mdn web docs

Normal Flow

This article explains normal flow, or the way that webpage elements lay themselves out if you haven't changed their layout.

Prerequisites:	The basics of HTML (study Introduction to HTML), and an idea of How CSS works (study Introduction to CSS.)
Objective:	To explain how browsers layout web pages by default, before we begin to make changes.

As detailed in the last lesson introducing layout, elements on a webpage lay out in normal flow if you haven't applied any CSS to change the way they behave. And, as we began to discover, you can change how elements behave either by adjusting their position in normal flow or by removing them from it altogether. Starting with a solid, well-structured document that's readable in normal flow is the best way to begin any webpage. It ensures that your content is readable even if the user's using a very limited browser or a device such as a screen reader that reads out the content of the page. In addition, since normal flow is designed to make a readable document, by starting in this way you're working *with* the document rather than struggling *against* it as you make changes to the layout.

Before digging deeper into different layout methods, it's worth revisiting some of the things you have studied in previous modules with regard to normal document flow.

How are elements laid out by default?

The process begins as the boxes of individual elements are laid out in such a way that any padding, border, or margin they happen to have is added to their content. This is what we call the **box model**.

By default, a block-level element's content fills the available inline space of the parent element containing it, growing along the block dimension to accommodate its content. The size of inline-level elements is just the size of their content. You can set `width` or `height` on some elements that have a default `display` property value of `inline`, like `<img>`, but `display` value will still remain `inline`.

If you want to control the `display` property of an inline-level element in this manner, use CSS to set it to behave like a block-level element (e.g., with `display: block;` or `display: inline-block;`, which mixes characteristics from both).

That explains how elements are structured individually, but how about the way they're structured when they interact with one another? The normal layout flow (mentioned in the layout introduction article) is the system by which elements are placed inside the browser's viewport. By default, block-level elements are laid out in the *block flow direction*, which is based on the parent's writing mode (*initial*: horizontal-tb). Each element will appear on a new line below the last one, with each one separated by whatever margin that's been specified. In English, for example, (or any other horizontal, top to bottom writing mode) block-level elements are laid out vertically.

Inline elements behave differently. They don't appear on new lines; instead, they all sit on the same line along with any adjacent (or wrapped) text content as long as there is space for them to do so inside the width of the parent block level element. If there isn't space, then the overflowing content will move down to a new line.

If two vertically adjacent elements both have a margin set on them and their margins touch, the larger of the two margins remains and the smaller one disappears. This is known as **margin collapsing**. Collapsing margins is only relevant in the **vertical direction**.

Let's look at a simple example that explains all of this:

HTML Play

```
<h1>Basic document flow</h1>
```

```
<p>
  I am a basic block level element. My adjacent block level elements sit on new
  lines below me.
</p>

<p>
  By default we span 100% of the width of our parent element, and we are as tall
  as our child content. Our total width and height is our content + padding +
  border width/height.
</p>

<p>
  We are separated by our margins. Because of margin collapsing, we are
  separated by the width of one of our margins, not both.
</p>

<p>
  Inline elements <span>like this one</span> and <span>this one</span> sit on
  the same line along with adjacent text nodes, if there is space on the same
  line. Overflowing inline elements will
  <span>wrap onto a new line if possible (like this one containing text)</span>,
  or just go on to a new line if not, much like this image will do:
  <img src="long.jpg" alt="snippet of cloth" />
</p>
```

