WEB DESIGN DECAL

LECTURE 5

CSS Positioning and Floats

Or, how to align and orient elements on your website

Andy Qin

Sad Announcement

This will be my last lecture for Web Design DeCal…

I've overcommitted this semester, so I'm not doing quality work. But it's been a great 2 years being involved with WDD.

Sad Announcement

I ❤ you guys.

Positioning can be annoying.

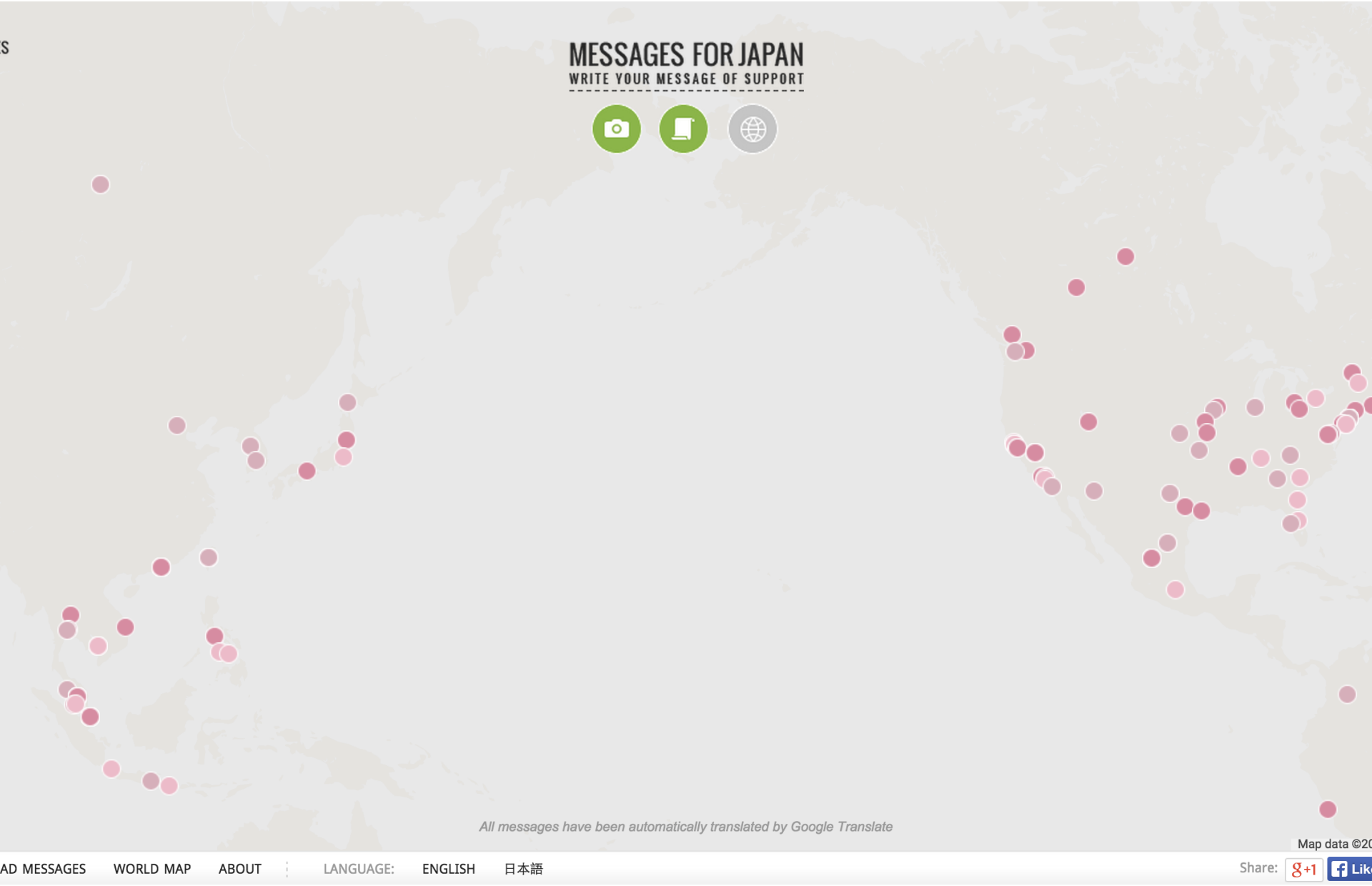
Incredibly Helpful Resource

Want a primer of CSS positioning and more?

learnlayout.com

Motivation for CSS Positioning

So far we've talked about how elements stack next to one another. Haven't talked enough about when they overlap.



