# CSS Positioning

Andy Qin, Programming Instructor

WEB DESIGN DECAL
PROGRAMMING | FALL 2014

# CSS Positioning in a NUTSH

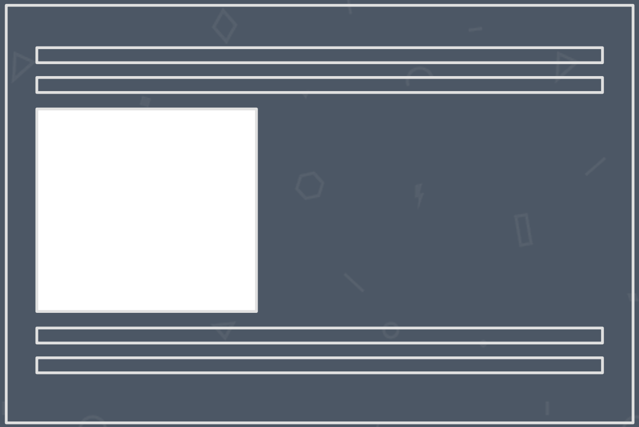
Positioning can be annoying if you don’t know what’s happening.

# CSS Positioning

- 4 ways to position:
  - static
  - relative
  - absolute
  - fixed
- 2 ways to float:
  - left
  - right

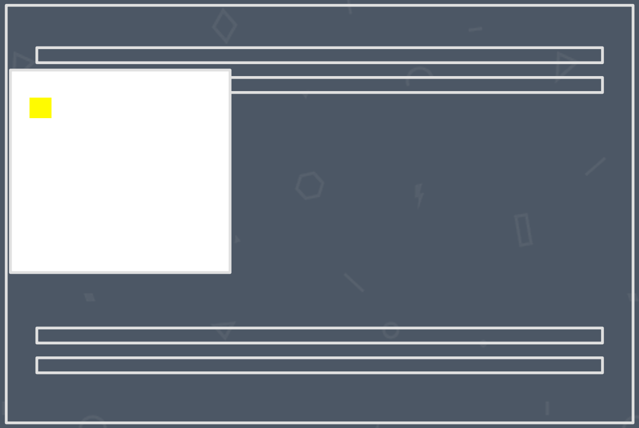
# CSS Positioning - *position*: *static*;

- The default!
- **Cannot** use offset properties: top, bottom, right, left



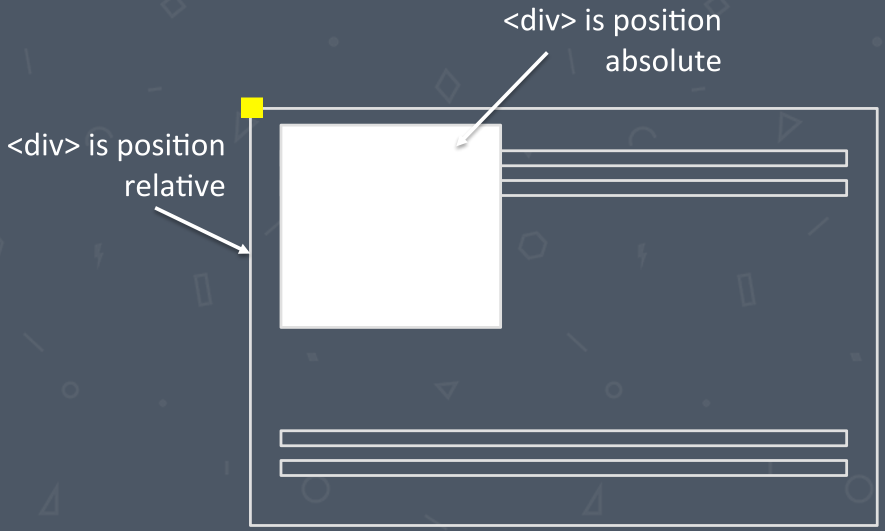
# CSS Positioning - *position*: *relative*;

- **Can** use offset properties: top, bottom, right, left
- Example: Set top to -10px, left to -10px



# CSS Positioning - *position*: *absolute*;

- Element moved relative to **first non-static parent**
- Example: Set left to 10px, top to 5px

