

# CSS Background Properties

Property	Description
<u>background</u>	Sets all the background properties in one declaration
<u>background-attachment</u>	Sets whether a background image is fixed or scrolls with the rest of the page
<u>background-color</u>	Sets the background color of an element
<u>background-image</u>	Sets the background image for an element
<u>background-position</u>	Sets the starting position of a background image
<u>background-repeat</u>	Sets how a background image will be repeated

## CSS Background

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

## Background Color

The `background-color` property specifies the background color of an element.

The background color of a page is set like this:

### Example

```
body {  
    background-color: #b0c4de;  
}
```

In the example below, the `<h1>`, `<p>`, and `<div>` elements have different background colors:

### Example

```
h1 {  
    background-color: #6495ed;  
}  
  
p {  
    background-color: #e0ffff;  
}  
  
div {  
    background-color: #b0c4de;  
}
```

EXAMPLE PROGRAM:

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {  
    background-color: #6495ed;  
}
```

```
p {  
    background-color: #e0ffff;
```

```
}

div {
  background-color: #b0c4de;
}

</style>

</head>

<body>

<h1>CSS background-color example!</h1>

<div>

This is a text inside a div element.

<p>This paragraph has its own background color.</p>

We are still in the div element.

</div>

</body>

</html>
```

## OUTPUT:

**CSS background-color example!**

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

## Background Image

The `background-image` property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

### Example

```
body {  
    background-image: url("paper.gif");  
}
```

EXAMPLE PROGRAM:

```
<html>  
  
<head>  
  
<style>  
  
body {  
  
    background-image: url("paper.gif");  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
  
  
  
  
  
  
  
  
<h1>Hello World!</h1>  
  
  
  
</body>  
  
</html>
```

OUTPUT:

# Hello World!

(BACKGROUND IMAGE WILL BE DISPLAYED HERE)

## Background Image - Repeat Horizontally or Vertically

By default, the `background-image` property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

### Example

```
body {  
  background-image: url("gradient_bg.png");  
}
```

If the image is repeated only horizontally (`repeat-x`), the background will look better:

### Example

```
body {  
  background-image: url("gradient_bg.png");  
  background-repeat: repeat-x;  
}
```

Showing the image only once is specified by the `background-repeat` property:

### Example

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
}
```

The position of the image is specified by the `background-position` property:

### Example

```
body {  
  background-image: url("img_tree.png");
```

```
background-repeat: no-repeat;
background-position: right top;
}
```

## Background - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.

The shorthand property for background is simply "`background`":

### Example

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

## CSS Text Properties

Property	Description
<u>color</u>	Sets the color of text
<u>direction</u>	Specifies the text direction/writing direction
<u>letter-spacing</u>	Increases or decreases the space between characters in a text
<u>line-height</u>	Sets the line height
<u>text-align</u>	Specifies the horizontal alignment of text
<u>text-decoration</u>	Specifies the decoration added to text
<u>text-indent</u>	Specifies the indentation of the first line in a text-block
<u>text-shadow</u>	Specifies the shadow effect added to text

<u>text-transform</u>	Controls the capitalization of text
<u>unicode-bidi</u>	Used together with the <u>direction</u> property to set or return whether the text should be overridden to support multiple languages in the same document
<u>vertical-align</u>	Sets the vertical alignment of an element
<u>white-space</u>	Specifies how white-space inside an element is handled
<u>word-spacing</u>	Increases or decreases the space between words in a text

## Text Color

The `color` property is used to set the color of the text.

With CSS, a color is most often specified by:

- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"
- a color name - like "red"

### Example

Define different HEX colors:

```
#p1 {background-color: #ff0000;} /* red */
#p2 {background-color: #00ff00;} /* green */
#p3 {background-color: #0000ff;} /* blue */
```

### Example

Define different RGB colors:

```
#p1 {background-color: rgb(255, 0, 0);} /* red */
#p2 {background-color: rgb(0, 255, 0);} /* green */
#p3 {background-color: rgb(0, 0, 255);} /* blue */
```

### Example

Define different RGB colors with opacity:

```
#p1 {background-color: rgba(255, 0, 0, 0.3);} /* red with opacity */
#p2 {background-color: rgba(0, 255, 0, 0.3);} /* green with opacity
*/
#p3 {background-color: rgba(0, 0, 255, 0.3);} /* blue with opacity */
```