Beautiful interactive maps

Creative (vertical) navigation bars

iPhone made of CSS + HTML

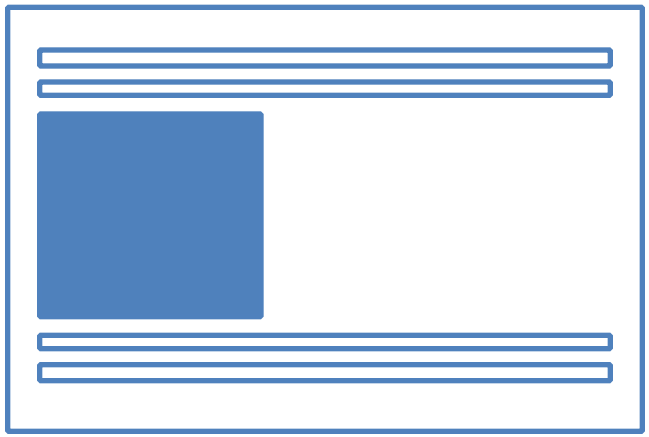
4 Ways to position in CSS:

CSS Positioning - position: static;

CSS Positioning - position: static;

The default!

Cannot use offset properties: top, bottom, right, left

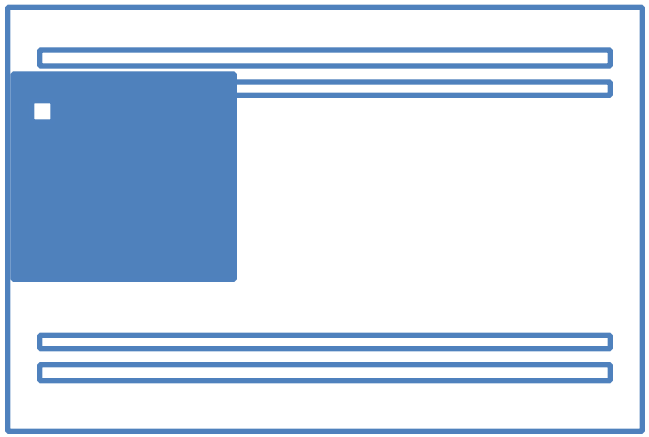


CSS Positioning - position: relative;

CSS Positioning - position: relative;

Behaves just like static unless you add extra properties: top, bottom, right, left

Example: Set top: -10px; left: -10px;