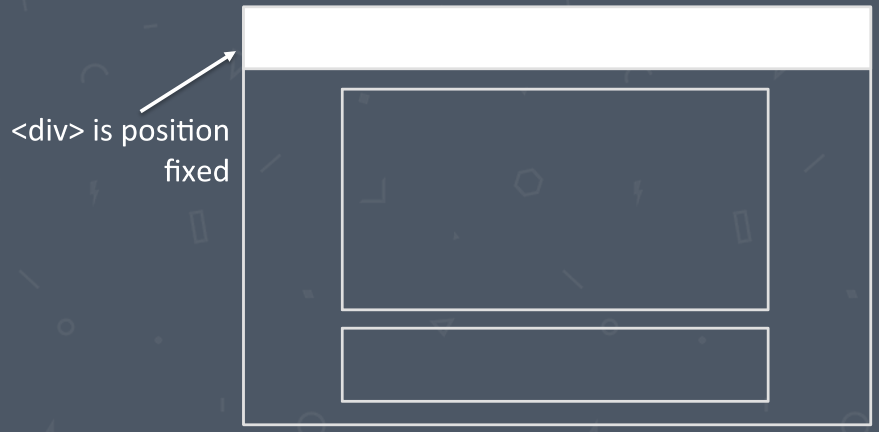


<div> is position absolute

<div> is position relative

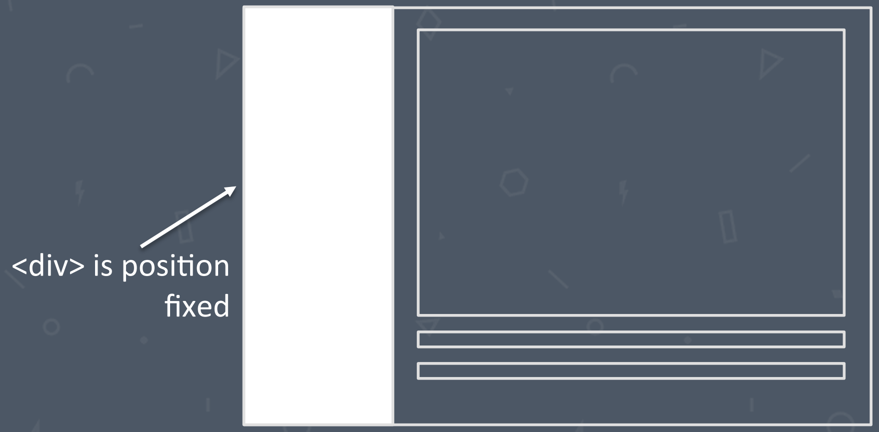
# CSS Positioning - *position*: *fixed*;

- Element stays, even when scrolling, relative to **viewport**
- Think horizontal navigation bars



# CSS Positioning - *position*: *fixed*;

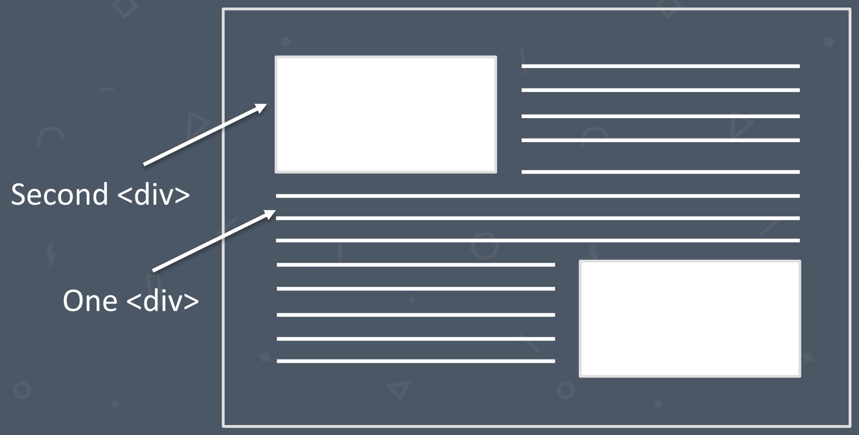
- Element stays, even when scrolling, relative to **viewport**
- Think vertical navigation bars



