Community





Block and Inline

Foundations Course

Introduction

models, and we can modify the box calculation by changing the box-sizing 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.

In the previous lesson, we discovered that different display types have unique box

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

Lesson overview

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
- You'll learn what divs and spans are.
- **Block vs inline**

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.

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

Most of the elements that you have learned about so far are block elements. In other

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.

generic boxes that can contain anything.

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

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

you understand how elements lay themselves out by default. 2. W3 schools' "HTML Block and Inline Elements" has a description and a list

Assignment

HTML element is appropriate.

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. Do the exercises in our <u>CSS exercises repository's</u> <u>foundations/block-and-</u>

of all the default block and inline elements.

02-margin-and-padding-2

to show off on your resume.

expected to memorize or master this knowledge.

adding some CSS to it!

1. The concept of "Normal flow" is implied in the box-model resources, but

isn't laid out very specifically. Read <u>"Normal Flow" from MDN</u> to make sure

order: 01-margin-and-padding-1

inline directory (remember that the instructions are in the README) in the

Note: Solutions for these exercises can be found in the solution folder of each exercise.

Foundations section? Well, it's rather *plain* looking, isn't it? Let's fix that by

How you actually style it is completely open, but you should use the

5. Remember the Recipe page you created as practice from the HTML

- external CSS method (for this practice and moving forward). You should also try to use several of the properties mentioned in the previous lesson (color, background color, typography properties, etc). Take some time to play around with the various properties to get a feel for what they do. For now, don't worry at all about making it look *good*. This
 - We haven't covered how to use a custom font for the font-family property yet, so for now take a look at <u>CSS Fonts</u> for a list of generic font families to use, and CSS Web Safe Fonts for a list of fonts that are web safe. Web safe means that these are fonts that are installed on basically every computer or device (but be sure to still include a generic font family as a fallback).

is just to practice and get used to writing CSS, not to make something

What is the difference between an inline element and an inline-block element? <u>ls an h1 block or inline?</u>

Additional resources

Knowledge check

Is button block or inline? ls div block or inline? Is span block or inline?

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

What is the difference between a block element and an inline element?

supplemental. Learn CSS Layout is a tutorial that is a little dated at this point, but its examples are

For a more interactive explanation and example, try this **Scrim on block and inline**

Watch this short video on what the term "Normal Flow" means in CSS.

clear. The first 6 slides cover the material we've seen so far.

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

<u>display</u>.

Improve on GitHub Report an issue

Mark Complete

Support us! The Odin Project is funded by the community. Join us in empowering

High quality coding education maintained by an open source community.

Support

learners around the globe by supporting The Odin Project! Donate now \rightarrow Learn more

See lesson changelog

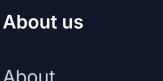
Legal

→ Next Lesson

Guides

THE ODIN PROJECT

View Course







About	FAQ	Community guides	Terms
Team	Contribute	Installation guides	Privacy
Blog	Contact us		
Success Stories			