

CSS Advanced

Done

Table Of Contents

- [What you will learn](#)
- [Useful Resources](#)
 - [Elements position](#)
 - [Dealing with overflow](#)
 - [CSS Pseudo Class Selectors](#)
 - [hover](#)
 - [children](#)
 - [not](#)
- [Feedback](#)

Last updated : April, 2nd 2021

What You Will Learn

- Positions
- Overflow
- Pseudo Class Selector

Useful Resources

- [Learn Layout](#)

Elements Position

The `position` property specifies the type of positioning method used for an element (static on the page, relative to his parent, etc..)

It can be:

- `static`: Default value. Elements render in order, as they appear in the document flow
- `relative`: The element is positioned relative to its normal position; you can add `left`, `right`, `top` or `bottom` arguments to add a gap by the element.
- `fixed`: Always stays in the same place even if the page is scrolled, `left`, `right`, `top` and `bottom` are used to position the element
- `absolute`: The element is positioned relative to its first positioned (not static) ancestor element
- `sticky`: The element is positioned based on the user's scroll position

[More on CSS Position](#)

[CSS Position Tricks](#)

Exercise 1

On the same HTML file (as the one in the CSS lesson)



Recreate this image with the **position** property

The elements are : a parent box (grey), a blue box and an orange box



Dealing With Overflow

The `overflow` property specifies whether to clip the content or add scrollbars when an element's content is too big to fit in the specified area.

It can be:

- `visible` - Default. The overflow is not clipped. The content renders outside the element's box
- `hidden` - The overflow is clipped, and the rest of the content will be invisible
- `scroll` - The overflow is clipped, and a scrollbar is added to see the rest of the content
- `auto` - Similar to `scroll`, but it adds scrollbars only when necessary

[Look at the demonstration Here](#)

CSS Pseudo Class Selectors

A pseudo-class is used to style an element when he is in a special state (for example, when the user hovers over it)

The syntax is : `element:pseudo-class {...}`



Hover

`hover` means “mouse is over it”, for example, to change the color of a `<div>` when the mouse goes over it:

```
div:hover {
  background-color: blue;
}
```

Children

You can target elements only if they are the *n*th child of their parent. For example, to style `<p>` tags that are the first child of another element, use:

```
p:first-child{
  color: blue;
}
```

The same thing works with `last-child` or `nth-child(n)`

Not

The `not` pseudo-class allows you to exclude elements; for example, select all the `divs` that are not of the `ignore` class.

```
div:not(.ignore){
  background-color: yellow;
}
```

For more, see [this](#).