Let’s talk
Floats! YAAAAAY
NOT AGAIN FUC

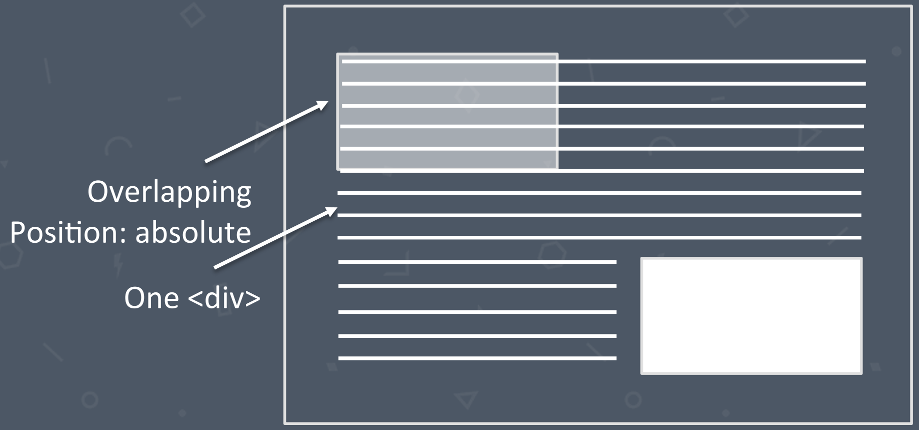
# CSS Positioning - *float*

- Floats allow elements to be part of the *flow* of the page



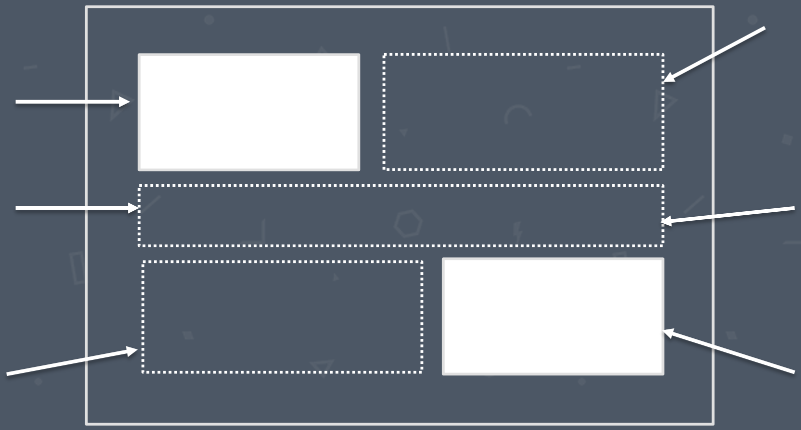
# CSS Positioning - *float*

- So why floats?



