

WEEK 4 | PROGRAMMING CSS POSITIONING

Shawn Park & Jeff Zhan

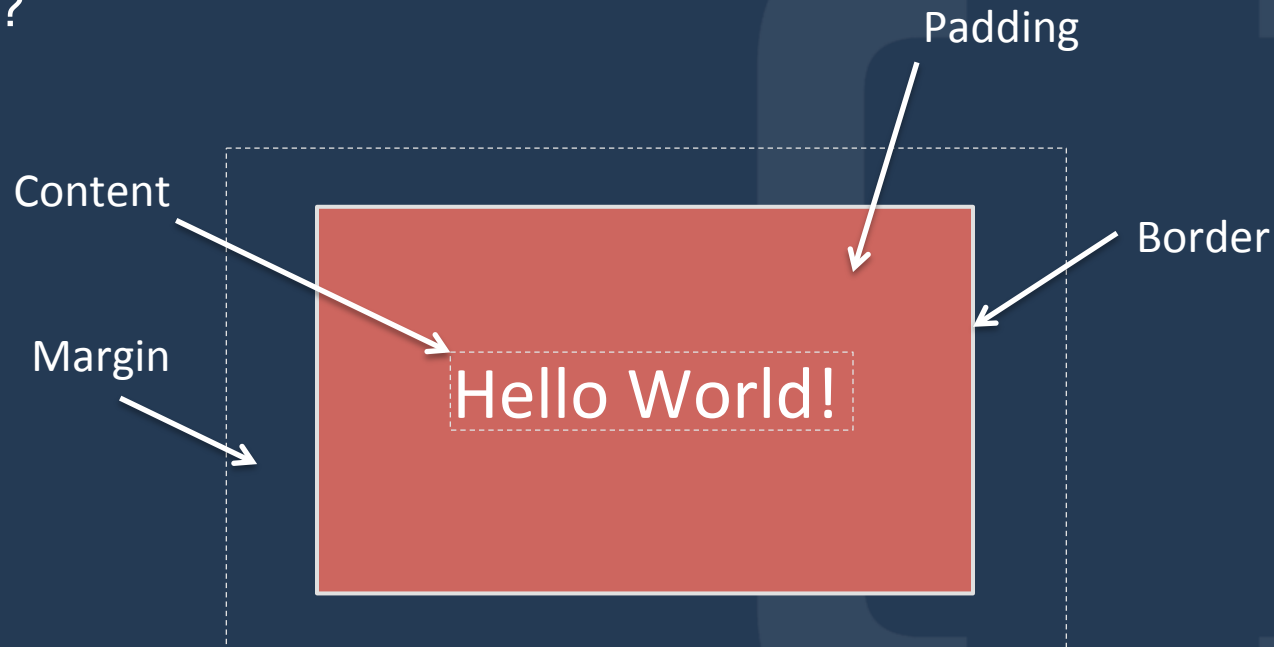


Web Design DeCal

DESIGN MEETS PROGRAMMING

Review

- We learned about the CSS Box Model. What are the 4 components of it?





Today's Outline

1. CSS Positioning
2. CSS Floats





Goal Today:
To Structure your Web Pages
Even Better



CSS Positioning



CSS Positioning

- 4 types of positioning in CSS:
 - `position: static`
 - `position: relative`
 - `position: absolute`
 - `position: fixed`
- By default, all your divs and elements are *position: static*

position: static

- Default position, follows the natural flow of the page
- **Cannot** use offsets: top, left, bottom, right



