

HTML Refresher: Block-level vs Inline Elements

It's not really a CSS topic, though it's related to it: The difference between **block-level** and **inline elements**.

You can read a more detailed article (which also includes a YouTube video about HTML at the top of the page) here: <https://academind.com/learn/html/beginner-s-guide/diving-deeper-into-html#block-level-vs-inline-elements>

Here's the executive summary:

Block-level elements are rendered as a block and hence take up all the available horizontal space. You can set margin-top and margin-bottom and two block-level elements will render in two different lines.

Some examples are: `<div>` , `<section>` , `<article>` , `<nav>` but also `<h1>` , `<h2>` etc and `<p>` .

Inline elements on the other hand only take up the space they require to fit their content in. Hence two inline-elements will fit into the same line (as long as the combined content doesn't take up the entire space in which case a line break would be added).

They also use the box-model you learned about but `margin-top` and `margin-bottom` have no effect on the element. `padding-top` and `padding-bottom` also have a different effect. They don't push the adjacent content away but they will do so with the element border. You can read more about that behavior in the following article: <https://hacks.mozilla.org/2015/03/understanding-inline-box-model/>

Additionally, setting a `width` or `height` on an inline element also has no effect. The width and height is auto to take as much space as required by the content.

Logically, this makes sense since you don't want your inline elements to destroy your multi-line text-layout. If you want to do so or need both block-level and inline behavior, you can set `display: inline-block` to merge behaviors.

Some example elements are: `<a>` , `` , ``