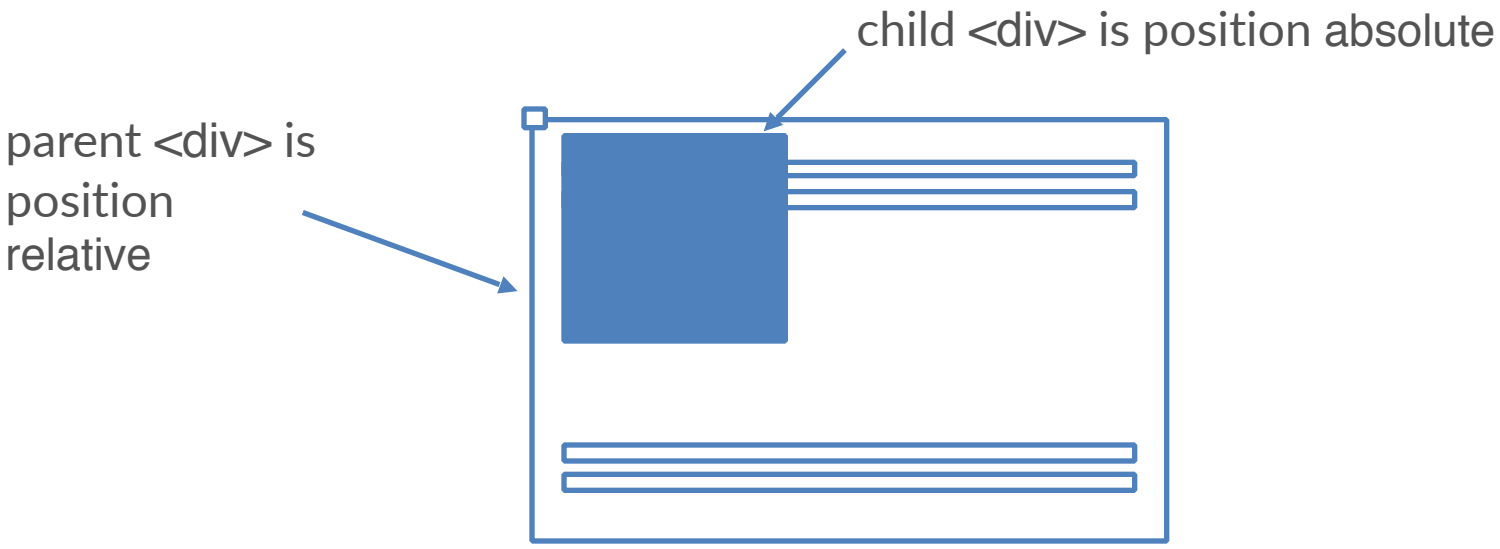


CSS Positioning - position: absolute;

CSS Positioning - position: absolute;

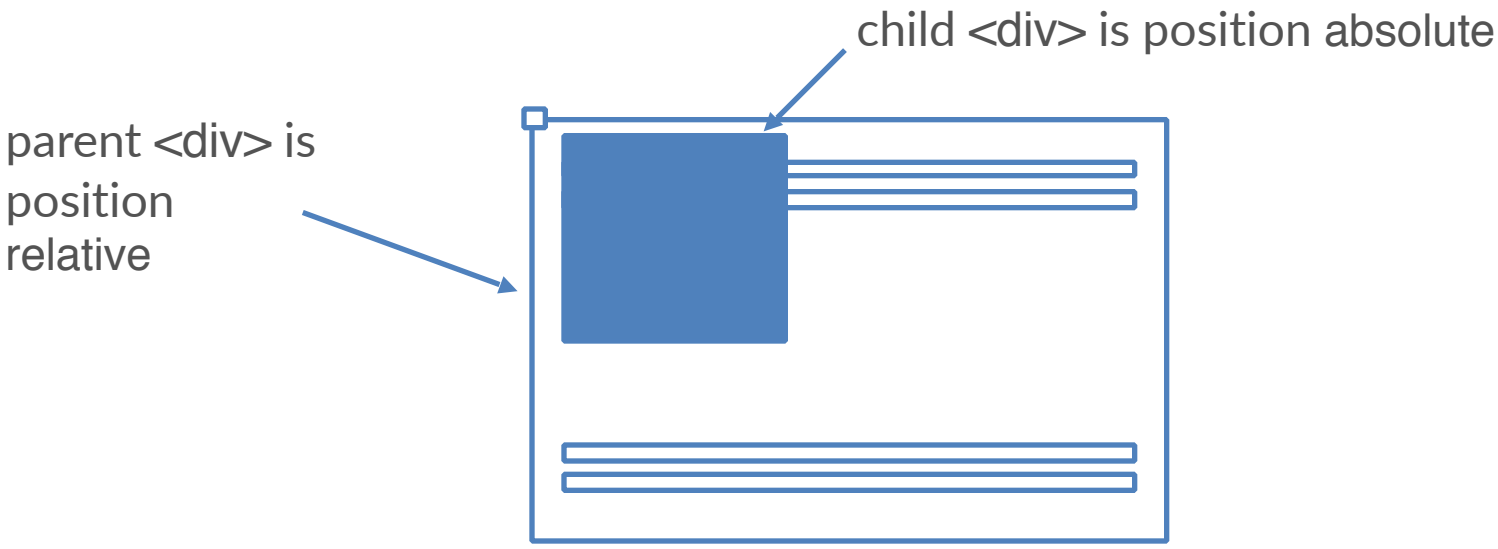
Element moved relative to first non-static parent

Example: Set top: 5px; left: 10px;



CSS Positioning - position: absolute;

Think of it as “rudely disregarding positioning of other elements” unless it’s the parent. Doesn’t disobey parent.