## Example

Define different HSL colors:

```
#p1 {background-color: hsl(120, 100%, 50%);} /* green */
#p2 {background-color: hsl(120, 100%, 75%);} /* light green */
#p3 {background-color: hsl(120, 100%, 25%);} /* dark green */
#p4 {background-color: hsl(120, 60%, 70%);} /* pastel green */
```

EXAMPLE:

```
<html>
<head>
<style>
  body {
    color: red;
  }

  h1 {
    color: #00ff00;
  }

  p.ex {
    color: rgb(0,0,255);
  }
</style>
</head>
<body>
```



```
<h1>This is heading 1</h1>
```

```
<p>This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.</p>
```

```
<p class="ex">This is a paragraph with class="ex". This text is blue.</p>
```

```
</body>
```

```
</html>
```

## OUTPUT:

### This is heading 1

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.

## Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align 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).

### Example

```
h1 {
  text-align: center;
}

p.date {
  text-align: right;
}

p.main {
  text-align: justify;
}
```

## EXAMPLE PROGRAM:

```
<html>
```

```
<head>
```

```
<style>
```

```
h1 {
```

```
    text-align: center;
```

```
}
```

```
p.date {
```

```
    text-align: right;
```

```
}
```

```
p.main {
```

```
    text-align: justify;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>CSS text-align Example</h1>
```

```
<p class="date">May, 2009</p>
```

```
<p class="main">In my younger and more vulnerable years my father gave me  
some advice that I've been turning over in my mind ever since. 'Whenever you feel  
like criticizing anyone,' he told me,
```

'just remember that all the people in this world haven't had the advantages that you've had.'

**Note:** Resize the browser window to see how the value "justify" works.

OUTPUT:

## CSS text-align Example

May, 2009

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

**Note:** Resize the browser window to see how the value "justify" works.

## Text Decoration

The `text-decoration` property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

### Example

```
a {  
  text-decoration: none;  
}
```

It can also be used to decorate text:

### Example

```
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}
```

#### EXAMPLE PROGRAM:

```
<html>  
  
<head>  
  
<style>  
  
h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}  
  
</style>  
  
</head>
```

```
<body>
```

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
</body>
```

```
</html>
```

OUTPUT:

**This is heading 1**

**~~This is heading 2~~**

**This is heading 3**

## Text Transformation

The `text-transform` property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

### Example

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize {
```

```
    text-transform: capitalize;
}
```

## EXAMPLE PROGRAM:

```
<html>
```

```
<head>
```

```
<style>
```

```
p.uppercase {
```

```
    text-transform: uppercase;
```

```
}
```

```
p.lowercase {
```

```
    text-transform: lowercase;
```

```
}
```

```
p.capitalize {
```

```
    text-transform: capitalize;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="uppercase">This is some text.</p>
```

```
<p class="lowercase">This is some text.</p>
```

```
<p class="capitalize">This is some text.</p>
```

```
</body>
```

```
</html>
```

OUTPUT:

THIS IS SOME TEXT.

this is some text.

This Is Some Text.

## CSS Font Properties

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text
<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

# CSS Font

CSS font properties define the font family, boldness, size, and the style of a text.

## Difference Between Serif and Sans-serif Fonts



## CSS Font Families

In CSS, there are two types of font family names:

- **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
- **font family** - a specific font family (like "Times New Roman" or "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends of characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have small lines at the ends of characters



Monospace

Courier New

All monospace characters have the

Lucida Console



**Note:** On computer screens, sans-serif fonts are considered easier to read than serif

## Font Family

The font family of a text is set with the `font-family` property.

The `font-family` property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

**Note:** If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

### Example

```
p {  
  font-family: "Times New Roman", Times, serif;  
}
```

For more commonly used font combinations, look at our [Web Safe Font Combinations](#).