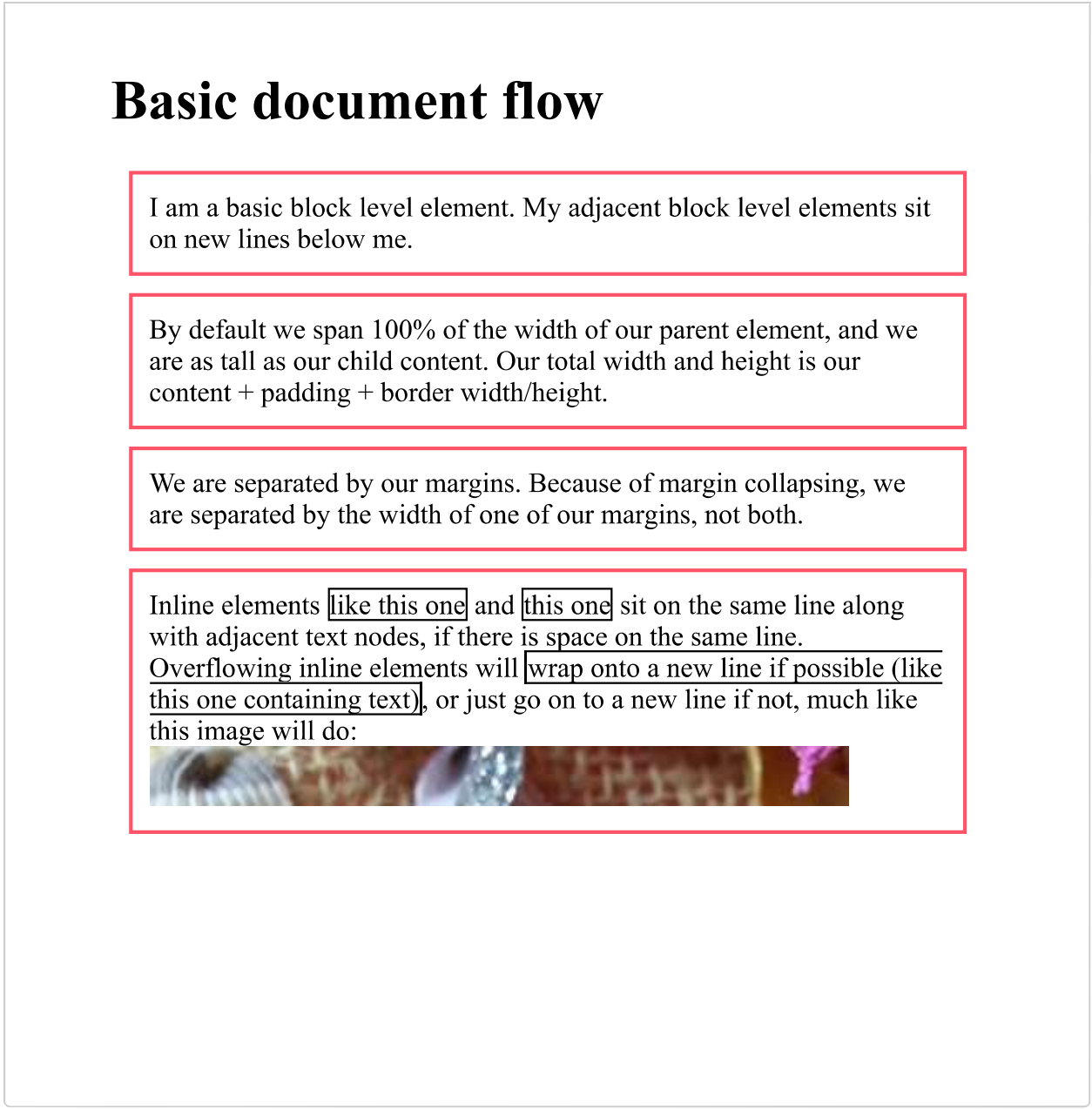
CSS Play

```
body {
  width: 500px;
  margin: 0 auto;
}

p {
  background: rgb(255 84 104 / 30%);
  border: 2px solid rgb(255 84 104);
  padding: 10px;
  margin: 10px;
}

span {
  background: white;
  border: 1px solid black;
}
```

Play



Summary

In this lesson you've learned the basics of normal flow — the default layout for CSS elements. By understanding how inline elements, block elements, and margins behave by default, it'll be easier to modify their behavior in the future.

In the next article, we'll build on this knowledge by making changes to CSS elements using flexbox.

Help improve MDN

Was this page helpful to you?

Yes No

Learn how to contribute.

This page was last modified on Nov 28, 2023 by MDN contributors.