

# Chapter 8: Margins

| Parameter       | Details   |
|-----------------|---|
| 0               | set margin to none  |
| auto            | used for centering, by evenly setting values on each side |
| units (e.g. px) | see parameter section in Units for a list of valid units  |
| inherit         | inherit margin value from parent element                  |
| initial         | restore to initial value                                  |

## Section 8.1: Margin Collapsing

When two margins are touching each other vertically, they are collapsed. When two margins touch horizontally, they do not collapse.

### Example of adjacent vertical margins:

Consider the following styles and markup:

```
div{
  margin: 10px;
}

<div>
  some content
</div>
<div>
  some more content
</div>
```

They will be 10px apart since vertical margins collapse over one and other. (The spacing will not be the sum of two margins.)

### Example of adjacent horizontal margins:

Consider the following styles and markup:

```
span{
  margin: 10px;
}

<span>some</span><span>content</span>
```

They will be 20px apart since horizontal margins don't collapse over one and other. (The spacing will be the sum of two margins.)

### Overlapping with different sizes

```
.top{
  margin: 10px;
}
.bottom{
  margin: 15px;
}

<div class="top">
  some content
```

```
</div>
<div class="bottom">
  some more content
</div>
```

These elements will be spaced 15px apart vertically. The margins overlap as much as they can, but the larger margin will determine the spacing between the elements.

### Overlapping margin gotcha

```
.outer-top{
  margin: 10px;
}
.inner-top{
  margin: 15px;
}
.outer-bottom{
  margin: 20px;
}
.inner-bottom{
  margin: 25px;
}

<div class="outer-top">
  <div class="inner-top">
    some content
  </div>
</div>
<div class="outer-bottom">
  <div class="inner-bottom">
    some more content
  </div>
</div>
```

What will be the spacing between the two texts? (hover to see answer)

The spacing will be 25px. Since all four margins are touching each other, they will collapse, thus using the largest margin of the four.

Now, what about if we add some borders to the markup above.

```
div{
  border: 1px solid red;
}
```

What will be the spacing between the two texts? (hover to see answer)

The spacing will be 59px! Now only the margins of .outer-top and .outer-bottom touch each other, and are the only collapsed margins. The remaining margins are separated by the borders. So we have 1px + 10px + 1px + 15px + 20px + 1px + 25px + 1px. (The 1px's are the borders...)

### Collapsing Margins Between Parent and Child Elements:

HTML:

```
<h1>Title</h1>
<div>
  <p>Paragraph</p>
</div>
```

CSS

```
h1 {
  margin: 0;
  background: #cff;
}
div {
  margin: 50px 0 0 0;
  background: #cfc;
}
p {
  margin: 25px 0 0 0;
  background: #cf9;
}
```

In the example above, only the largest margin applies. You may have expected that the paragraph would be located 60px from the h1 (since the div element has a margin-top of 40px and the p has a 20px margin-top). This does not happen because the margins collapse together to form one margin.

## Section 8.2: Apply Margin on a Given Side

### Direction-Specific Properties

CSS allows you to specify a given side to apply margin to. The four properties provided for this purpose are:

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

The following code would apply a margin of 30 pixels to the left side of the selected div. [View Result](#)

### HTML

```
<div id="myDiv"></div>
```

### CSS

```
#myDiv {
  margin-left: 30px;
  height: 40px;
  width: 40px;
  background-color: red;
}
```

#### Parameter

#### Details

margin-left The direction in which the margin should be applied.

30px The width of the margin.

### Specifying Direction Using Shorthand Property

The standard margin property can be expanded to specify differing widths to each side of the selected elements. The syntax for doing this is as follows:

```
margin: <top> <right> <bottom> <left>;
```

The following example applies a zero-width margin to the top of the div, a 10px margin to the right side, a 50px margin to the left side, and a 100px margin to the left side. [View Result](#)

## HTML

```
<div id="myDiv"></div>
```

## CSS

```
#myDiv {  
    margin: 0 10px 50px 100px;  
    height: 40px;  
    width: 40px;  
    background-color: red;  
}
```

## Section 8.3: Margin property simplification

```
p {  
    margin:1px;                /* 1px margin in all directions */  
  
    /*equals to:*/  
  
    margin:1px 1px;  
  
    /*equals to:*/  
  
    margin:1px 1px 1px;  
  
    /*equals to:*/  
  
    margin:1px 1px 1px 1px;  
}
```

Another exapmle:

```
p{  
    margin:10px 15px;          /* 10px margin-top & bottom And 15px margin-right & left*/  
  
    /*equals to:*/  
  
    margin:10px 15px 10px 15px;  
  
    /*equals to:*/  
  
    margin:10px 15px 10px;  
    /* margin left will be calculated from the margin right value (=15px) */  
}
```

## Section 8.4: Horizontally center elements on a page using margin

As long as the element is a **block**, and it has an **explicitly set width value**, margins can be used to center block elements on a page horizontally.

We add a width value that is lower than the width of the window and the auto property of margin then distributes the remaining space to the left and the right:

```
#myDiv {  
  width:80%;  
  margin:0 auto;  
}
```

In the example above we use the shorthand margin declaration to first set 0 to the top and bottom margin values (although this could be any value) and then we use auto to let the browser allocate the space automatically to the left and right margin values.

In the example above, the #myDiv element is set to 80% width which leaves use 20% leftover. The browser distributes this value to the remaining sides so:

$$(100\% - 80\%) / 2 = 10\%$$

## Section 8.5: Example 1:

It is obvious to assume that the percentage value of margin to margin-left and margin-right would be relative to its parent element.

```
.parent {  
  width : 500px;  
  height: 300px;  
}  
  
.child {  
  width : 100px;  
  height: 100px;  
  margin-left: 10%; /* (parentWidth * 10/100) => 50px */  
}
```

But that is not the case, when comes to margin-top and margin-bottom. Both these properties, in percentages, aren't relative to the height of the parent container but to the **width** of the parent container.

So,

```
.parent {  
  width : 500px;  
  height: 300px;  
}  
  
.child {  
  width : 100px;  
  height: 100px;  
  margin-left: 10%; /* (parentWidth * 10/100) => 50px */  
  margin-top: 20%; /* (parentWidth * 20/100) => 100px */  
}
```

## Section 8.6: Negative margins

Margin is one of a few CSS properties that can be set to negative values. This property can be used to **overlap elements without absolute positioning**.

```
div{
```

```
display: inline;
}

#over{
margin-left: -20px;
}

<div>Base div</div>
<div id="over">Overlapping div</div>
```