## Font Style

The `font-style` property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

## Example

```
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}
```

## Font Size

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

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like `<h1>` - `<h6>` for headings and `<p>` for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

- Sets the text to a specified size
- Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
- Absolute size is useful when the physical size of the output is known

Relative size:

- Sets the size relative to surrounding elements
- Allows a user to change the text size in browsers



**Note:** If you do not specify a font size, the default size for normal text, like paragraphs (16px=1em).

# Set Font Size With Pixels

Setting the text size with pixels gives you full control over the text size:

## Example

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

h2 {
  font-size: 30px;
}

p {
  font-size: 14px;
}
```

**Tip:** If you use pixels, you can still use the zoom tool to resize the entire page.

# Set Font Size With Em

To allow users to resize the text (in the browser menu), many developers use em instead of pixels.

The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula:  $pixels/16=em$

## Example

```
h1 {
  font-size: 2.5em; /* 40px/16=2.5em */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}

p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
```

In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.

Unfortunately, there is still a problem with older versions of IE. The text becomes larger than it should when made larger, and smaller than it should when made smaller.

## Use a Combination of Percent and Em

The solution that works in all browsers, is to set a default font-size in percent for the <body> element:

### Example

```
body {  
    font-size: 100%;  
}  
  
h1 {  
    font-size: 2.5em;  
}  
  
h2 {  
    font-size: 1.875em;  
}  
  
p {  
    font-size: 0.875em;  
}
```

## CSS Links

### Styling Links

Links can be styled with any CSS property (e.g. `color`, `font-family`, `background`, etc.).

### Example

```
a {
  color: #FF0000;
}
```

In addition, links can be styled differently depending on what **state** they are in.

The four link states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

## Example

```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

## EXAMPLE PROGRAM:

```
<html>
```

```
<head>
```

```
<style>
```

```
/* unvisited link */
```

```
a:link {
```

```
    color: #FF0000;
}
```

```
/* visited link */
```

```
a:visited {
    color: #00FF00;
}
```

```
/* mouse over link */
```

```
a:hover {
    color: #FF00FF;
}
```

```
/* selected link */
```

```
a:active {
    color: #0000FF;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b><a href="default.asp" target="_blank">This is a
link</a></b></p>
```

**Note:** a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

**Note:** a:active MUST come after a:hover in the CSS definition in order to be effective.

</body>

</html>

**OUTPUT:**

[This is a link](#)

**Note:** a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

**Note:** a:active MUST come after a:hover in the CSS definition in order to be effective.

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

## Common Link Styles

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

## Text Decoration

The `text-decoration` property is mostly used to remove underlines from links:

Example

```
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a:active {
    text-decoration: underline;
}
```

## Background Color

The `background-color` property specifies the background color for links:

### Example

```
a:link {
    background-color: #B2FF99;
}

a:visited {
    background-color: #FFFF85;
}

a:hover {
    background-color: #FF704D;
}

a:active {
    background-color: #FF704D;
}
```

### EXAMPLE PROGRAM: CREATING ADVANCED LINKS

**<html>**

**<head>**

**<style>**



```
a.one:link {color:#ff0000;}
```

```
a.one:visited {color:#0000ff;}
```

```
a.one:hover {color:#ffcc00;}
```

```
a.two:link {color:#ff0000;}
```

```
a.two:visited {color:#0000ff;}
```

```
a.two:hover {font-size:150%;}
```

```
a.three:link {color:#ff0000;}
```

```
a.three:visited {color:#0000ff;}
```

```
a.three:hover {background:#66ff66;}
```

```
a.four:link {color:#ff0000;}
```

```
a.four:visited {color:#0000ff;}
```

```
a.four:hover {font-family:monospace;}
```

```
a.five:link {color:#ff0000;text-decoration:none;}
```

```
a.five:visited {color:#0000ff;text-decoration:none;}
```

```
a.five:hover {text-decoration:underline;}
```

```
</style>
```

```
</head>
```

```
<body>
```

<p>Mouse over the links and watch them change layout:</p>

<p><b><a class="one" href="default.asp" target="\_blank">This link changes color</a></b></p>

<p><b><a class="two" href="default.asp" target="\_blank">This link changes font-size</a></b></p>

<p><b><a class="three" href="default.asp" target="\_blank">This link changes background-color</a></b></p>

<p><b><a class="four" href="default.asp" target="\_blank">This link changes font-family</a></b></p>

<p><b><a class="five" href="default.asp" target="\_blank">This link changes text-decoration</a></b></p>

