

Theory: Positioning Properties

🕒

9 minutes

0 / 5

problems solved

Skip this topic

Start practicing

1138

users solved this topic. Latest completion was about 1 hour ago.

When working hard on creating a web page, it may be necessary to move a certain block a few pixels to the left, right, lower or higher. To do that, CSS offers intuitive and easy to remember rules for element placement. Let's consider which properties can help position elements and how to work with them.

§1. Property not applicable

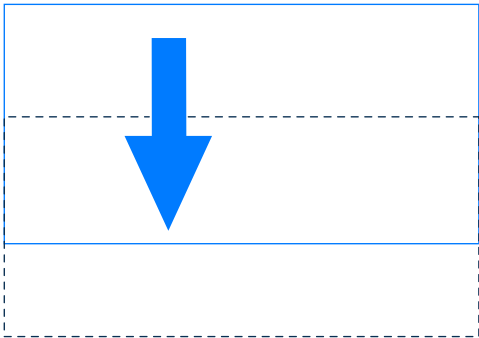
Before we begin to work with positioning, it's important to note one small but very important detail. The properties discussed in the next section will not work unless the `position` property is specified for the required elements. The default value of this property is `static`.

The value can be anything but `static`. Thus, we can move an element with `position: relative;` from its original location to the one we specify with the help of the studied properties. We won't consider `position` in detail, but feel free to explore it and [learn more](#) about its possible values.

To place an element over another or stack multiple layers, use the `z-index` property; we invite you [read more](#) about this property and try it as well.

§2. top

The `top` property determines how far an item should be moved from the top. It can take both positive and negative values in any unit of measurement, as well as a percentage (the same applies to other properties in the topic).



Consider the syntax:

```
1  div {
2    position: relative;
3    top: 10px;
4  }
```

In this example, the top edge of the `<div>` element is shifted down by `10` pixels.

There is an additional `position` property in the code. Without it, it wouldn't be possible to shift the element.

§3. bottom

The `bottom` property indicates the distance by which the lower edge of the element is shifted upwards.

Current topic:

[Positioning Properties](#)

Stage 2

Topic depends on:

✗ [Absolute units](#)

Stage 2

✗ [Relative units](#)

Stage 2

✗ [Position](#)

Stage 2

Topic is required for:

[z-index](#)

Stage 3

Table of contents:

[1 Positioning Properties](#)

[§1. Property not applicable](#)

[§2. top](#)

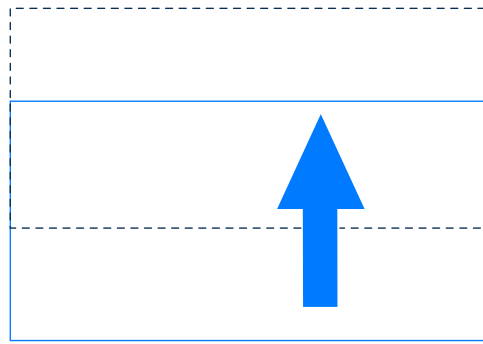
[§3. bottom](#)

[§4. left](#)

[§5. right](#)

[§6. Conclusion](#)

[Feedback & Comments](#)



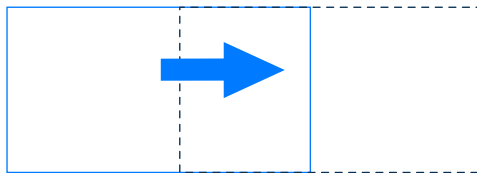
Look at a simple example:

```
1 p {  
2   position: relative;  
3   bottom: 5px;  
4 }
```

This moves up the bottom edge of `<p>` by `5` pixels.

§4. left

The `left` property indicates the distance by which the left edge moves to the right. Positive values move the element inside the block where it is located, while negative values move the element outside the block.



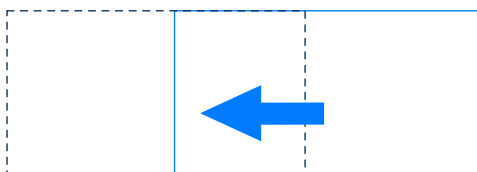
Here is an example:

```
1 div {  
2   position: relative;  
3   left: -25px;  
4 }
```

This moves the left edge of the `<div>` by `-25` pixels.

§5. right

The `right` property indicates the distance by which the right edge moves to the left.



For example:

```
1 div {  
2   position: relative;  
3   right: -2%;  
4 }
```

This code results in the `2` percent offset of the right edge of the `<div>` block.

§6. Conclusion

With the help of these properties, the arrangement of elements can be set very precisely and flexibly. Positioning is useful in creating complex interfaces such as galleries or pop-ups, and it surely comes handy in the layout of small decorative elements. Knowing how to manage the location of the elements, you can create unique pages and web applications that really stand out.

106 users liked this theory. 7 didn't like it. What about you?



Start practicing

[Comments \(9\)](#)[Hints \(0\)](#)[Useful links \(0\)](#)[Show discussion](#)