CSS Positioning - position: absolute;

Element is moved relative to first non-static parent element!

```
<div id="container">
  <div id="content">
    Hello, World!
  </div>
</div>
```

```
#container {
  position: relative;
}
#content {
  position: absolute;
  top: 20px;
  left: 20px;
}
```

CSS Positioning - position: absolute;

If no non-static parent element, then relative to <body>

```
<div id="container">
  <div id="content">
    Hello, World!
  </div>
</div>
```

```
#container {
  position: relative;
}
#content {
  position: absolute;
  top: 20px;
  left: 20px;
}
```

CSS Positioning - position: fixed;

CSS Positioning - position: fixed;

Simple: element stays, even when scrolling, relative to window

Think navigation bars!

 is position fixed

CSS Positioning - Common confusion

Q: How do top, left, right, and bottom work with margin-top, margin-left, margin-right, and margin-bottom?

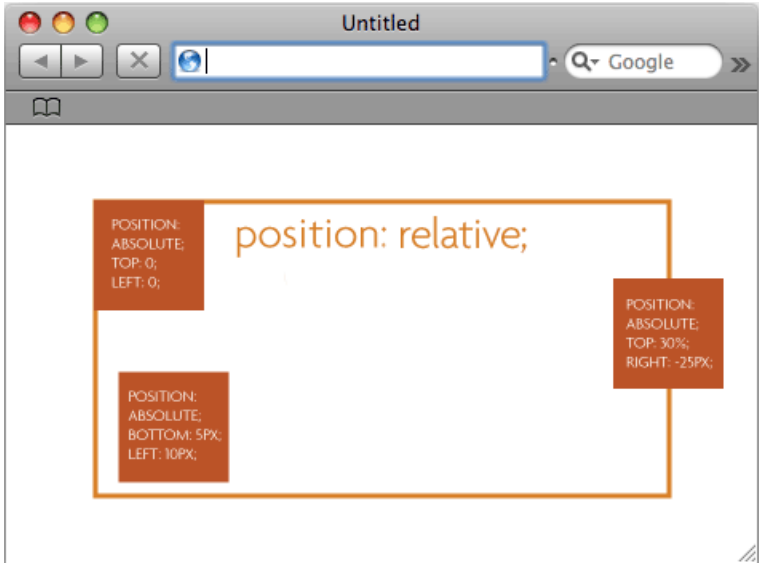
A: They work with each other. Non-static elements still follow the CSS box model, just have the new offset properties in addition.

CSS Positioning - Common practice

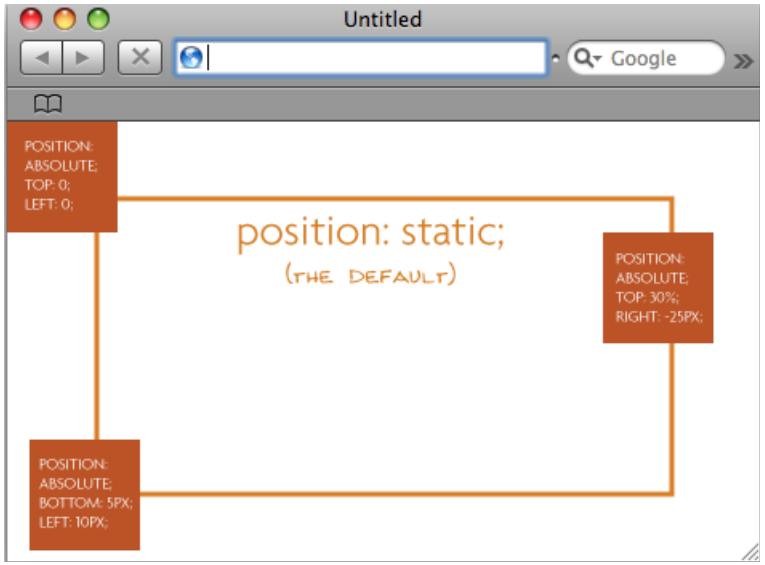
Q: What's a great practice for using these positioning rules?

A: Commit this to memory, dude. **Relative** parents with **absolute** children. This way, you have precise control of where the children go, even when the browser size changes.

