

CSS Backgrounds

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: lightblue;  
}
```

With CSS, a color is most often specified by:

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

Look at [CSS Color Values](#) for a complete list of possible color values.

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

Example

```
h1 {  
    background-color: green;  
}  
  
div {  
    background-color: lightblue;  
}  
  
p {  
    background-color: yellow;  
}
```

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");  
}
```

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

Example

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

Background Image - Set position and no-repeat

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

Example

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

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

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 Image - Fixed position

To specify that the background image should be fixed (will not scroll with the rest of the page), use the `background-attachment` property:

Example

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

CSS Borders

Border Style

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border
- `groove` - Defines a 3D grooved border. The effect depends on the border-color value
- `ridge` - Defines a 3D ridged border. The effect depends on the border-color value
- `inset` - Defines a 3D inset border. The effect depends on the border-color value
- `outset` - Defines a 3D outset border. The effect depends on the border-color value
- `none` - Defines no border
- `hidden` - Defines a hidden border

The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

Result:

=====A dotted border.=====

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

Class Work

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.dotted {border-style: dotted;}
```

```
p.dashed {border-style: dashed;}
```

```
p.solid {border-style: solid;}
```

```
p.double {border-style: double;}
```

```
p.groove {border-style: groove;}
```

```
p.ridge {border-style: ridge;}
```

```
p.inset {border-style: inset;}
```

```
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}  
</style>  
</head>  
<body>
```

```
<h2>The border-style Property</h2>
```

```
<p>This property specifies what kind of border to display:</p>
```

```
<p class="dotted">A dotted border.</p>  
<p class="dashed">A dashed border.</p>  
<p class="solid">A solid border.</p>  
<p class="double">A double border.</p>  
<p class="groove">A groove border.</p>  
<p class="ridge">A ridge border.</p>  
<p class="inset">An inset border.</p>  
<p class="outset">An outset border.</p>  
<p class="none">No border.</p>  
<p class="hidden">A hidden border.</p>  
<p class="mix">A mixed border.</p>
```

```
</body>
```

```
</html>
```

Border Width

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

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.

The `border-width` property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example

```
p.one {  
    border-style: solid;  
    border-width: 5px;  
}  
  
p.two {  
    border-style: solid;  
    border-width: medium;  
}  
  
p.three {  
    border-style: solid;  
    border-width: 2px 10px 4px 20px;  
}
```

Class Work

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.one {
```

```
    border-style: solid;
```

```
    border-width: 5px;
```

```
}
```

```
p.two {
```

```
    border-style: solid;
```

```
    border-width: medium;
```

```
}
```

```
p.three {
```

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

```
    border-width: 2px;
```

```
}
```

```
p.four {
```

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

```
    border-width: thick;
```

```
}
```

```
p.five {
```

```
border-style: double;
border-width: 15px;
}
```

```
p.six {
border-style: double;
border-width: thick;
}
```

```
p.seven {
border-style: solid;
border-width: 2px 10px 4px 20px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>The border-width Property</h2>
```

```
<p>This property specifies the width of the four borders:</p>
```

```
<p class="one">Some text.</p>
```

```
<p class="two">Some text.</p>
```

```
<p class="three">Some text.</p>
```

```
<p class="four">Some text.</p>
```

```
<p class="five">Some text.</p>
```

```
<p class="six">Some text.</p>
```

```
<p class="seven">Some text.</p>
```

<p>Note: The "border-width" property does not work if it is used alone.

Always specify the "border-style" property to set the borders first.</p>

```
</body>
```

```
</html>
```

Border Color

The `border-color` property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- Hex - specify a hex value, like "#ff0000"
- transparent

The `border-color` property can have from one to four values (for the top border, right border, bottom border, and the left border).

If `border-color` is not set, it inherits the color of the element.

Example

```
p.one {  
    border-style: solid;  
    border-color: red;  
}
```

```
p.two {  
    border-style: solid;  
    border-color: green;  
}
```

```
p.three {  
    border-style: solid;
```

```
    border-color: red green blue yellow;
}
```

Class Work

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.one {
```

```
    border-style: solid;
```

```
    border-color: red;
```

```
}
```

```
p.two {
```

```
    border-style: solid;
```

```
    border-color: green;
```

```
}
```

```
p.three {
```

```
    border-style: solid;
```

```
    border-color: red green blue yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>The border-color Property</h2>
```

<p>This property specifies the color of the four borders:</p>

<p class="one">A solid red border</p>

<p class="two">A solid green border</p>

<p class="three">A solid multicolor border</p>

<p>Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>

</body>

</html>

Border - Individual Sides

From the examples above you have seen that it is possible to specify a different border for each side.

In CSS, there is also properties for specifying each of the borders (top, right, bottom, and left):

Example

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Class Work

```
<!DOCTYPE html>

<html>

<head>

<style>

p {

    border-top-style: dotted;

    border-right-style: solid;

    border-bottom-style: dotted;

    border-left-style: solid;

}

</style>

</head>

<body>


<p>2 different border styles.</p>


</body>

</html>
```

The example above gives the same result as this:

Example

```
p {
    border-style: dotted solid;
}
```

Border - Shorthand Property

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

To shorten the code, it is also possible to specify all the individual border properties in one property.

The `border` property is a shorthand property for the following individual border properties:

- `border-width`
- `border-style` (required)
- `border-color`

Example

```
p {  
    border: 5px solid red;  
}
```

CSS Margins

The CSS margin properties set the size of the white space OUTSIDE the border.

Note: The margins are completely transparent - and cannot have a background color!

With CSS, you have full control over the margins. There are CSS properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties can have the following values:

- `auto` - the browser calculates the margin
- `length` - specifies a margin in px, pt, cm, etc.
- `%` - specifies a margin in % of the width of the containing element
- `inherit` - specifies that the margin should be inherited from the parent element

Note: It is also possible to use negative values for margins; to overlap content.

The following example sets different margins for all four sides of a `<p>` element:

Example

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

Class Work

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

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

```
p.ex {  
    border:1px solid red;  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

<h2>Using Individual margin Properties:</h2>

<p>This is a paragraph with no specified margins.</p>

<p class="ex">This paragraph has a top and bottom margin of 100px, a left margin of 80px, and a right margin of 150px.</p>

</body>

</html>

The following example lets the left margin be inherited from the parent element:

Example

```
div.container {  
    border: 1px solid red;  
    margin-left: 100px;  
}
```

```
p.one {  
    margin-left: inherit;  
}
```

Class Work

<!DOCTYPE html>

<html>

<head>

<style>

```
div.container {  
    border: 1px solid red;  
    margin-left: 100px;  
}
```

```
p.one {  
    margin-left: inherit;
```

```
}  
</style>  
</head>  
<body>  
  
<h2>Use of the inherit Value</h2>  
  
<p>Let the left margin be inherited from the parent element:</p>  
  
<div class="container">  
  
<p class="one">This is a paragraph with an inherited left margin (from the div  
element).</p>  
  
</div>  
  
</body>  
</html>
```

Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The `margin` property is a shorthand property for the following individual margin properties:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Example

```
p {  
    margin: 100px 150px 100px 80px;  
}
```

Use of The auto Value

You can set the margin property to `auto` to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

Example

```
div {  
    width: 300px;  
    margin: auto;  
    border: 1px solid red;  
}
```

Class Work

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<style>  
  
div {  
    width:300px;  
    margin: auto;  
    border: 1px solid red;  
}  
  
</style>  
  
</head>
```

<body>

<h2>Use of the auto Value</h2>

<p>You can set the margin property to auto to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:</p>

<div>

This div will be centered because it has margin: auto;

</div>

</body>

</html>