

POSITION

HOME / CSS ALMANAC / PROPERTIES / P /

position



Chris Coyier on Mar 19, 2012 (Updated on Feb 22, 2021)

PROPERTIES

- > A
- > B
- > C
- > D
- > E
- > F
- > G
- > H
- > I
- > J
- > L
- > M
- > O
- > P
- > Q
- > R
- > S
- > T
- > U
- > V
- > W
- > Z

SELECTORS

- > A
- > B
- > C
- > D
- > E

The position property can help you manipulate the location of an element, for example:

```
.element {  
  position: relative;  
  top: 20px;  
}
```

Relative to its original position the element above will now be nudged down from the top by 20px. If we were to animate these properties we can see just how much control this gives us (although this isn't a good idea for performance reasons):

HTML SCSS Result EDIT ON CODEPEN

```
.parent {  
  border: 2px solid #0074d9;  
  color: #0074d9;  
  padding: 20px;  
}  
  
.element {  
  border: 1px dotted #000;  
  background-color: #eee;  
  padding: 20px;  
  color: #000;  
  position: relative;  
  margin-top: 10px;  
  animation: push ease 5s alternate infinite;  
  animation-delay: 1.5s;  
}
```

> F
> G
> H
> I
> L
> M
> N
> O
> P
> R
> S
> T
> U
> V
> W

```
@keyframes push {  
  0% {  
    left: 0;  
    top: 0;  
  }  
}
```

Resources View Compiled

relative is only one of six values for the position property. Here are the others:

🔗 Values

- **static:** every element has a static position by default, so the element will stick to the normal page flow. So if there is a [left/right/top/bottom/z-index](#) set then there will be no effect on that element.
- **relative:** an element's original position remains in the flow of the document, just like the static value. But now [left/right/top/bottom/z-index](#) will work. The positional properties “nudge” the element from the original position in that direction.
- **absolute:** the element is removed from the flow of the document and other elements will behave as if it's not even there whilst all the other positional properties will work on it.
- **fixed:** the element is removed from the flow of the document like absolutely positioned elements. In fact they behave almost the same, only fixed positioned elements are always relative to the document, not any particular parent, and are unaffected by scrolling.
- **sticky (experimental):** the element is treated like a relative value until the scroll location of the viewport reaches a specified threshold, at which point the element takes a fixed position where it is told to stick.
- **inherit:** the position value doesn't cascade, so this can be used to specifically force it to, and inherit the positioning value from its parent.

🔗 Absolute

If a child element has an absolute value then the parent element will behave as if the child isn't there at all:

```
.element {  
  position: absolute;  
}
```

