize</div>

<div style = "cursor:text">text</div>

<div style = "cursor:wait">wait</div>

<div style = "cursor:help">help</div>

</body>

</html>

**CSS Outline Property**

Outlines are very similar to borders, but there are few major differences as well −

* An outline does not take up space.
* Outlines do not have to be rectangular.
* Outline is always the same on all sides; you cannot specify different values for different sides of an element.

**NOTE** − The outline properties are not supported by IE 6 or Netscape 7.

You can set the following outline properties using CSS.

* The **outline-width** property is used to set the width of the outline.
* The **outline-style** property is used to set the line style for the outline.
* The **outline-color** property is used to set the color of the outline.
* The **outline** property is used to set all the above three properties in a single statement.

The outline-width Property

The *outline-width* property specifies the width of the outline to be added to the box. Its value should be a length or one of the values *thin, medium, or thick,* just like the border-width attribute.

A width of zero pixels means no outline.

Here is an example −

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;">

This text is having thin outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:solid;">

This text is having thick outline.

</p>

<br />

<p style = "outline-width:5px; outline-style:solid;">

This text is having 5x outline.

</p>

</body>

</html>

The outline-style Property

The *outline-style* property specifies the style for the line (solid, dotted, or dashed) that goes around an element. It can take one of the following values −

* **none** − No border. (Equivalent of outline-width:0;)
* **solid** − Outline is a single solid line.
* **dotted** − Outline is a series of dots.
* **dashed** − Outline is a series of short lines.
* **double** − Outline is two solid lines.
* **groove** − Outline looks as though it is carved into the page.
* **ridge** − Outline looks the opposite of groove.
* **inset** − Outline makes the box look like it is embedded in the page.
* **outset** − Outline makes the box look like it is coming out of the canvas.
* **hidden** − Same as none.

Here is an example −

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;">

This text is having thin solid outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:dashed;">

This text is having thick dashed outline.

</p>

<br />

<p style = "outline-width:5px;outline-style:dotted;">

This text is having 5x dotted outline.

</p>

</body>

</html>

The outline-color Property

The *outline-color* property allows you to specify the color of the outline. Its value should either be a color name, a hex color, or an RGB value, as with the color and border-color properties.

Here is an example −

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;outline-color:red">

This text is having thin solid red outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:dashed;outline-color:#009900">

This text is having thick dashed green outline.

</p>

<br />

<p style = "outline-width:5px;outline-style:dotted;outline-color:rgb(13,33,232)">

This text is having 5x dotted blue outline.

</p>

</body>

</html>

The outline Property

The *outline* property is a shorthand property that allows you to specify values for any of the three properties discussed previously in any order but in a single statement.

Here is an example −

<html>

<head>

</head>

<body>

<p style = "outline:thin solid red;">

This text is having thin solid red outline.

</p>

<br />

<p style = "outline:thick dashed #009900;">

This text is having thick dashed green outline.

</p>

<br />

<p style = "outline:5px dotted rgb(13,33,232);">

This text is having 5x dotted blue outline.

</p>

</body>

</html>

**CSS Scrollbars**

There may be a case when an element's content might be larger than the amount of space allocated to it. For example, given width and height properties do not allow enough room to accommodate the content of the element.

CSS provides a property called *overflow* which tells the browser what to do if the box's contents is larger than the box itself. This property can take one of the following values −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **visible**  Allows the content to overflow the borders of its containing element. |
| 2 | **hidden**  The content of the nested element is simply cut off at the border of the containing element and no scrollbars is visible. |
| 3 | **scroll**  The size of the containing element does not change, but the scrollbars are added to allow the user to scroll to see the content. |
| 4 | **auto**  The purpose is the same as scroll, but the scrollbar will be shown only if the content does overflow. |

Here is an example −

<html>

<head>

<style type = "text/css">

.scroll {

display:block;

border: 1px solid red;

padding:5px;

margin-top:5px;

width:300px;

height:50px;

overflow:scroll;

}

.auto {

display:block;

border: 1px solid red;

padding:5px;

margin-top:5px;

width:300px;

height:50px;

overflow:auto;

}

</style>

</head>

<body>

<p>Example of scroll value:</p>

<div class = "scroll">

I am going to keep lot of content here just to show you how

scrollbars works if there is an overflow in an element box.

This provides your horizontal as well as vertical scrollbars.

</div>

<br />

<p>Example of auto value:</p>

<div class = "auto">

I am going to keep lot of content here just to show you how

scrollbars works if there is an overflow in an element box.

This provides your horizontal as well as vertical scrollbars.

</div>

</body>

</html>