

# Theory: Box-Shadow

🕒 12 minutes   0 / 5 problems solved

Skip this topic

Start practicing

219 users solved this topic. Latest completion was 4 days ago.

Have you ever looked at an `HTML` element you just made and thought: "I want to add some cool shadow to it but I don't know how"? If the answer is yes, then the property `box-shadow` is exactly what you are looking for. As you probably guessed by the name, this property allows you to add a shadow effect to your `HTML` element. In this topic, you will learn more about this useful property and its parameters.

## §1. Box-Shadow syntax

Here is what the syntax of this property generally looks like:

```
1 box-element {
2   box-shadow: offset-x offset-y blur-radius spread-radius color;
3 }
```

Let's take it apart and look closely at each of these parameters. We will start with two mandatory parameters:

- `offset-x` sets the position of the shadow on the horizontal axis relative to the element. A positive value shifts the shadow to the right, while a negative value shifts it to the left.
- `offset-y` sets the position of the shadow on the vertical axis relative to the element. A positive value shifts the shadow downwards, while a negative value shifts it upwards.

There are other parameters that are optional. Let's take a look at them:

- `blur-radius` gives the shadow a blurry effect. The larger this value, the bigger the blur, so the shadow becomes bigger and lighter. Negative values will not have any effect.
- `spread-radius` will make the shadow expand if its value is positive, or shrink if the value is negative. The default value is 0.
- `color` sets the color of the shadow. If no value is defined, it will be based upon the `currentcolor`. This parameter is also optional.

If both parameters `offset-x` and `offset-y` are 0, the shadow is placed behind the element. If `blur-radius` or `spread-radius` are set, this will generate a blur effect on our element.

This may seem really theoretical, so let's visualize this information using an example. Note: in this example, we also added a `background-color` to the element for better viewing; you do not need to worry about it right now.

Take a look:

```
1 <div id="box">I want a shadow</div>
```

```
1 #box {
2   background-color: darkred;
3   box-shadow: 5px 5px lightseagreen;
4 }
```

The result will look like this:

I want a shadow

If you want two or more shadows with different colors, this is also possible. All you need to do is separate each one with a comma:

Current topic:

[Box-Shadow](#) ...

Topic depends on:

- ✗ [CSS Selectors](#) Stage 2 ...
- ✗ [Colors](#) Stage 2 ...
- ✗ [Absolute units](#) Stage 2 ...
- ✗ [Relative units](#) Stage 2 ...

Table of contents:

[1 Box-Shadow](#)

[§1. Box-Shadow syntax](#)


[§2. Examples](#)

[§3. Conclusion](#)

[Feedback & Comments](#)

```
1  #box {  
2    background-color: darkred;  
3    box-shadow: 5px 5px lightseagreen, -5px -5px black;  
4  }
```

Now, the same element will have two shadows:



I want a shadow

## §2. Examples

Let's see some more specific examples of the property `box-shadow` and learn what we can achieve with it.

We can add a shadow so that the element seems like it's floating on the screen:

```
1  #box {  
2    box-shadow: 5px 5px 2px black  
3  }
```

This will be the result:

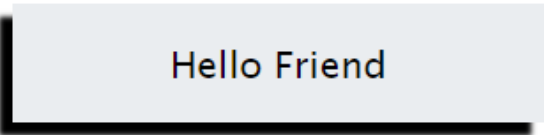


Hello Friend

Now let's change `offset-x` to a negative value:

```
1  #box {  
2    box-shadow: -5px 5px 2px black  
3  }
```

As a result, the shadow changes direction from right to left:



Hello Friend

Now let's experiment and make the value of `offset-y` negative:

```
1  #box {  
2    box-shadow: -5px -5px 2px black  
3  }
```


You can probably guess what the result will be:



Hello Friend

If we increase `blur-radius` and `spread-radius`, we will get something like this:

```
1  #box {  
2    box-shadow: 5px 0 8px 8px rgb(18, 16, 148)  
3  }
```



Hello Friend

We can also have one shadow behind another by setting `offset-x`, `offset-y` and `blur-radius` to 0.

```
1  #box {
2      box-shadow: 0 0 0 15px rgb(0, 0, 255),
3                  0 0 0 30px rgb(255, 0, 0);
4  }
```



We can make our element prettier if we set the parameters correctly. We can use different shadows and make a beautiful combination:

```
1  #box {
2      box-shadow: 0 -1px 1px rgba(0,0,0,0.2),
3                  0 0 0 3px rgb(255, 255, 255),
4                  0.2em 0.2em 1.5em rgba(0,0,0,0.3);
5  }
```




As you can see, our element has a slight shadow, making it more visually appealing without distracting us from the text.

### §3. Conclusion

With the help of the `box-shadow` property and its parameters, we can accentuate our `HTML` elements and make them look more elegant. With great power comes great responsibility, so use this property carefully as it can also make our elements look really bad.

 Report a typo

20 users liked this theory.  didn't like it. What about you?



Start practicing