# Theory: z-index



7 minutes 0 / 4 problems solved



If we want to change the position of an `HTML` element on the screen, we can modify the X (horizontal) and Y (vertical) axes.
But what if we want to position an element closer to the screen?
That's when the `z-index` property comes in handy. This property works with the Z-axis which is responsible for the layer order of an `HTML` element. The element with a higher layer order will be placed closer to the screen and in front of the element with a lower layer order.

## §1. Syntax

The `z-index` property is defined by a positive or negative integer value, or by an `auto` value. An `auto` value gives the element the same layer order as its parent's.

- auto: Same layer order as its parent's. This is the default value.
- number: Sets the layer order of the element. Negative numbers can also be used.
- inherit: Inherits this property from its parent element.

```
/* Auto value */
z-index: auto;

/* Numbers */
z-index: 5;
z-index: 0;
z-index: -2;
```

The `z-index` property will only work on elements whose position has been defined as `absolute`, `fixed`, or `relative`.

## §2. Examples of property usage

To have a better understanding let's see some examples.

We will create three elements and define their position as `absolute`.

```
<body>
  <div class="container">
    <div class="box box1">1</div>
    <div class="box box2">2</div>
    <div class="box box3">3</div>
  </div>
</body>
```

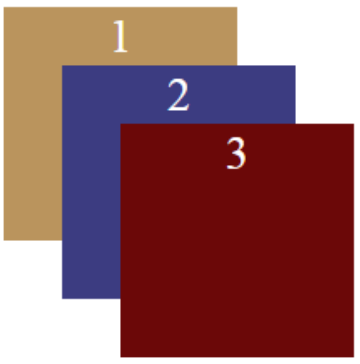
Current topic:
z-index Stage 3

Topic depends on:
CSS Selectors Stage 2
Colors Stage 2
Positioning Properties Stage 2

Table of contents:
↑ z-index
§1. Syntax
§2. Examples of property usage
§3. Conclusion
Feedback & Comments

```
.box {
  position: absolute;
  height: 80px;
  width: 80px;
}

.box1 {
  background-color: rgb(186, 148, 93);
}



.box2 {

  background-color: rgb(60, 60, 129);

  margin: 20px;

}



.box3 {

  background-color: rgb(107, 8, 8);

  margin: 40px;

}
```

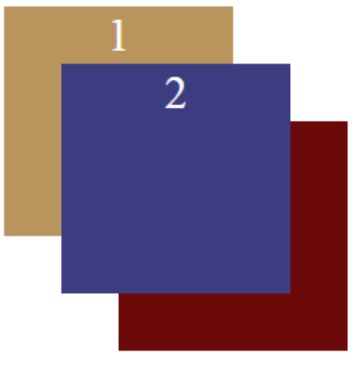
The result will be the following.




Here we can see that the elements are in front of each other by the order they were placed in the DOM. What if we want our second element to be in front of the other two? We can achieve this by setting the `z-index` property to 1.

```
.box2 {
  background-color: rgb(60, 60, 129);
  margin: 20px;
  z-index: 1;
}
```

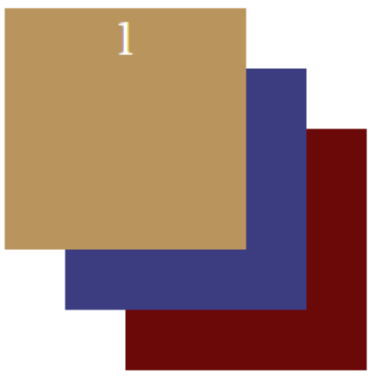
And this is the result we get:




Now if we want our `box1` element to be in front of the other two we need to set the property to 2, since we've set the second element to 1 our `box1` element needs a layer number greater than 1 to be in front of the other two.

```
.box1 {
  background-color: rgb(186, 148, 93);
  z-index: 2;
}
```

Let's see the result:




When defining the `z-index` property of the element with the value of 2 we change its position on the Z-axis, thus changing the natural order of the DOM.

## §3. Conclusion

We have just learned that besides changing the position of an element on the X and Y axis, we can also change its position on the Z-axis, thus placing one element on top of another. And don't forget that we need to define the `position` as `absolute`, `fixed`, or `relative` so that everything works as intended.

Report a typo

**21** users liked this theory. **0** didn't like it. **What about you?**

Start practicing

Comments (2) Hints (0) Useful links (0) Show discussion