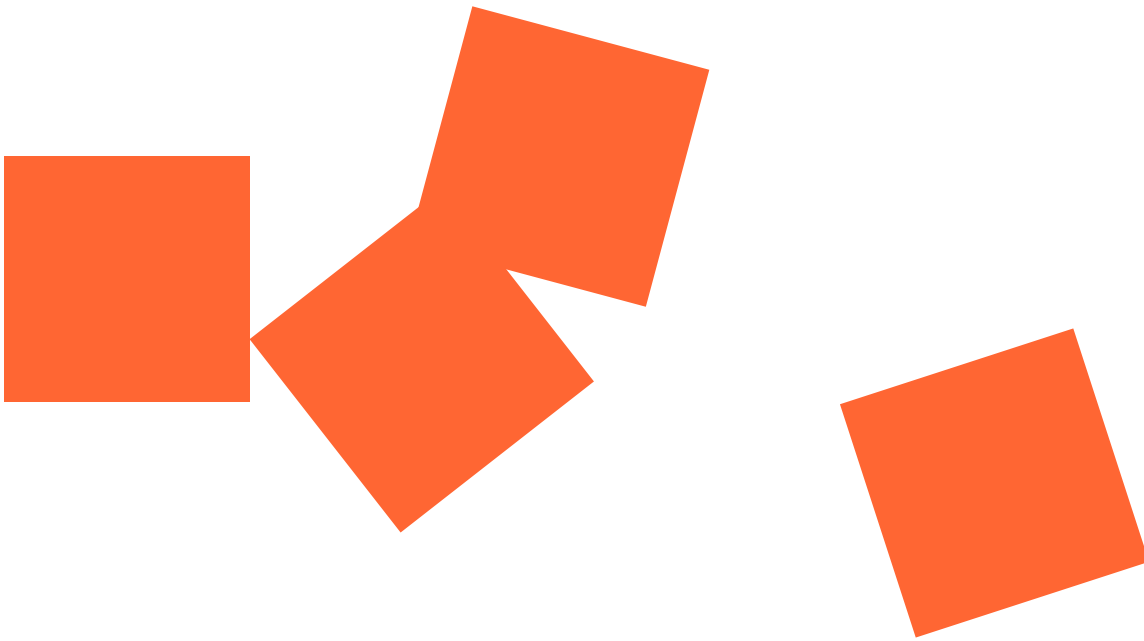
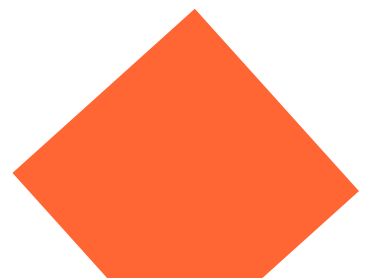


Float Basics



Floats are one of the most difficult concepts in CSS. For now, I **just** want you to read through this handout, and get a **preview** of what floats are. This will make it easier next class when we use a simple float to create text wrap. It will be included under “Code part 3,” and I will explain it then.

