## COMP 1017

Day 15
Introduction to Flexbox

websites with one big long block of content.

So far, we've been designing

... but a lot of websites have more than one column of stuff.

So, how can we get our boxes to

render side by side?

## Introducing Flexbox

weird flex, but ok

## Model (Flexbox) is a way to order, align, and lay out website content.

The CSS Flexible Box Layout

While it became an official W3C recommendation in 2012, we've been waiting for some browsers to reach their end of life before using this as the go-to layout technique.



Flexbox is our new go-to and, for the purpose of this course, we won't worry about backwards compatibility or using other techniques.

# column layouts?

... so, how can we create multiple

# Block-Level & Inline Elements

listen, blockhead!

## display property?

What is an element's

and lay it out on the page by default.

It is how the browser will render it

### Let's take a look at two types: block-level elements and inline elements.

By default, many HTML elements will render as a block — that is, it will take up the entire width of its container.

# By default, block-level elements will always occupy their own

unique row.

Even if there is enough space,

block-level elements will also

always start a new line.

... and we've seen tonnes of these!

```
>
            <address>
<h1>-<h6>
            ,
            >
<
            <d1>
<blook<br/>quote>
```

```
.box {
    display: block;
}
```

I'm a block-level element. I'm a block-level element.

I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element I'm a block-level element. I'm a block-level element I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element.

I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element. I'm a block-level element.

Inline elements are a little different. They do not start a new row and only take up as much space (width) as they need.

# If there is any space leftover, inline elements are rendered side-by-side.

## inline elements, but we will be learning a few more.

We've only seen a few

However, they don't accept width or height properties, and you cannot set the top and bottom margins.

```
.box {
   display: inline;
}
```

Clouds are free.

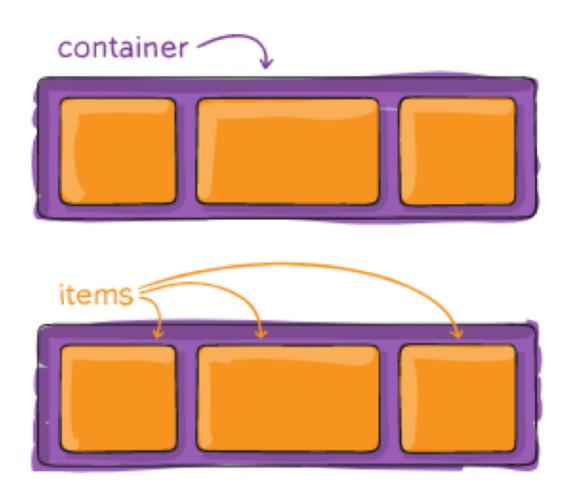
They just float around the sky all day and have fun.

Absolutely no pressure. You are just a whisper floating across a mountain. Isn't that fantastic?

### So, let's use flexbox.

## In order to use it, you must apply it to a parent container.

# The parent becomes the flex container and the children become flex items.



```
div {
   display: flex;
}
```

# for flexbox; don't worry about them yet.

There are *many* other properties

For now, we'll use it to create

multiple column layouts.

### Supplemental Video

Build an HTML + CSS Layout with Flexbox

https://www.youtube.com/watch?v=aRMIdKRYg6c