position: relative

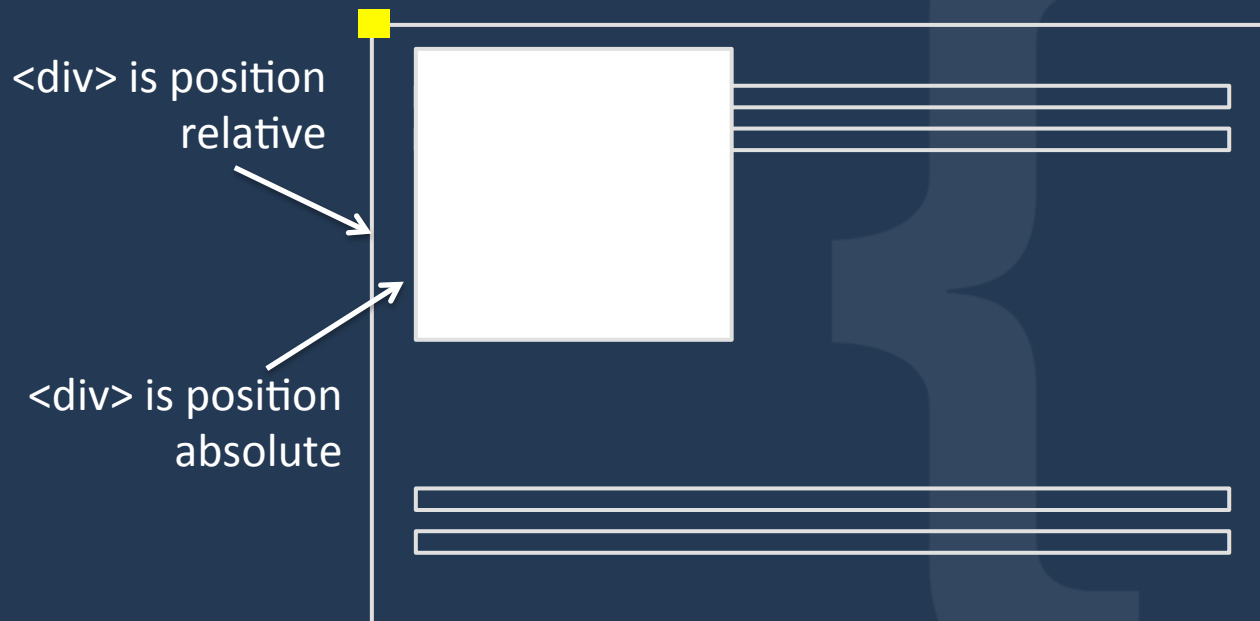
- Offsets element using position values (top, left, right, bottom)



Example: Set top to -10px, left to -10px

position: absolute

- Element is moved relative to first non-static parent (default: *top: 0*, *left: 0*)



Example: Set left to 10px, top to 5px

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;  
}
```

position: absolute

- If no *non-static* parent element found, move relative to <body>

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

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

position: absolute

- Example of *div* relative to *div* relative to *div* (*div-ception!!*)

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

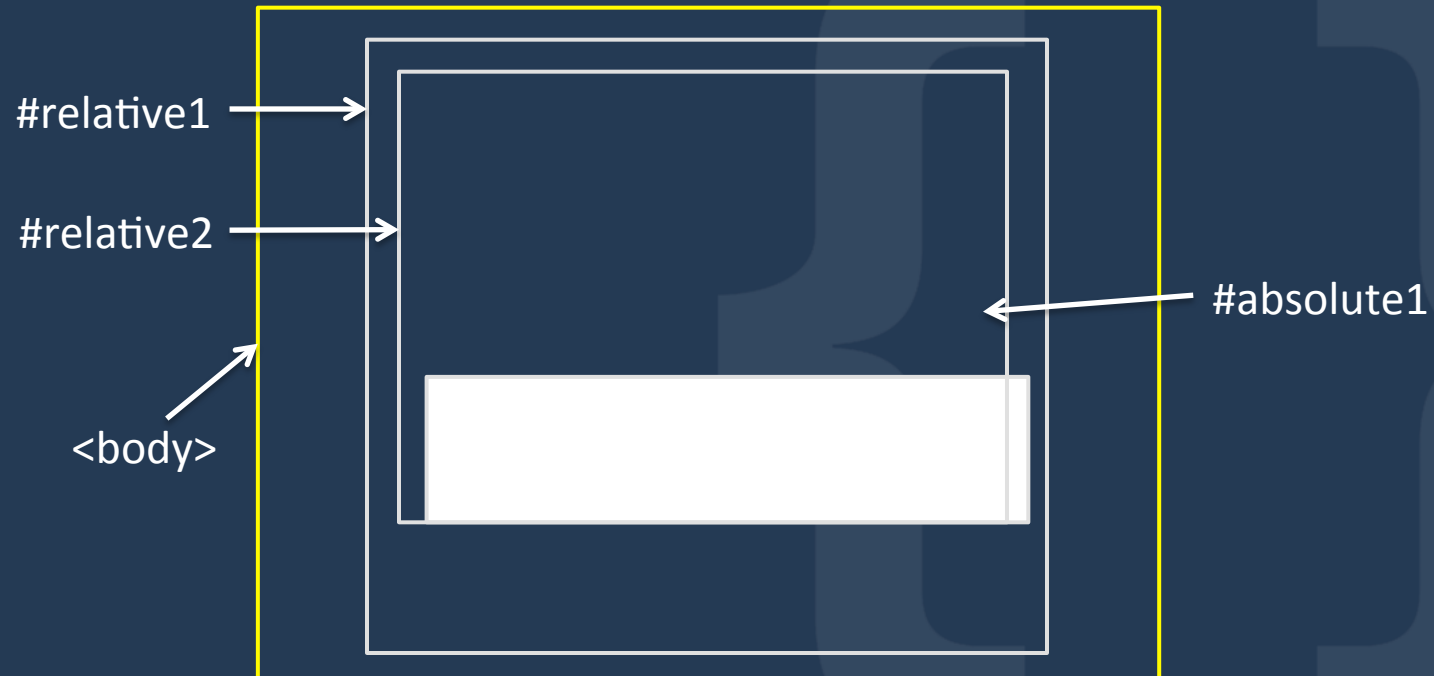
```
#container { position: relative; }
#content { position: absolute; top: 20px; left: 20px; }
#inner-content { position: absolute; top: 30px; left: 10px; }
```

#inner-content will be **relative** to #content
So in total, it is 50px from top of #container, 30px left of #container

position: absolute