</body>

</html>

## OUTPUT:

Mouse over the links and watch them change layout:

[This link changes color](#)

[This link changes font-size](#)

[This link changes background-color](#)

[This link changes font-family](#)

[This link changes text-decoration](#)

## EXAMPLE-2

```
<html>
```

```
<head>
```

```
<style>
```

```
a:link, a:visited {
```

```
    display: block;
```

```
    font-weight: bold;
```

```
    color: #ffffff;
```

```
    background-color: #98bf21;
```

```
    width: 120px;
```

```
    text-align: center;
```

```
    padding: 4px;
```

```
    text-decoration: none;
```

```
}
```

```
a:hover, a:active {
```

```
    background-color: #9A991D;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<a href="default.asp" target="_blank">This is a link</a>
```

```
</body>
```

```
</html>
```

**OUTPUT:**

[This is a link](#)

# CSS Lists

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker

Property	Description
<u>list-style</u>	Sets all the properties for a list in one declaration
<u>list-style-image</u>	Specifies an image as the list-item marker
<u>list-style-position</u>	Specifies if the list-item markers should appear inside or outside the content flow
<u>list-style-type</u>	Specifies the type of list-item marker

## List

In HTML, there are two types of lists:

- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

## Different List Item Markers

The type of list item marker is specified with the `list-style-type` property:

### Example

```
ul.a {  
    list-style-type: circle;  
}
```

```
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {  
    list-style-type: lower-alpha;  
}
```

### EXAMPLE PROGRAM:

```
<html>
```

```
<head>
```

```
<style>
```

```
ul.a {
```

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

```
}
```

```
ul.b {
```

```
    list-style-type: square;
```

```
}
```

```
ol.c {
```

```
    list-style-type: upper-roman;
```

```
}
```

```
ol.d {  
    list-style-type: lower-alpha;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Example of unordered lists:</p>
```

```
<ul class="a">
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ul>
```

```
<ul class="b">
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ul>
```

```
<p>Example of ordered lists:</p>
```

```
<ol class="c">
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ol>
```

```
<ol class="d">
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

## OUTPUT:

Example of unordered lists:

- Coffee
- Tea
- Coca Cola
  
- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola



- a. Coffee
- b. Tea
- c. Coca Cola

## An Image as The List Item Marker

To specify an image as the list item marker, use the `list-style-image` property:

### Example

```
ul {  
  list-style-image: url('sqpurple.gif');  
}
```

### EXAMPLE:

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
  list-style-image: url('sqpurple.gif');
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
  <li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

### OUTPUT:

- Coffee
- Tea
- Coca Cola

# CSS Tables

## Table Borders

To specify table borders in CSS, use the `border` property.

The example below specifies a black border for `<table>`, `<th>`, and `<td>` elements:

### Example

```
table, th, td {  
  border: 1px solid black;  
}
```

### EXAMPLE PROGRAM-1:

```
<html>
```

```
<head>
```

```
<style>
```

```
table, th, td {  
    border: 1px solid black;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<th>Firstname</th>
```

```
<th>Lastname</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Peter</td>
```

```
<td>Griffin</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Lois</td>
```

```
<td>Griffin</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

**OUTPUT:**

---

Firstname	Lastname
Peter	Griffin
Lois	Griffin

## Table Width and Height

Width and height of a table is defined by the `width` and `height` properties.

The example below sets the width of the table to 100%, and the height of the `<th>` elements to 50px:

### Example

```
table {  
    width: 100%;  
}  
  
th {  
    height: 50px;  
}
```

### EXAMPLE PROGRAM-2

```
<html>
```

```
<head>
```

```
<style>
```

```
table, td, th {
```

```
    border: 1px solid black;
```

```
}
```

```
table {  
    width: 100%;  
}
```

```
th {  
    height: 50px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<th>Firstname</th>
```

```
<th>Lastname</th>
```

```
<th>Savings</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Peter</td>
```

```
<td>Griffin</td>
```

```
<td>$100</td>
```

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>\$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>\$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>\$250</td>

</tr>

</table>

</body>

</html>

## OUTPUT:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

## Horizontal Text Alignment

The `text-align` property sets the horizontal alignment, like left, right, or center.

