Flexbox and Layouts

GENERAL
ASSEMBLY

Learning Objectives

- **Describe** the difference between Flexbox and Floats for layouting
- **Describe** and **implement** common Flexbox CSS properties
- **Apply** Flexbox to effectively create a layout that we provide

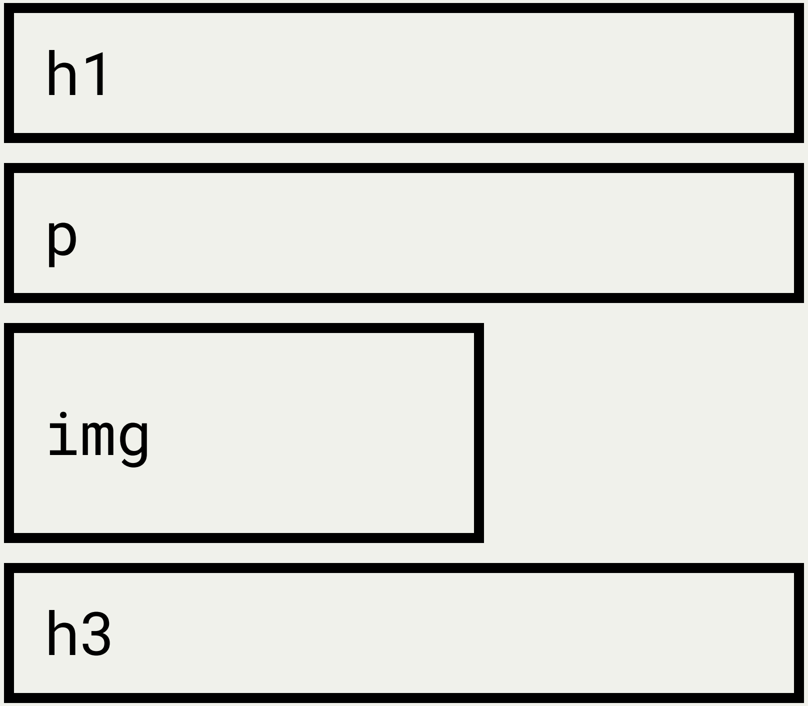
Agenda

- Review
- Flexbox Overview
- Flexbox Codealong
- Exercise: Flexbox Froggy
- Codealong: Typical Layout with Flexbox
- Lab time: Relaxr Column Layout

Review: Floats

GENERAL ASSEMBLY

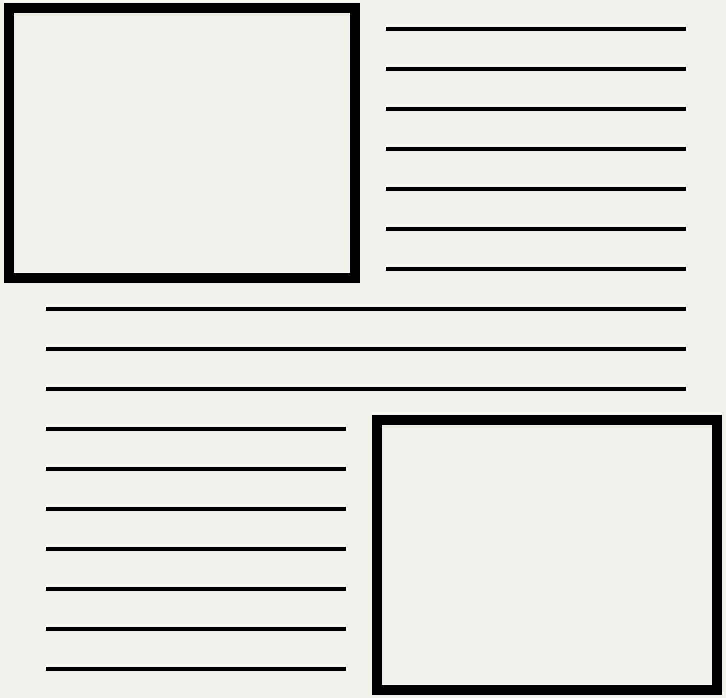
Document Flow



- How elements sit on a page
- The natural flow of a webpage

But what if we want to change that?

What are floats?