CSS Positioning - Common practice



vs



CSS Positioning - z-index

CSS Positioning - z-index

Used to determine which layers are “on top of” other layers.
Can only be used on non-static elements!
(only for position: relative, position: absolute, and position: fixed)

Please control yourselves and keep z-index in reasonable range.
(i.e. z-index: 10)

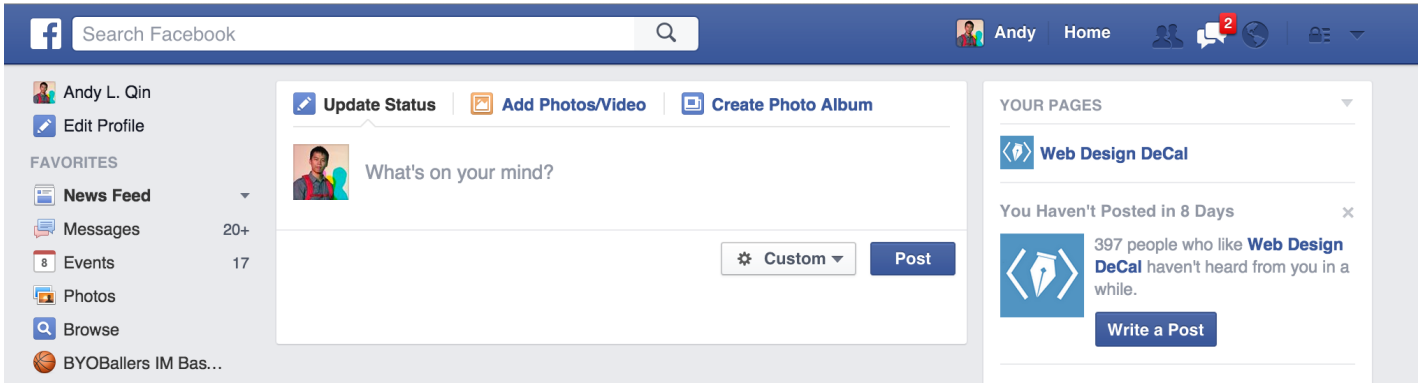
CSS Positioning - z-index

Check this out, yo:

bit.ly/z-index-1

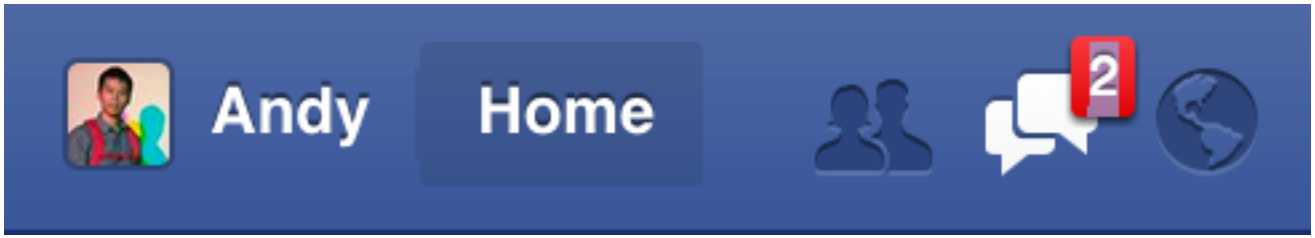
How would you?

Make the Facebook blue bar?



How would you?

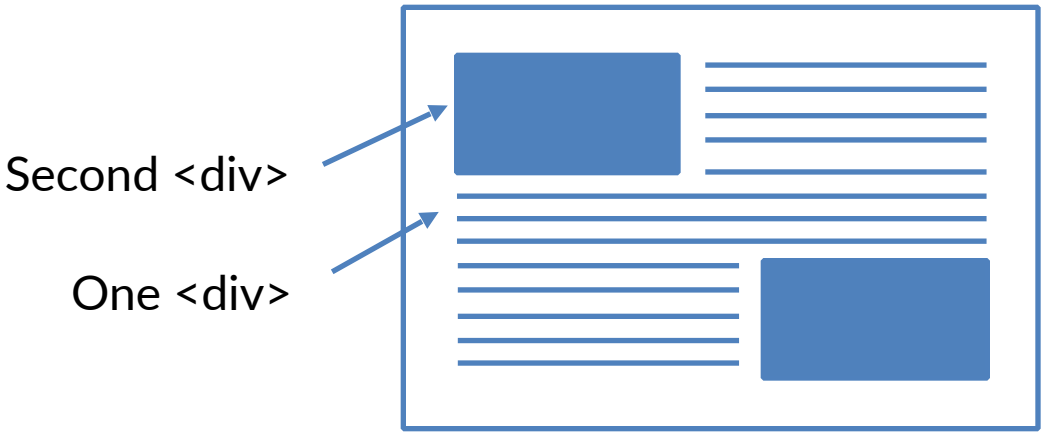
How about just the notification widget?