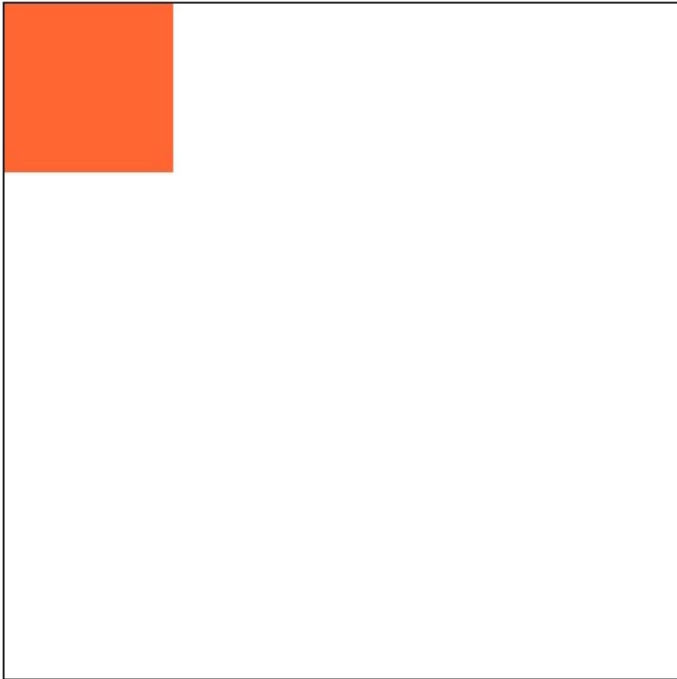


Float basics

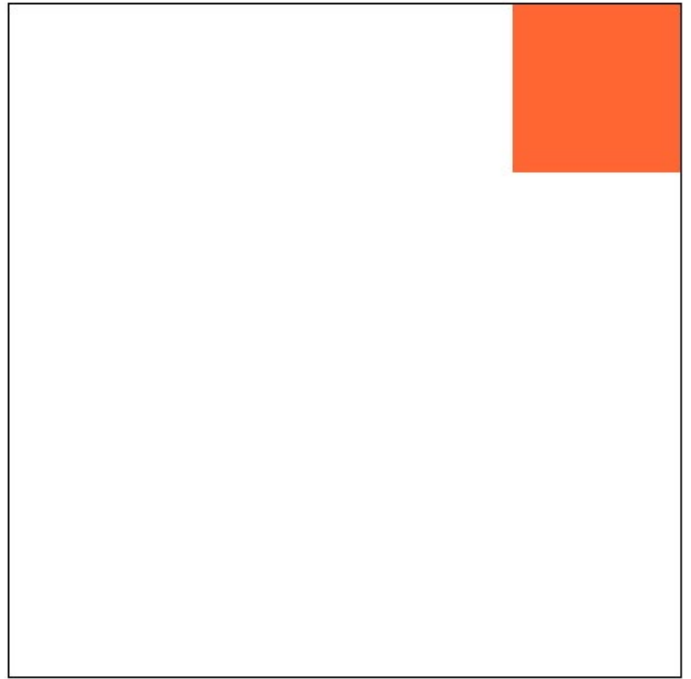
What is a float?

When you float an element it becomes a block box. This box can then be shifted to the left or right on the current line. The markup options are "float: left", "float: right" or "float: none".

