



Difference between Positionings

To remember the difference between absolute, relative, and fixed positioning, we really need to focus on three key differences.

The first one is what happens to the content that is after the positioned element?

- For absolute and fixed position elements, content that comes after the element will ignore the positioned element and move up underneath it.
- For relative position in the content after the element will leave a space for the positioned element.

The second question is, do absolutely positioned elements have a width or do they shrink wrap?

- For absolute and fixed position elements, they are initially shrink-wrapped but can be sized or stretched.
- For relatively position elements, it depends on the nature of the element being used.
 - - If they're a block level element, they'll initially be stretched but they can be given size.
 - - If they're an inline element, they'll initially be shrink-wrapped and cannot be given size.

And the third question is, what are these elements positioned against?

- For absolutely positioned elements, they are positioned relative to the parent with positioning or viewport if no position parent is present.
- For relatively positioned elements, they are positioned relative to the element's original position.
- For fixed positioning, they are positioned relative to the viewport.

Now, let's have a quick look at z-index. The z-index property will only work on elements that have been explicitly set with positioning such as absolute, fixed or relative. The z-index property is used to define the stack order of an HTML element.

The **stack order** refers to the element's positioning on the Z axis, as opposed to the X or Y axis. A higher z-index value means the element will be closer to the top of the stacking order. z-index values can be declared using a positive, or a negative integer value. We can also use `auto`, `initial`, or `inherit` values.

- A positive z-index value will place the element higher in the stacking order.
- A negative z-index value will place the element below all positively stacked elements in the stack order.
- A value of `auto` sets the stack order equivalent to its parent.
- A value of `initial` sets the property to its default value.
- A value of `inherit` inherits its value from the parent element.

Exercise

Let's do an exercise on z-index. We are going to focus on elements test01, test02 and test03. In terms of stack order, test01 is setting out off the page, and then test02 is setting on top of it, and test03 is setting on top of that.

```
.test01 { z-index: 10; }
```

The test01 is now higher than the other two elements.

```
.test02 { z-index: -5; }
```

The element will now sit behind the other two elements.

```
.test03 { z-index: 20; }
```

test03 will now be on top of other elements.