Lets talk floats!

CSS Positioning - float

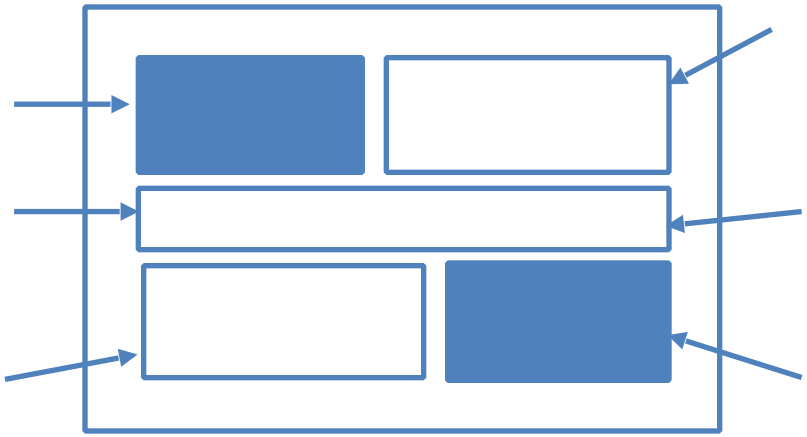
Floats allow elements to be part of the flow of the page.
Let's say you wanted to create this:



CSS Positioning - float

So why floats?

If you tried to this with display: inline-block...



Ewwww.

CSS Positioning - float

Can float: left or float: right

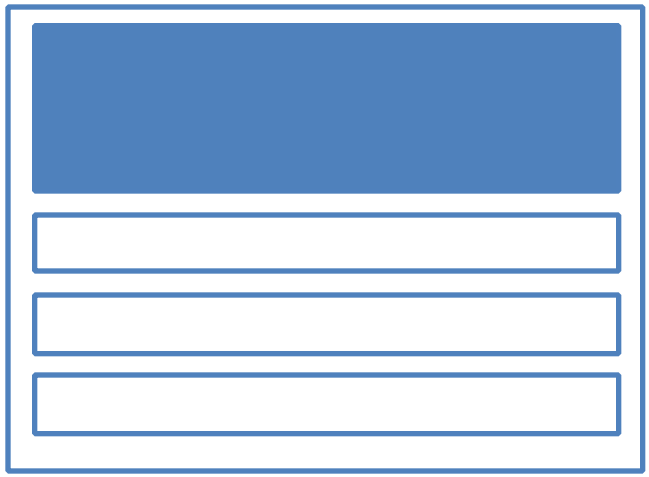
Even for block elements, float takes precedence and acts like display: inline-block

The next elements will stack next to the floated element!

Most importantly, text wraps around floated elements

CSS Positioning - float

Just standard divs:



CSS Positioning - float

But just add float and:

bit.ly/positioning-1

CSS Positioning - float: left;

Suppose you're creating a menu:

Could use display: inline-block... but this creates gaps in between!
May or may not be desired...

CSS Positioning - float: left;

Suppose you're creating a menu:

Use float: left instead!

bit.ly/float-1

How to stop floating left or right?

Need to use clear: both (aka “clearfix”)!

Breaks the float, so we finally stack vertically again, like block elements should

How to stop floating left or right?

We usually always define a class “clearfix”:

```
.clearfix {
  clear: both;
}
```

```
<div class="clearfix"></div>
```

And we put it after all the elements that are floated, so we can start the other content on a new line.

CSS Positioning - clear: both;

Blue box is float: left;

CSS Positioning - clear: both;

White boxes have a clear: both right before them!

Questions?