"float: left" element

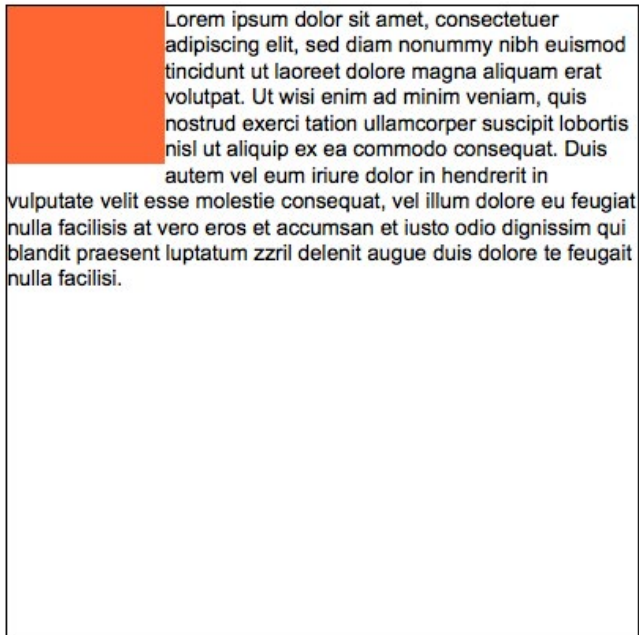


"float: right" element

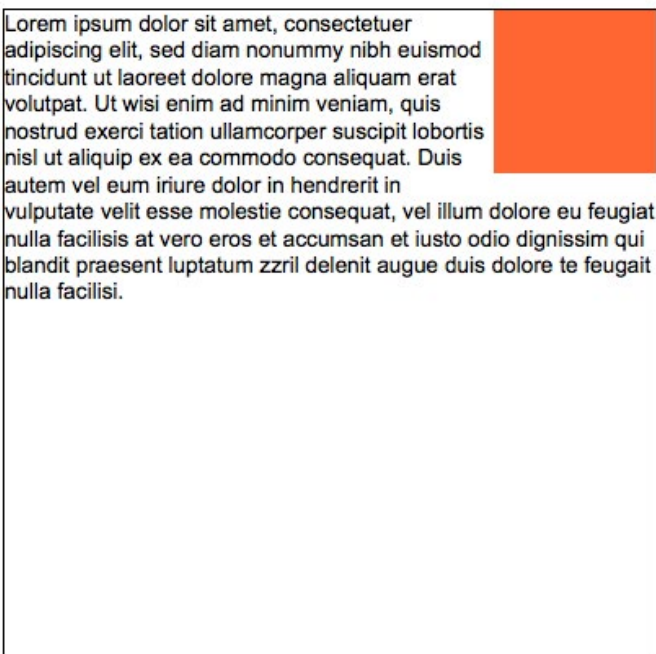


A floated box is laid out according to the normal flow, then taken out of the flow and shifted to the left or right as far as possible. Content can flow down the right side of a left-floated box and down the left side of a right-floated box.

"float: left" element with content flowing down right edge

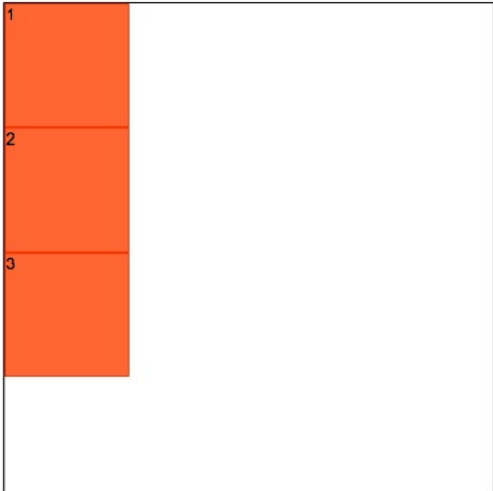


"float: right" element with content flowing down left edge

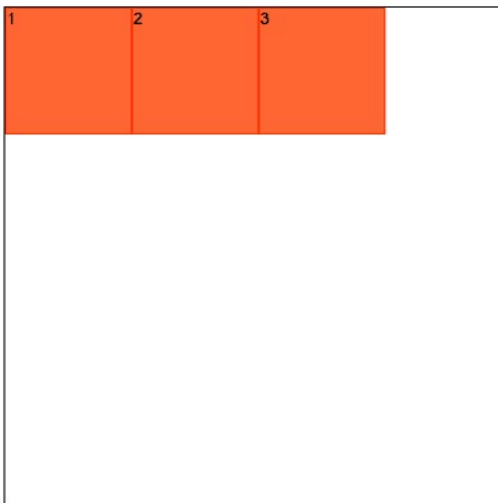


You can also put several floats beside each other.

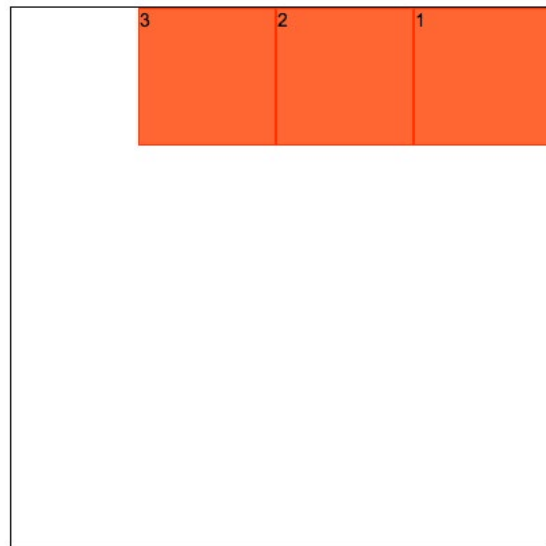
Three elements without any float applied