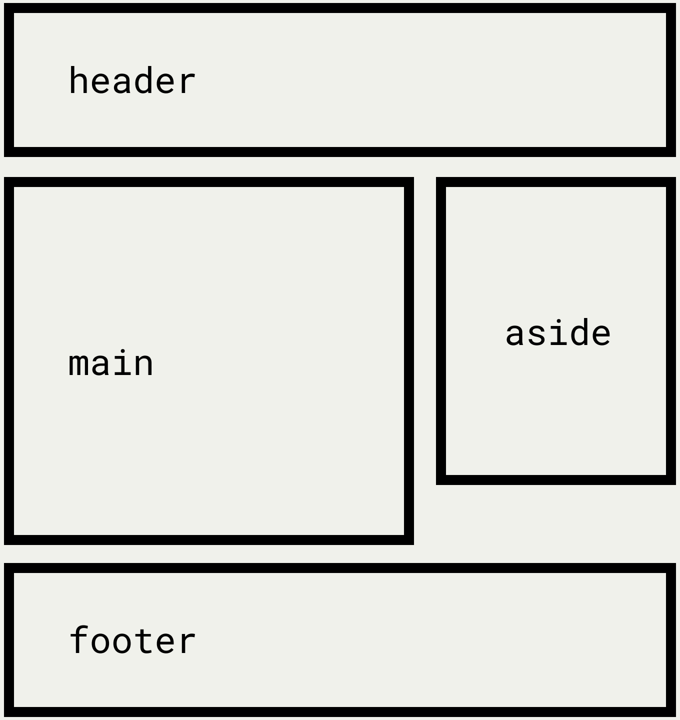
Elements with the float property:

- Remain in the document flow
- They can be pushed to the left or right
- Other content can flow around them

Float

```
element {
  float: left | right | none;
}
```

What can we do with float?



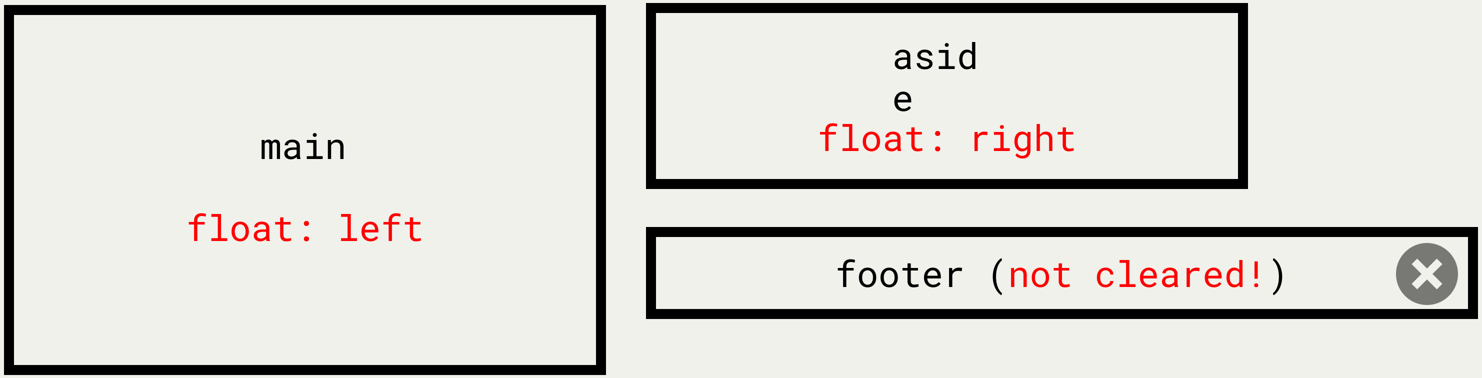
Beyond just wrapping text:

- Building full page layouts
- Smaller parts of the page
  - Navigations
  - etc.

Clear

A problem with floats

Once you have floated something, everything after it in the document flow will float up and try and fill whatever gap there is