By default, the text in `<th>` elements are center-aligned and the text in `<td>` elements are left-aligned.

The following example left-aligns the text in `<th>` elements:

### Example

```
th {  
    text-align: left;  
}
```

## Vertical Text Alignment

The `vertical-align` property sets the vertical alignment, like top, bottom, or middle.

By default, the vertical alignment of text in a table is middle (for both `<th>` and `<td>` elements).

The following example sets the vertical text alignment to bottom for `<td>` elements:

### Example

```
td {  
    height: 50px;  
    vertical-align: bottom;  
}
```

## Table Padding

To control the space between the border and content in a table, use the padding property on <td> and <th> elements:

### Example

```
td {  
    padding: 15px;  
}
```

### EXAMPLE PROGRAM-3:

```
<head>
```

```
<style>
```

```
table, td, th {
```

```
    border: 1px solid black;
```

```
}
```

```
td {
```

```
    height: 50px;
```

```
    vertical-align: bottom;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
    <th>Firstname</th>
```



<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>\$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>\$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>\$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>\$250</td>

`</tr>`

`</table>`

`</body>`

`</html>`

**OUTPUT:**

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

**EXAMPLE PROGRAM-4**

`<html>`

`<head>`

`<style>`

`table, td, th {`

`border: 1px solid black;`

`}`

`td {`

```
padding: 15px;
}
</style>
</head>
<body>

<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Griffin</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Lois</td>
    <td>Griffin</td>
    <td>$150</td>
  </tr>
```

```
<tr>
  <td>Joe</td>
  <td>Swanson</td>
  <td>$300</td>
</tr>
<tr>
  <td>Cleveland</td>
  <td>Brown</td>
  <td>$250</td>
</tr>
</table>
</body>
</html>
```

**OUTPUT:**

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

# Table Color

The example below specifies the color of the borders, and the text and background color of <th> elements:

## Example

```
table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}
```

## EXAMPLE-4

```
<html>
```

```
<head>
```

```
<style>
```

```
table, td, th {
```

```
    border: 1px solid green;
```

```
}
```

```
th {
```

```
    background-color: green;
```

```
    color: white;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<th>Firstname</th>
```

```
<th>Lastname</th>
```

```
<th>Savings</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Peter</td>
```

```
<td>Griffin</td>
```

```
<td>$100</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Lois</td>
```

```
<td>Griffin</td>
```

```
<td>$150</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Joe</td>
```

```
<td>Swanson</td>
```

```
<td>$300</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>Cleveland</td>
```

```
  <td>Brown</td>
```

```
  <td>$250</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

**OUTPUT:**

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

## The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to add a border around elements, and to define space between elements.

The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

## Example

```
div {  
  width: 300px;  
  padding: 25px;  
  border: 25px solid navy;  
  margin: 25px;  
}
```

## EXAMPLE PROGRAM-1

**<html>**

**<head>**

**<style>**

**div {**

**background-color: lightgrey;**

**width: 300px;**



```
padding: 25px;

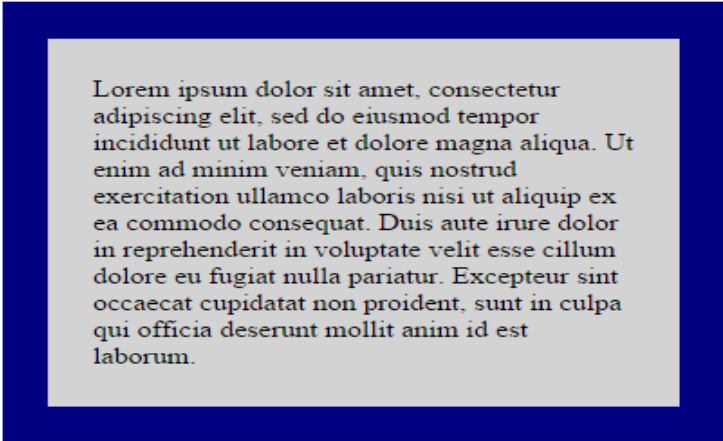
border: 25px solid navy;

margin: 25px;
}
</style>
</head>
<body>

<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

**OUTPUT:**



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## EXAMPLE PROGRAM-2

```
<html>
<head>
<style>
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
</style>
</head>
<body>



<div>The picture above is 350px wide. The total width of this element is
also 350px.</div>

</body>
</html>
```

## OUTPUT:



The picture above is 350px wide. The total width of this element is also 350px.