# CSS Positioning - *float*

- So why floats?
- If you tried to this with *display*: *inline-block*…



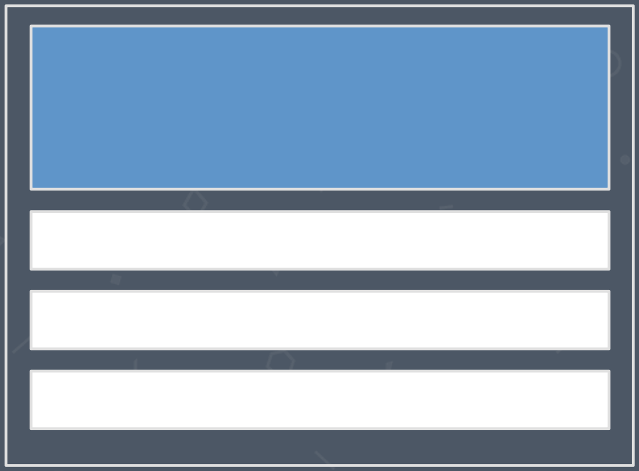
Ewwwwwwwwwwwww

# 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 div!

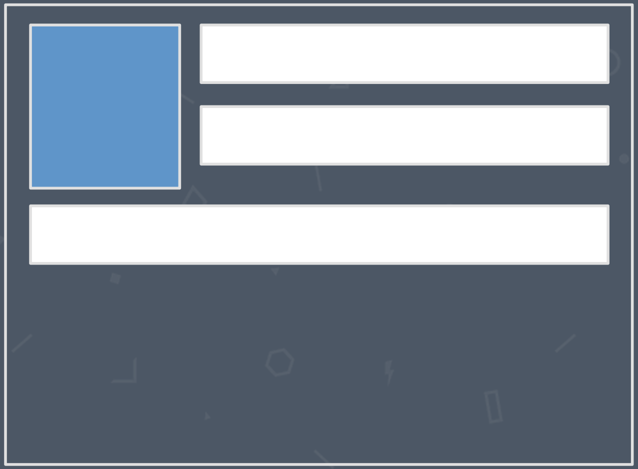
# CSS Positioning - *float*

Just standard divs:



# CSS Positioning - *float*

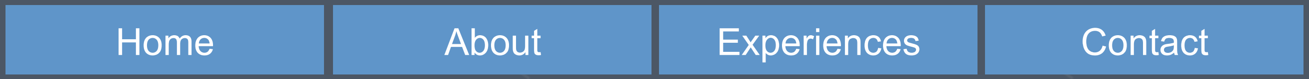
But just add float and:



Example at: www.tinyurl.com/calhacks

# CSS Positioning - *float: left*;

Suppose you’re creating a menu:

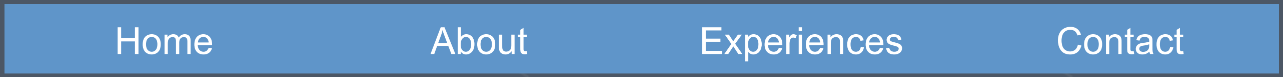


Home About Experiences Contact

Could use *display*: *inline-block*… but this creates gaps in between!

# CSS Positioning - *float: left*;

Suppose you're creating a menu:



So use *float*: *left* instead!

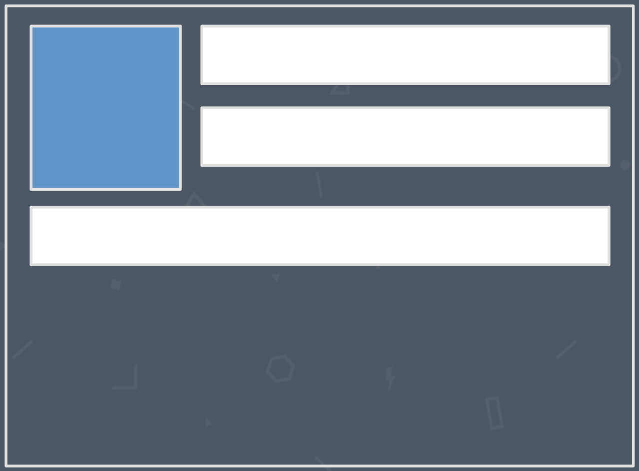
Example at: www.tinyurl.com/calhacks2

# How to stop floating left or right?

- Need to use *clear*: *both*!
- Breaks the float, so we finally stack vertically again, like block elements should

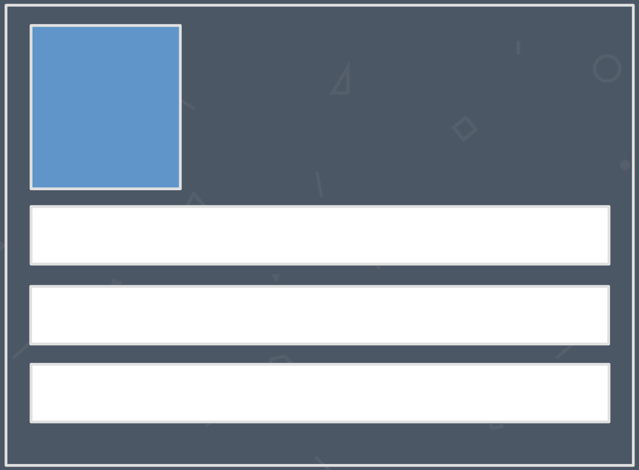
# CSS Positioning - *clear*: *both*;

Blue box is *float*: *left*;



# CSS Positioning - *clear*: *both*;

White boxes use *clear*: *both*;



# Questions?

WEB DESIGN DECAL
PROGRAMMING | FALL 2014