```
<body>  
  <div id="relative1">  
    <div id="relative2">  
      <div id="absolute1"> ....
```

What are possible top, right, bottom, left values for #absolute1?

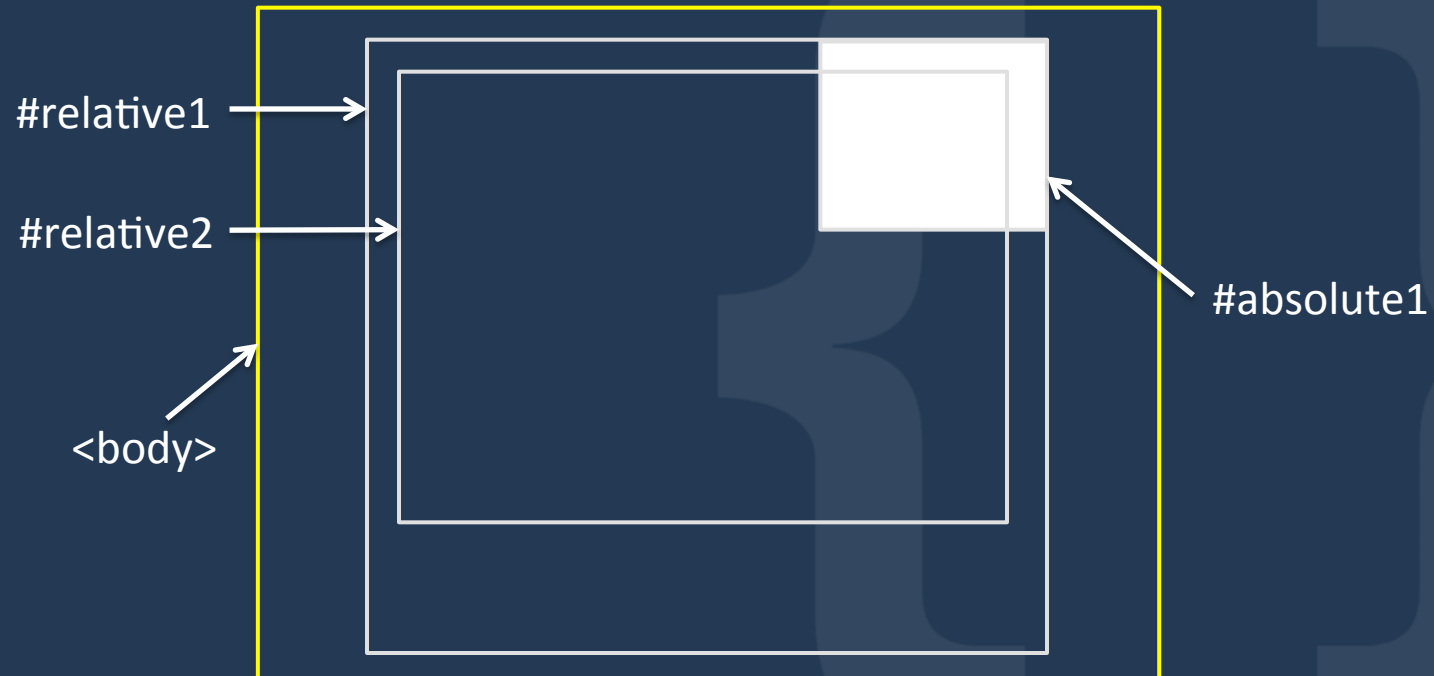


*#absolute1 is moved relative to #relative2.
So... bottom: 0, left: 5px (example)*

position: absolute

