MDN Web Docs
moz://a

Using z-index

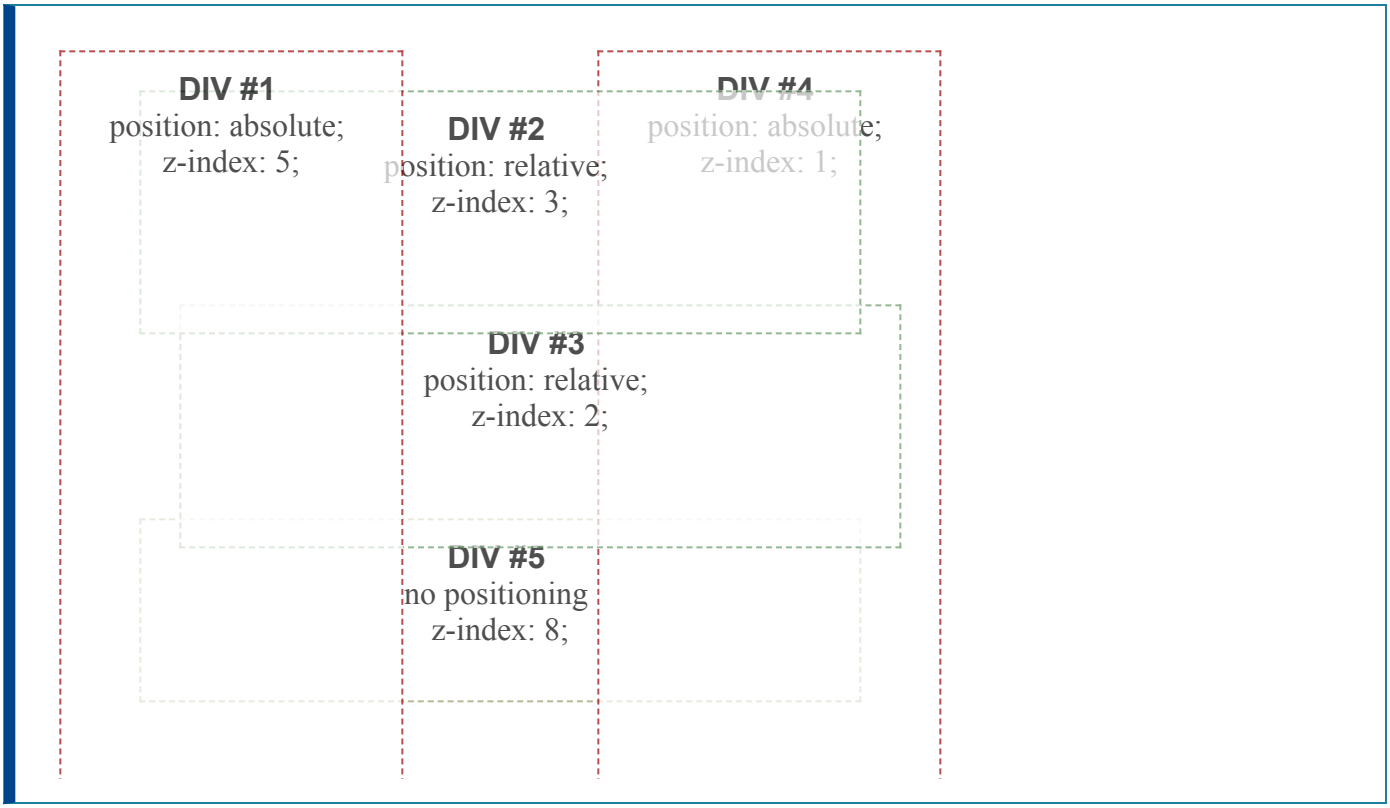
The first part of this article, Stacking without the z-index property, explains how stacking is arranged by default. If you want to create a custom stacking order, you can use the `z-index` property on a positioned element.

The `z-index` property can be specified with an integer value (positive, zero, or negative), which represents the position of the element along the z-axis. If you are not familiar with the z-axis, imagine the page as a stack of layers, each one having a number. Layers are rendered in numerical order, with larger numbers above smaller numbers.

- bottom layer *(farthest from the observer)*
- ...
- Layer -3
- Layer -2
- Layer -1
- Layer 0 *(default rendering layer)*
- Layer 1
- Layer 2
- Layer 3
- ...
- top layer *(closest to the observer)*

> **Notes:**
>
> - When no `z-index` property is specified, elements are rendered on the default rendering layer 0 (zero).
> - If several elements share the same `z-index` value (i.e., they are placed on the same layer), stacking rules explained in the section Stacking without the z-index property apply.

In the following example, the layers' stacking order is rearranged using `z-index`. The `z-index` of element #5 has no effect since it is not a positioned element.



Source code for the example

HTML

```
<div id="abs1">
  <b>DIV #1</b>
  <br />position: absolute;
  <br />z-index: 5;
</div>

<div id="rel1">
  <b>DIV #2</b>
  <br />position: relative;
  <br />z-index: 3;
</div>

<div id="rel2">
  <b>DIV #3</b>
```

```
  <br />position: relative;
  <br />z-index: 2;
</div>

<div id="abs2">
  <b>DIV #4</b>
  <br />position: absolute;
  <br />z-index: 1;
</div>

<div id="sta1">
  <b>DIV #5</b>
  <br />no positioning
  <br />z-index: 8;
</div>
```

CSS

```
div {
  padding: 10px;
  opacity: 0.7;
  text-align: center;
}

b {
  font-family: sans-serif;
}

#abs1 {
  z-index: 5;
  position: absolute;
  width: 150px;
  height: 350px;
  top: 10px;
  left: 10px;
  border: 1px dashed #900;
  background-color: #fdd;
}

#rel1 {
  z-index: 3;
  height: 100px;
  position: relative;
```

```
  position: relative;
  top: 30px;
  border: 1px dashed #696;
  background-color: #cfc;
  margin: 0px 50px 0px 50px;

}

#rel2 {
  z-index: 2;
  height: 100px;
  position: relative;
  top: 15px;
  left: 20px;
  border: 1px dashed #696;
  background-color: #cfc;
  margin: 0px 50px 0px 50px;
}

#abs2 {
  z-index: 1;
  position: absolute;
  width: 150px;
  height: 350px;
  top: 10px;
  right: 10px;
  border: 1px dashed #900;
  background-color: #fdd;
}

#sta1 {
  z-index: 8;
  height: 70px;
  border: 1px dashed #996;
  background-color: #ffc;
  margin: 0px 50px 0px 50px;
}
```

See also

- Stacking without the z-index property: The stacking rules that apply when `z-index` is not used.
- Stacking with floated blocks: How floating elements are handled with stacking.