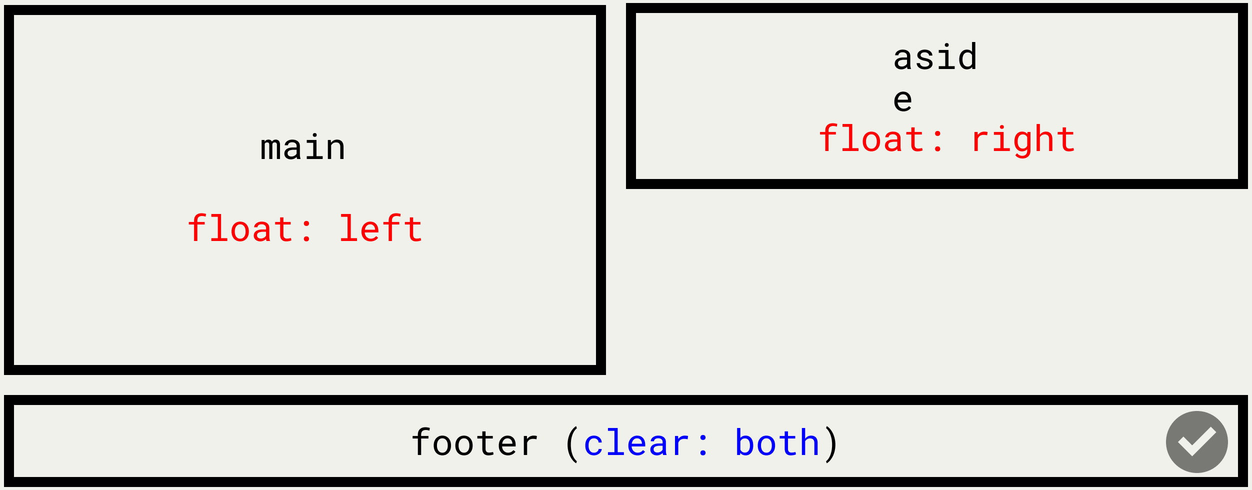


main

float: left

asid
e
float: right

footer (not cleared!) ✖



main

float: left

asid
e
float: right

footer (clear: both) ✓

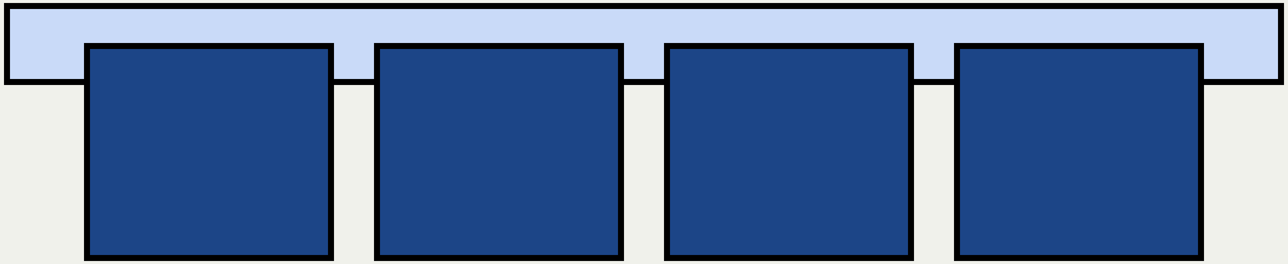
Clearing

Clearing is drawing a line and saying things can't float past this point

```
element {
  clear: left | right | both | none;
}
```

The Collapsing Parent

If everything inside an element is floated, that element will have a height of zero



Unless you clear it! (Solutions here)

So how do we clear a float?

Techniques for Clearing

1. **The Empty Div Method**
2. **The Overflow Method**
3. **The Clearfix Method**

The Empty Div Method

Adding an empty element to your HTML that has **clear both** attached to it. Add it wherever you need to stop the floats

```
...

<div class="clear"></div>

...
```

```
...

.clear {
    clear: both;
}

...
```

The Overflow Method

Add **overflow auto** or **overflow hidden** to the parent element that contains all of the floats. This will force the element to care about floated elements

The Clearfix Method

Use the clever **:after** CSS pseudo-selector to clear the floats. This pseudo-selector adds content to the page, and you can style that content to have **clear both**

