

Position Types

Hello, and welcome to the lesson about the CSS position types. The CSS position property defines, as the name says, how the element is positioned on the web page and the top right bottom and left properties, determine the final location of positioned elements.

The z-index property specifies, the stack order of positioned element and there are five main values a position property can have, static, this is the default value. All HTML elements are positioned static by default. It simply means that elements will be positioned in order as they appear in the document, elements with the static position won't be affected by the top bottom left and right properties. Now when we use the term positioned element, we refer to to an element whose position value is either relative, absolute, fixed or sticky. In other words, it's anything except static. And I meant with position relative is positioned relative to its normal position.

Setting the top right bottom and left properties of relatively positioned element will cause it to be adjusted away from its normal position and other content. Will not be adjusted to fit into any gap left by the element and element with position fixed is positioned relative to the viewport, which means it always stays in the same place. Even if the page is scrolled, the top right bottom and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located. Now element with position: absolute is positioned relative to the nearest position ancestor instead of

position, relative to the viewport like fixed. However, if an absolute positioned element has no position ancestors, it uses the document body and moves along with page scrolling. An element with position sticky is positioned based on the users scroll position.

Sticky element, toggles between relative and fixed depending on the scroll position. It is positioned relative until a given offset position is met in the viewport, then it sticks in place like position: fixed. Now, I know this may all sound confusing so let's look at some examples. This is my HTML structure. I have three divs here and a few paragraphs. It's so that I can scroll my preview here. Now, inside the stylesheet, I defined the - oh and if we look here each div has a class of box and I have additional classes: first, second and third. Okay? So in the style sheet, all the boxes are have set width and height and some styling for the border and text alignment. And now I change different background colors for each of the classes. Each of these elements has positioned static by default. So if I want the second to be moved, if I add left 20 pixels, nothing happens. That's because the properties top bottom left and right will not take any effect on element that has a static position. But as soon as I add a position, relative. We can see that the property left has moved the element relative to its original position, removing any top left bottom or right values and just adding position:relative does nothing to an element doesn't change its appearance. But if I add left 20 and let's say Top 30 pixels. This will move the element relative to its original position and as you can see the gap that were the element was originally located is still there. None of the other elements position hasn't been changed. We can also do negative values. For example, top -30 would move it to the top. Now, what if I say, set the position to fixed, let's see how that behaves. So the fixed position, you can see that it moved, other content, the Gap where the element was originally located is no longer

here and if I scroll the page it stays fixed, always visible. But it's, it's positioned relative to Viewport. So top negative value moves it outside of the viewport. If I define top 30 pixels, it will be fixed 30 pixels from the top and 20 pixels from the left. Now position sticky. And I need to know that it does not work in Internet Explorer if for whatever reason you need to support it. So position sticky behaves as relative at the beginning but as soon as users scroll to the position it would have if that would be fixed as soon as it reaches that place, it becomes as it would have been fixed. So right now it's kind of relative. Now, it is fixed. That's how the position sticky works and final position value absolute. And as you can see, setting position: absolute moves element outside of the location it would have been and doesn't leave the gap. So other elements are being moved as if the element is not even here. And right now, the left and top values, you see that it's positioned relative to the viewport, but that is only because it doesn't have any positioned parents. So let's try moving the second box inside the third. Okay?

So let's go back here and I will move that box inside this third div. And now, if my third div will have position: relative... That second div will be positioned relatively to its parent. So let's maybe make this one a bit bigger. Instead of top that would be bottom, you see its bottom zero. That's relative to the parent container. As long as the parent has position: relative. If it doesn't, that's relative to the viewport. So it will look for the first parent container with position: relative and it will be positioned absolutely relative to that container that parent element. If we add a relative back here, the absolute the all the top left, right left and bottom values will be relative to the parent container. So it doesn't stay fixed or visible once you scroll. So that would be the difference between different position types.

Now, another property that is important to mention here is the z-index property because let's try actually overlapping that element to the top. Let's do - 40. Okay. And now you see that this second box is on top of the first one. And as you probably already know Z-index specifies, the Z axis opposed to X and Y. It defines how close the element is towards us. So by default the order in the HTML, document would define the order of which element is on top of another. But if we specify the z-index property to this first one, let's say four or five. It doesn't do anything. The z-index only works on positioned elements. So I do need to add `position:relative`. And right now, my first div has a higher Z-index because default is one kind of. So it has higher that index value than this third div, the second one, sorry the blue one, but if I change the z-index here to even six, okay? That is on top of that one, but there is a very important aspect here that the position of an element will be relative to the current position stack. What I mean is if a third div has a z-index of three I can set 99999 here and it won't move it on top of the z-index five here that is because the z-index value can be only as high as the z-index value that is defined for a parent container. That is very, very important. And I know that it is confusing and in Divi, a column, for example, can have a different Z index property and then moving your elements outside overlapping them can cause different issues, but as long as you remember, that the z-index can only be as high as the parent Z-index value. And you remember that the Z-index can only be applied along with the position other than static, then you should be able to troubleshoot and position your website elements easier. So hopefully that is helpful.