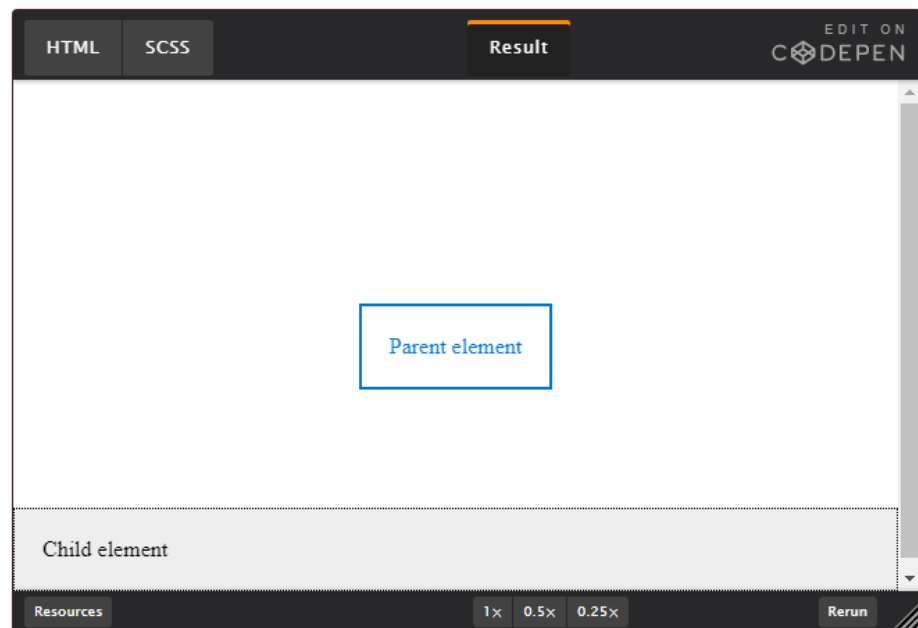
HTML SCSS Result EDIT ON CODEPEN

```
border: 1px dotted #000;  
background-color: #eee;  
padding: 20px;  
color: #000;  
position: absolute;  
}  
  
/* Alignment styles */  
  
body, html {  
  height: 100%;  
}  
  
body {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

Resources View Compiled

And when we try to set other values such as `left`, `bottom`, and `right` we'll find that the child element is responding not to the dimensions of its parent, but the document:

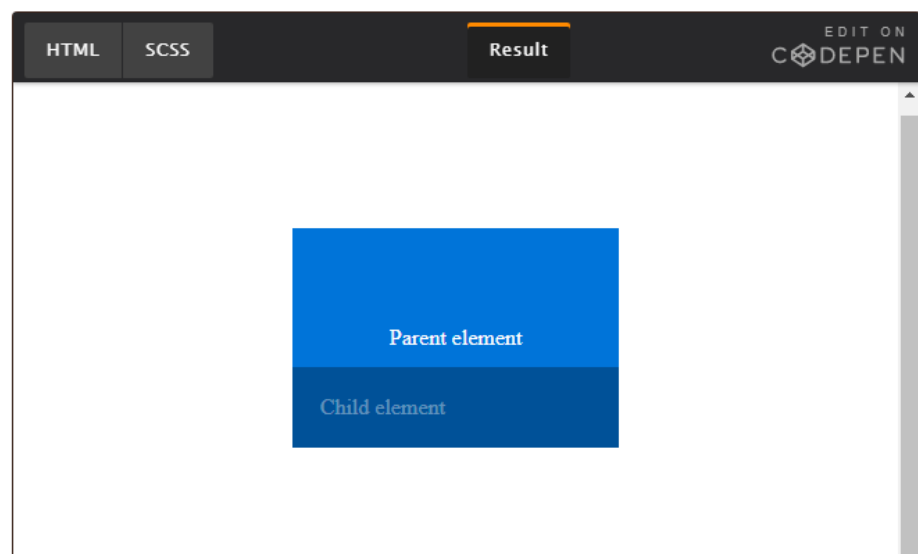
```
.element {  
  position: absolute;  
  left: 0;  
  right: 0;  
  bottom: 0;  
}
```



To make the child element positioned absolutely from its parent element we need to set this on the parent element itself:

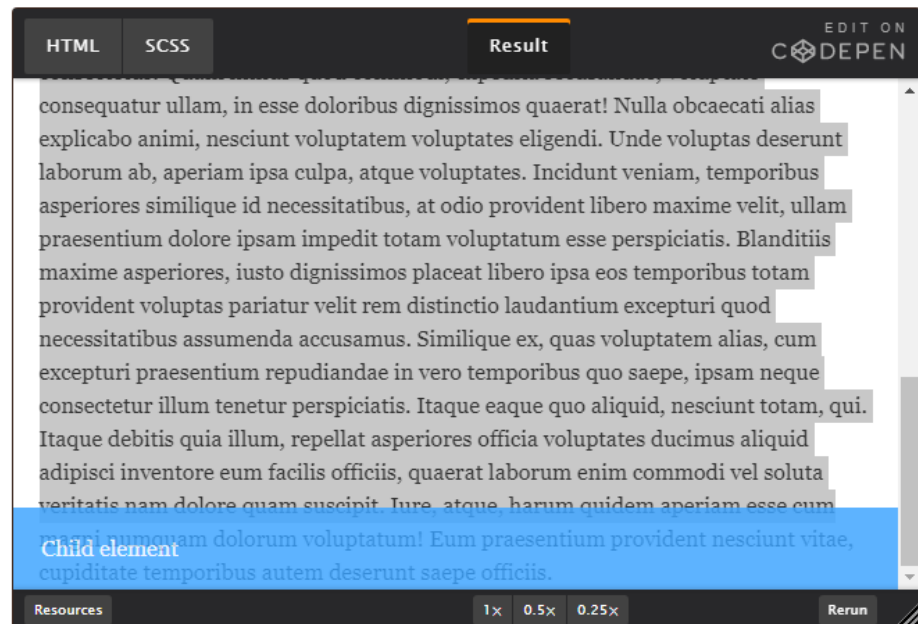
```
.parent {  
  position: relative;  
}
```

Now properties such as left, right, bottom and top will refer to the parent element, so that if we make the child element transparent we can see it sitting right at the bottom of the parent:








Fixed

The fixed value is similar to absolute as it can help you position an element anywhere relative to the document, however this value is unaffected by scrolling. See the child element in the demo below and how, once you scroll, it continues to stick to the bottom of the page:



This browser support data is from [CanIuse](#), which has more detail. A number indicates that browser supports the feature at that version and up.

Desktop

				
Chrome	Firefox	IE	Edge	Safari
4	2	7	12	3.1

Mobile / Tablet

			
Android Chrome	Android Firefox	Android	iOS Safari

94

92

3

8

Sticky

The sticky value is like a compromise between the *relative* and *fixed* values. As of this writing, it is currently an experimental value, meaning it is not part of the official spec and only partially adopted by select browsers. In other words, it's probably not the best idea to use this on a live production website.

What does it do? Well, it allows you to position an element relative to anything on the document and then, once a user has scrolled past a certain point in the viewport, fix the position of the element to that location so it remains persistently displayed like an element with a *fixed* value.

Take the following example:

```
.element {  
  position: sticky; top: 50px;  
}
```

The element will be relatively positioned until the scroll location of the viewport reaches a point where the element will be 50px from the top of the viewport. At that point, the element becomes sticky and remains at a fixed position 50px top of the screen.

The following demo illustrates that point, where the top navigation is default relative positioning and the second navigation is set to sticky at the very top of the viewport. Please note that the demo will only work in Chrome, Safari and Opera at the time of this writing.