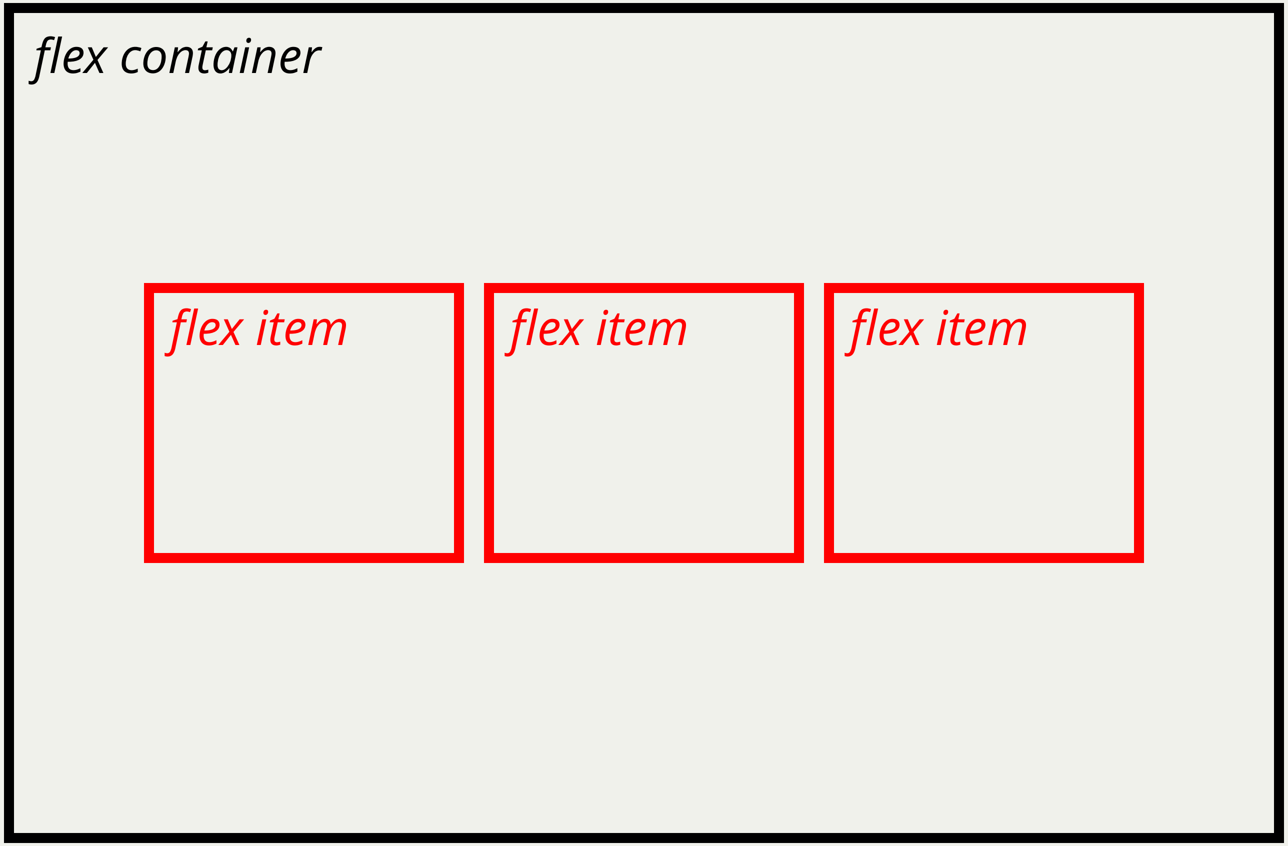
```css
.clearfix:after {
    content: "";
    visibility: hidden;
    display: block;
    height: 0;
    clear: both;
}
```

Flexbox

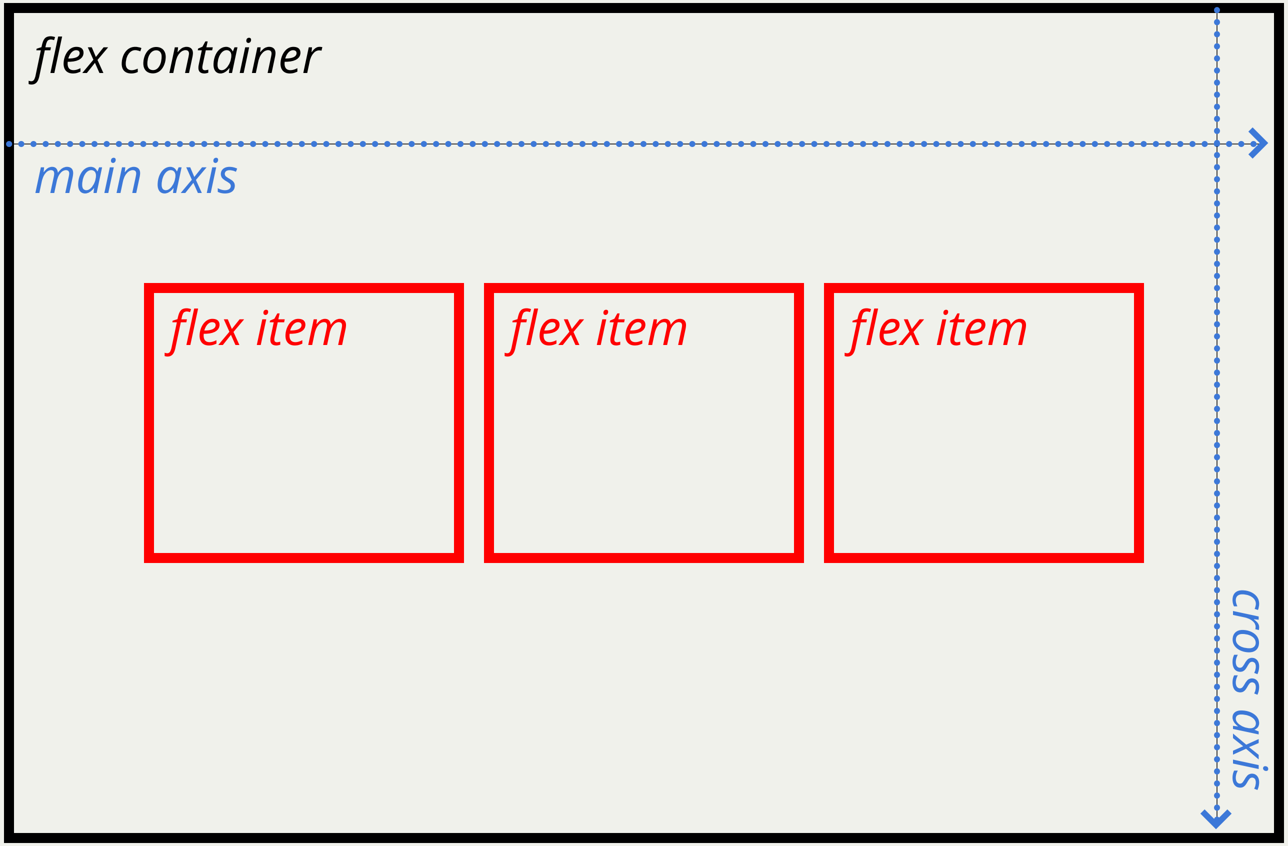
What is it?