## CSS Border Properties

Property	Description
<u><a href="#">border</a></u>	Sets all the border properties in one declaration
<u><a href="#">border-bottom</a></u>	Sets all the bottom border properties in one declaration
<u><a href="#">border-bottom-color</a></u>	Sets the color of the bottom border
<u><a href="#">border-bottom-style</a></u>	Sets the style of the bottom border
<u><a href="#">border-bottom-width</a></u>	Sets the width of the bottom border
<u><a href="#">border-color</a></u>	Sets the color of the four borders
<u><a href="#">border-left</a></u>	Sets all the left border properties in one declaration

<u>border-left-color</u>	Sets the color of the left border
<u>border-left-style</u>	Sets the style of the left border
<u>border-left-width</u>	Sets the width of the left border
<u>border-right</u>	Sets all the right border properties in one declaration
<u>border-right-color</u>	Sets the color of the right border
<u>border-right-style</u>	Sets the style of the right border
<u>border-right-width</u>	Sets the width of the right border
<u>border-style</u>	Sets the style of the four borders
<u>border-top</u>	Sets all the top border properties in one declaration
<u>border-top-color</u>	Sets the color of the top border
<u>border-top-style</u>	Sets the style of the top border
<u>border-top-width</u>	Sets the width of the top border
<u>border-width</u>	Sets the width of the four borders

### EXAMPLE-1

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  border-style: dotted;
```

```
border-width: 4px;
border-color: red;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

OUTPUT:

**This is a Heading**

This is a paragraph.

EXAMPLE-2

```
<html>
<head>
<style>
```

```
p {  
    border: 10px solid green;  
}  
  
</style>  
  
</head>  
  
<body>  
  
<h1>This is a Heading</h1>  
  
<p>This is a paragraph.</p>  
  
</body>  
  
</html>
```

**OUTPUT:**

**This is a Heading**

This is a paragraph.

# CSS Combinators



A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS3:

- descendant selector
- child selector
- adjacent sibling selector
- general sibling selector

## Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

### Example

```
div p {  
    background-color: yellow;  
}
```

### EXAMPLE PROGRAM-1

```
<html>
```

```
<head>
```

```
<style>
```

```
div p {
```

```
    background-color: yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<p>Paragraph 1 in the div.</p>
```

```
<p>Paragraph 2 in the div.</p>
```

```
<span><p>Paragraph 3 in the div.</p></span>
```

```
</div>
```

```
<p>Paragraph 4. Not in a div.</p>
```

```
<p>Paragraph 5. Not in a div.</p>
```

```
</body>
```

```
</html>
```

## OUTPUT:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.



# Child Selector

The child selector selects all elements that are the immediate children of a specified element.

The following example selects all <p> elements that are immediate children of a <div> element:

## Example

```
div > p {  
    background-color: yellow;  
}
```

## EXAMPLE-2

```
<html>
```

```
<head>
```

```
<style>
```

```
div > p {
```

```
    background-color: yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<p>Paragraph 1 in the div.</p>
```

```
<p>Paragraph 2 in the div.</p>
```

```
<span><p>Paragraph 3 in the div.</p></span> <!-- not Child but  
Descendant -->
```

`</div>`

`<p>Paragraph 4. Not in a div.</p>`

`<p>Paragraph 5. Not in a div.</p>`

`</body>`

`</html>`

## OUTPUT:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

## Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all `<p>` elements that are placed immediately after `<div>` elements:

### Example

```
div + p {  
    background-color: yellow;  
}
```

## EXAMPLE PROGRAM 2:

```
<html>
<head>
<style>
  div + p {
    background-color: yellow;
  }
</style>
</head>
<body>

<div>
<p>Paragraph 1 in the div.</p>
<p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in a div.</p>
<p>Paragraph 4. Not in a div.</p>

</body>
</html>
```