```
<body>  
  <div id="relative1">  
    <div id="absolute1">  
      <div id="relative2"> ....
```

What are possible top, right, bottom, left values for #absolute1?



*#absolute1 is moved relative to #relative1.
So... top: 0, right: 0 (don't even need to type px)*



Demo

position: fixed

- Element stays on the screen, even when scrolling. Positioned relative to browser window




Example: Set left to 10px, top to 5px

position: fixed

- Element stays on the screen, even when scrolling. Positioned **relative to browser window**

`<div>` is position
fixed

A diagram of a browser window. The window is represented by a large rectangle. Inside, there is a solid white horizontal bar at the top, representing a fixed header. Below this bar, there are two white-outlined rectangular boxes, one above the other, representing content elements. An arrow points from the text '<div> is position fixed' to the top-left corner of the white header bar.

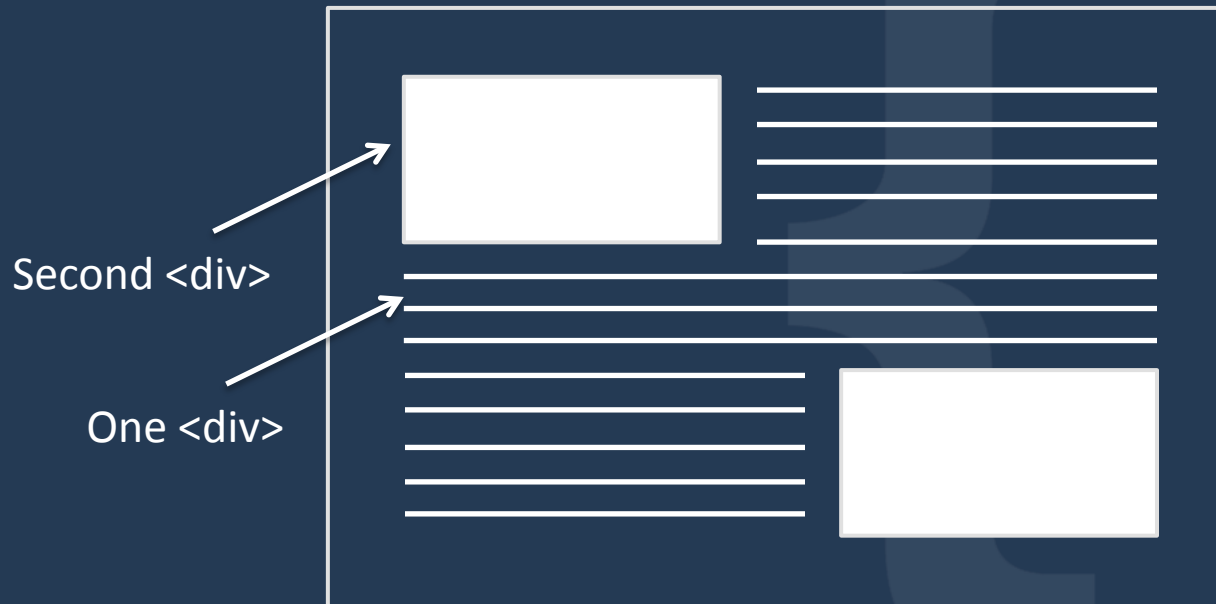


Floats

Float

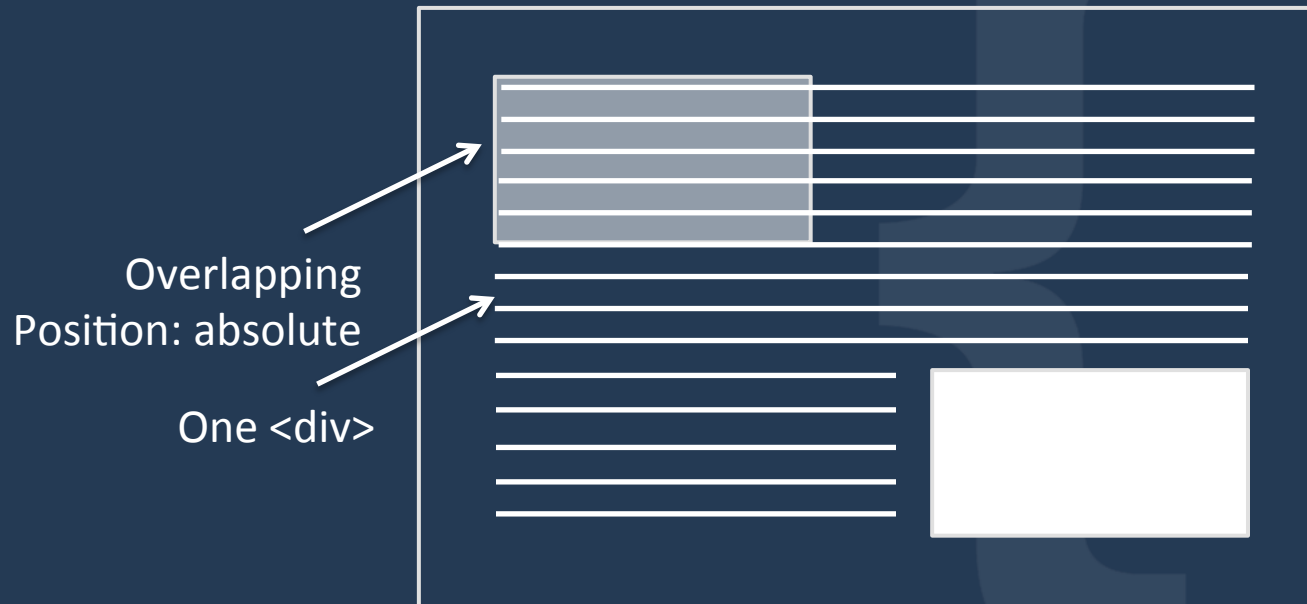
(ex: www.jsfiddle.net/r9M8Q)