- A new CSS3 display value
- Two Fundamental Concepts
  - Flex Container
  - Flex Item

Terminology



Terminology



Flex Container?

When you set a parent element to have **display: flex** or **display: inline-flex**

It becomes a flex container, and the children become flex items

Flex Direction

A way to change the direction of the main axis of that flex container

- row
- column
- row-reverse
- column-reverse

Flex Wrap

Flexbox will try to work with the widths, but if it can't - it'll make them flexible!

- nowrap
- wrap (use the widths and fit the heights!)
- wrap-reverse (flip the cross axis!)

Order

The default order is 0. The order property receives a number without a unit

Justify Content

How are the items aligned on the main axis?

- flex-start
- flex-end
- center
- space-between
- space-around

Align Items

Pretty much the same as Justify content, but focussed on the cross axis. How are the items aligned on the cross axis?

- flex-start
- flex-end
- center
- baseline
- stretch

Align Content

Same values as justify content. Align content organises the extra space on the cross space. Only works when we have multiple lines in our flex container! (needs a wrap)

- flex-start
- flex-end
- center
- space-between
- space-around

Align Self

Pretty much the same as Justify content, but focussed on the cross axis. How is a particular item aligned on the cross axis?

- flex-start
- flex-end
- center
- baseline
- stretch

Flex

What does flex do? It answers what do I do with the extra space along the main axis? And what do I do when I don't have enough space?

How should I scale myself up or down when I have extra space or not enough space? It is a ratio

Flex

I want twice as much space, or the same amount, or half the amount

Flex receives three parameters:

- flex-grow
- flex-shrink
- flex-basis

Flex Grow

Flex Grow says that when we have extra space, how should we divide it across everyone on the same line

How much should I take up in terms of space? If I have flex-grow 10... I am going to take up ten times more of the extra space than the other elements

Flex Shrink

Flex Grow says that when we don't have enough space, how should we divide it across everyone on the same line

How much should I give up in terms of space? If I have flex-shrink 10... I am going to shrink myself ten times more than everything else

Flex Basis

In an ideal world, this is how wide the element should be

All together now

```
el {
    flex: flex-grow | flex-shrink | flex-basis;
}
```

Exercise!

Flexbox Froggy

Flexbox Resources

- CSS Tricks
- Flexbox.io
- Flexbox Zombies
- An animated guide
- Everything you need to know
- MDN
- Flexbox Defence

Exercise!

Relaxr Part Two

Homework

- Finish Relaxr Part Two
- Keep playing with Flexbox

Next class?

- A little bit about the position property in CSS
- But mostly an exercise

Q & A

GENERAL ASSEMBLY

Feedback Time

Lesson 5: **Flexbox and Layout Lab**

https://ga.co/fewd32syd

Thanks!

GENERAL ASSEMBLY