Block and Inline

Foundations Course

Introduction

In the previous lesson, we discovered that different display types have unique box models, and we can modify the box calculation by changing the `display` property. CSS has two box types: `block` and `inline` boxes, which determine element behavior and interaction. The `display` property controls how HTML elements appear on the webpage. We will explore its various options further in this lesson.

Lesson overview

This section contains a general overview of topics that you will learn in this lesson.

- You'll learn about "Normal flow".
- You'll learn the difference between `block` and `inline` elements.
- You'll learn which elements default to `block` and which elements default to `inline`.
- You'll learn what divs and spans are.

Block vs inline

Most of the elements that you have learned about so far are block elements. In other words, their default style is `display: block`. By default, block elements will appear on the page stacked atop each other, each new element starting on a new line.

Inline elements, however, do not start on a new line. They appear in line with whatever elements they are placed beside. A clear example of an inline element is a link, or `<a>` tag. If you stick one of these in the middle of a paragraph of text, the link will behave like a part of the paragraph. Additionally, padding and margin behave differently on inline elements. In general, you do not want to try to put extra padding or margin on inline elements.

The middle ground inline-block

Inline-block elements behave like inline elements, but with block-style padding and margin. `display: inline-block` is a useful tool to know about, but in practice, you'll probably end up reaching for flexbox more often if you're trying to line up a bunch of boxes. Flexbox will be covered in-depth in the next lesson.

Divs and spans

We can't talk about block and inline elements without discussing divs and spans. All the other HTML elements we have encountered so far give meaning to their content. For example, paragraph elements tell the browser to display the text it contains as a paragraph. Strong elements tell the browser which texts within are important and so on. Yet, divs and spans give no particular meaning to their content. They are just generic boxes that can contain anything.

Having elements like this available to us is a lot more useful than it may first appear. We will often need elements that serve no other purpose than to be "hook" elements. We can give an id or class to target them for styling with CSS. Another use case we will face regularly is grouping related elements under one parent element to correctly position them on the page. Divs and spans provide us with the ability to do this.

Div is a block-level element by default. It is commonly used as a container element to group other elements. Divs allow us to *divide* the page into different blocks and apply styling to those blocks.

Span is an inline-level element by default. It can be used to group text content and inline HTML elements for styling and should only be used when no other semantic HTML element is appropriate.

Assignment

1. The concept of "Normal flow" is implied in the box-model resources, but isn't laid out very specifically. Read "Normal Flow" from MDN to make sure you understand how elements lay themselves out by default.
2. W3 schools' "HTML Block and Inline Elements" has a description and a list of all the default block and inline elements.
3. The Digital Ocean tutorial "Inline vs Inline-block Display in CSS" has a couple of great examples that clarify the difference between `inline` and `inline-block`.
4. Go to our CSS exercises repository and navigate to the `margin-and-padding` directory. Review each README file prior to completing the following exercises in order:
   - `01-margin-and-padding-1`
   - `02-margin-and-padding-2`

   Note: Solutions for these exercises can be found in the `solution` folder of each exercise.
5. Apply what you learned about the box model to improve the look of your Recipe page's `index.html` homepage. Currently, it's just a plain list, so get creative with layouts, colors, and styles to make your page uniquely captivating.

Knowledge check

The following questions are an opportunity to reflect on key topics in this lesson. If you can't answer a question, click on it to review the material, but keep in mind you are not expected to memorize or master this knowledge.

- What is the difference between a block element and an inline element?
- What is the difference between an inline element and an inline-block element?
- Is an `h1` block or inline?
- Is `button` block or inline?
- Is `div` block or inline?
- Is `span` block or inline?

Additional resources

This section contains helpful links to related content. It isn't required, so consider it supplemental.

- Learn CSS Layout is a tutorial that is a little dated at this point, but its examples are clear. The first 6 slides cover the material we've seen so far.
- Watch this short video on what the term "Normal Flow" means in CSS.
- For a more interactive explanation and example, try this Scrim on block and inline display.