- What is a *float*?
 - They allow your elements to be part of *the flow* on the page



Float

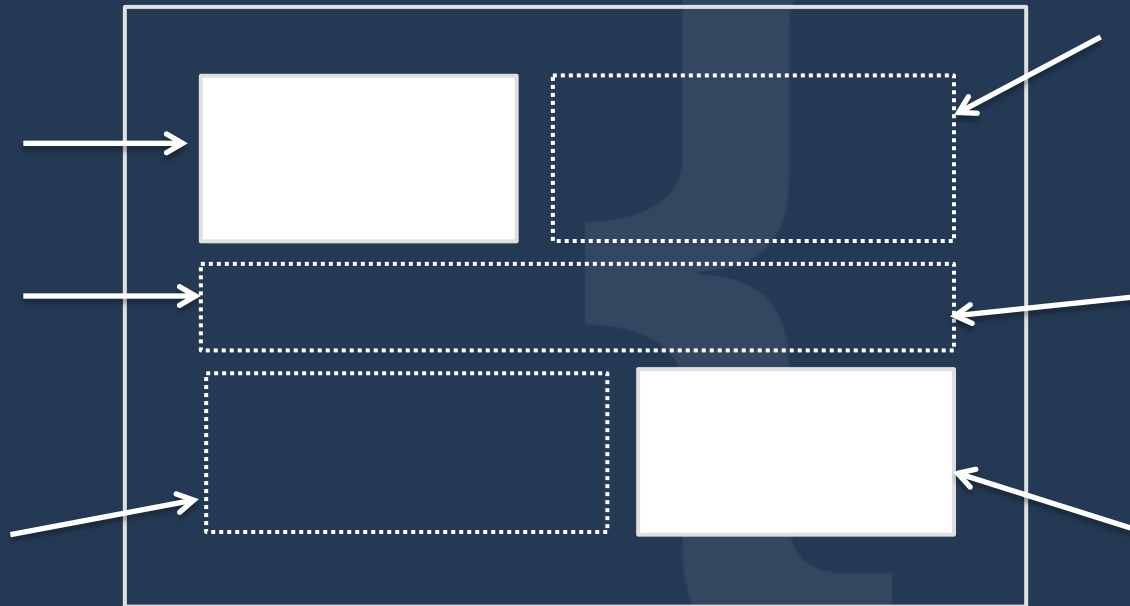
- If you made one div *position: absolute*...



Float

(ex: www.jsfiddle.net/r9M8Q)

- If you were to recreate the page with `<div>` elements set to *display: inline-block*...



Not efficient!

Float

(ex: www.jsfiddle.net/r9M8Q)

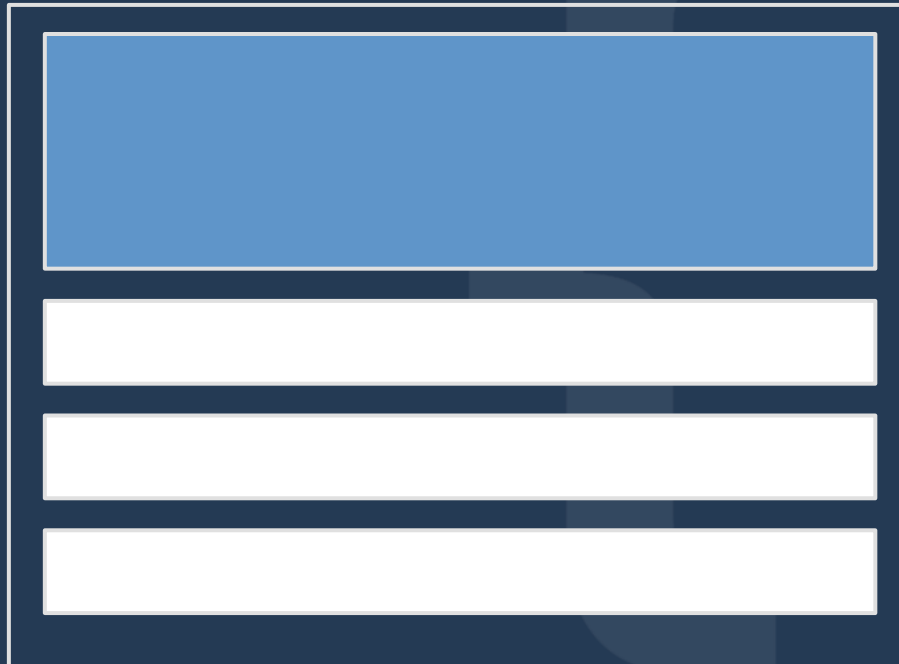
- *float*
 - 2 properties
 - *float: left*
 - *float: right*
- Even if an element is a `<div>` or has *display: block*, float will essentially bypass these properties and act like *display: inline-block*, but automatically floating left or floating right
- The following element will be next to the floated `<div>`



Float

(ex: www.jsfiddle.net/r9M8Q)

No floats, just `<div>`





Float

(ex: www.jsfiddle.net/r9M8Q)

First box, *float: left*



Float

One Use Case: Menus

Home

About

Experiences

Contact

Float

Floating Menu Items (each with set *width*)

Home

About

Experiences

Contact

Notice no gaps!

If you use *display: inline-block* for these 4 elements:

Home

About

Experiences

Contact

display: inline-block produces gaps on the sides

Float

If you use *display: inline-block* for these 4 elements:

Home

About

Experiences

Contact

display: inline-block produces gaps on the sides

- Work-around:
 - Add *margin-right: -4px*
 - Or...

*Awkward! Use margin-right: -4px,
or floats!*

```
<div>Home</div>  
>About</div>  
>Experiences</div>  
>Contact</div>
```

```
<div>Home</div><!--  
--><div>About</div><!--  
--><div>Experiences</div><!--  
--><div>Contact</div>
```



CSS Clear

- *clear*
 - *clear: both*
 - Individually, *clear: left* and *clear: right*
- If one element floats left, the following element will appear beside it
- Use *clear: both* for the following element to **break** the float, so it appears below the floating element



CSS Clear

First box, *float: left*





CSS Clear

Second box, using *clear: both*





Summary

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

All lecture material, handouts, and homework can be found at:
<http://www.thewebdesignworkshop.co>