

CSS Layout

There are different ways to layout your web page using CSS. Page layout is the process of placing and arranging the text and image content on the page to produce a visually pleasing page. We need to place all the elements in the proper place on the page so it will look like our wireframe. So, what's the best way to do that?

Up until now, our elements have been in 'Normal Flow', meaning each element is displayed one after another based on how we placed them in the HTML. Some might take more or less room, but they are all in order according to the HTML. This is normal flow.

What are some ways to take blocks out of normal flow?

There are options like float that will shift a box to the right or left and allow content to display wrapped around it. `*.item {float: left;}` The only time I use floats is if I have a small image, button, or logo that I want text to wrap around. Keep in mind that float should **not** be used to layout an entire page. I rarely use float.

The position property can also be used to take elements out of their normal flow. All elements in normal flow have a position property value of static. Meaning, in normal flow, all items display one after the other and if you scroll, they scroll with the page.

Position: relative changes this flow, it would reference a certain point from where the element would normally be in normal flow. `*.item { position: relative; bottom: 50px;}`

Position: absolute would be removed completely from normal flow and positioned relative to its parent element (if you give the parent position: relative), or the viewport of the page itself. It usually will leave normal flow and go to the top left of the page. Then you use offset values to move it where you want it. `*.item { position: absolute; top: 20px; right: 30px;}`

Both position: relative and position: absolute should be used very sparingly. I have only used these properties when I have a very specific reasons to do so. But again, not to layout the entire page. They can be very challenging to use, especially for beginning web learners. And if overused they can quickly become a nightmare.

Position: fixed will fix an element in place so that it does not scroll with the page. This is popular for navigations or background images to remain in place as users scroll. But again, it would not address the needs of laying out the entire page.

Flex-box is another layout method if you want your items to be in a row or column. It uses the display property with a value of flex. Then children of that element will lay out as rows or columns. What is great about flex-box is how responsive it can make your page. It is great for things like product galleries or photo galleries where the rows can start out as 4 across and as the device gets smaller to goes to 3 then 2 then 1 on a phone. It makes it really easy to have

beautifully responsive areas on the page when there are a number of items grouped together. I also love flex-box for the horizontal links in the navigation or menu. It does a great job of spacing the links to give them a nice look especially if one link is longer than the rest. But again, it's not the best for laying out an entire page. It works great on certain content, but not the whole page.

Grid is the method we will use in this course to layout our page. It is a lot like flex-box in many ways but where flex-box is one-dimensional (a row or a column), Grid is two dimensional (rows and columns at the same time). Grid makes it much easier to layout a page without having to ever use float or positioning. Grid allows you to move blocks around on the page regardless of the order it is in HTML. You could still also use flex-box for certain parts of the page. Grid and flex-box can be used together.

When you start taking items out of the natural flow, sometimes they can overlap on top of each other. So far, you've seen top, bottom, right, left that have to do with a 2 dimensional x and y axis but there is the z-axis that is 3 dimensional and has to do with which element will be on top of overlapping elements, or the stacking order. Perhaps you've used word processors or other programs that allow you to bring items to the front or send them to the back. It's the same idea. It's called z-index. Z-index is the property name and the value you give it is a number. The bigger the number the more toward the front that element will be in the stacking order.