## OUTPUT:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

## General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all `<p>` elements that are siblings of `<div>` elements:

### EXAMPLE PROGRAM-4

```
<html>
```

```
<head>
```

```
<style>
```

```
div ~ p {
```

```
    background-color: yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<p>Paragraph 1 in the div.</p>
```

```
<p>Paragraph 2 in the div.</p>
```

```
</div>
```

```
<p>Paragraph 3. Not in a div.</p>
```

```
<p>Paragraph 4. Not in a div.</p>
```

```
</body>
```

```
</html>
```

## OUTPUT:

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

## The ::first-line Pseudo-element

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all `<p>` elements:

### Example

```
p::first-line {  
  color: #ff0000;  
  font-variant: small-caps;  
}
```



## OUTPUT:

YOU CAN USE THE `::FIRST-LINE` PSEUDO-ELEMENT TO ADD A SPECIAL EFFECT TO THE FIRST LINE OF A TEXT.  
Some more text. And even more, and more, and more, and more, and more, and more, and more,  
and more, and more, and more, and more, and more.

The following properties apply to the `::first-line` pseudo-element:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

## The `::first-letter` Pseudo-element

The `::first-letter` pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all `<p>` elements:

### Example

```
p::first-letter {  
    color: #ff0000;  
    font-size: xx-large;  
}
```

## EXAMPLE PROGRAM-2

`<html>`

`<head>`

`<style>`

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

**<p>You can use the ::first-letter pseudo-element to add a special effect to the first character of a text!</p>**

```
</body>
```

```
</html>
```

**OUTPUT:**

**Y**ou can use the ::first-letter pseudo-element to add a special effect to the first character of a text!

## CSS - The ::before Pseudo-element

The `::before` pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before each `<h1>` element:

Example

```
h1::before {  
  content: url(smiley.gif);  
}
```



## EXAMPLE PROGRAM:

```
<html>
<head>
<style>
h1::before {
    content: url(smiley.gif);
}
</style>
</head>
<body>

<h1>This is a heading</h1>

<p>The ::before pseudo-element inserts content before an
element.</p>

<h1>This is a heading</h1>

<p><b>Note:</b> IE8 supports the content property only if a !DOCTYPE
is specified.</p>

</body>
</html>
```

## OUTPUT:

## This is a heading

The `::before` pseudo-element inserts content before an element.

## This is a heading

**Note:** IE8 supports the content property only if a `!DOCTYPE` is specified.

## CSS Pseudo Elements

Selector	Example	Example description
<u><code>::after</code></u>	<code>p::after</code>	Insert content after every <code>&lt;p&gt;</code> element
<u><code>::before</code></u>	<code>p::before</code>	Insert content before every <code>&lt;p&gt;</code> element
<u><code>::first-letter</code></u>	<code>p::first-letter</code>	Selects the first letter of every <code>&lt;p&gt;</code> element
<u><code>::first-line</code></u>	<code>p::first-line</code>	Selects the first line of every <code>&lt;p&gt;</code> element
<u><code>::selection</code></u>	<code>p::selection</code>	Selects the portion of an element that is selected

## All CSS Pseudo Classes

Selector	Example	Example description
<u><code>:active</code></u>	<code>a:active</code>	Selects the active link
<u><code>:checked</code></u>	<code>input:checked</code>	Selects every checked <code>&lt;input&gt;</code> element
<u><code>:disabled</code></u>	<code>input:disabled</code>	Selects every disabled <code>&lt;input&gt;</code> element

<u>:empty</u>	p:empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p:first-child	Selects every <p> elements that is the first child of its parent
<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<u>:focus</u>	input:focus	Selects the <input> element that has focus
<u>:hover</u>	a:hover	Selects links on mouse over
<u>:in-range</u>	input:in-range	Selects <input> elements with a value within a specified range
<u>:invalid</u>	input:invalid	Selects all <input> elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<u>:last-child</u>	p:last-child	Selects every <p> elements that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links
<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that

		is the second child of its parent
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects <input> elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects <input> elements with a value outside a specified range
<u>:read-only</u>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects <input> elements with no "readonly" attribute
<u>:required</u>	input:required	Selects <input> elements with a "required" attribute specified
<u>:root</u>	Root	Selects the document's root element
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL

		containing that anchor name)
<u>:valid</u>	input:valid	Selects all <input> elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links