Three elements with "float: left" applied

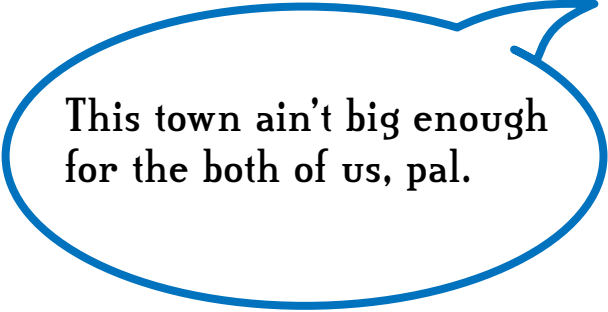


Three elements with "float: right" applied

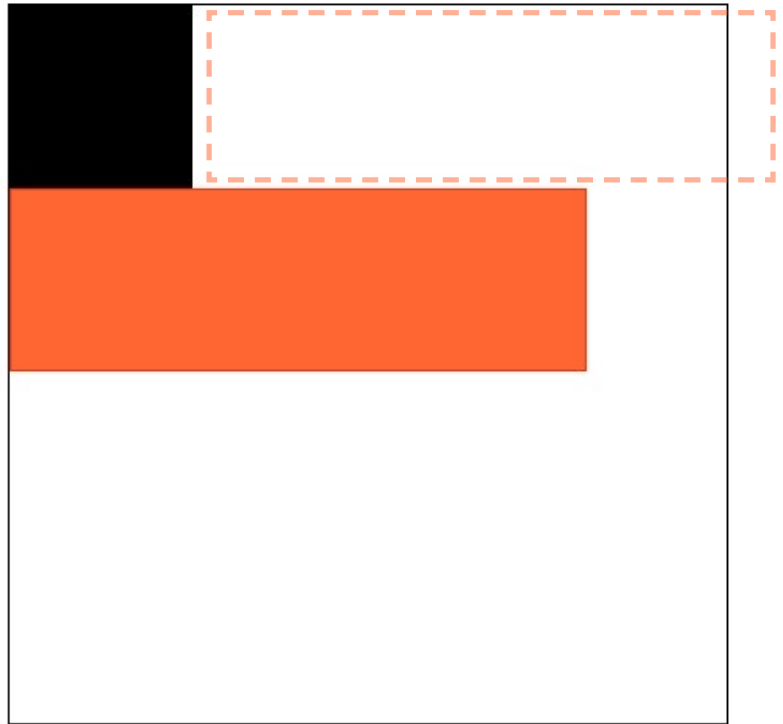


If there isn't enough horizontal room on the current line for the floated box, it will move downward, line by line, until a line has room for it.

"float: left" element shifted down



This town ain't big enough
for the both of us, pal.



This is called "float drop." It looks like your float is not working, but it is. You're just out of room.

It happens most often when you have added padding and margin but have not done the math.

You should always set a width on floated items (except if applied directly to an image - which has implicit width). W3C's Cascading Style Sheets, level 2, CSS2 Specifications states:

"A floated box must have an explicit width..."

If no width is set, the results can be unpredictable. Theoretically, a floated element with an undefined width should shrink to the widest element within it. This could be a word, a sentence or even a single character

"float: left" with no width set



Elements above and below floated elements

Block level elements above a floated element will not be affected by it. However, elements below will wrap around the floated element:

Elements that are block elements are ones that take up the whole line by default, pushing the next element afterward down to the next line.

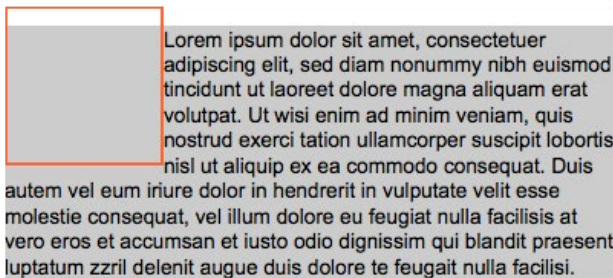
Examples of elements that are block by default:

```
<div>  
<header>  
<footer>  
<h1>...<h6>
```

Borders, background images and background color

While content will wrap around a floated element, border, background image and background color will extend underneath.

"float: left" element with background-colored block level element below



If you want to avoid this effect, make the gray box float to the opposite side by applying float:right

How to stop the floating

Say you wanted the first image to float so that the paragraph wraps around it, but you want the floating to stop now. You **don't** want the following object to come up in that leftover space. What do you do?

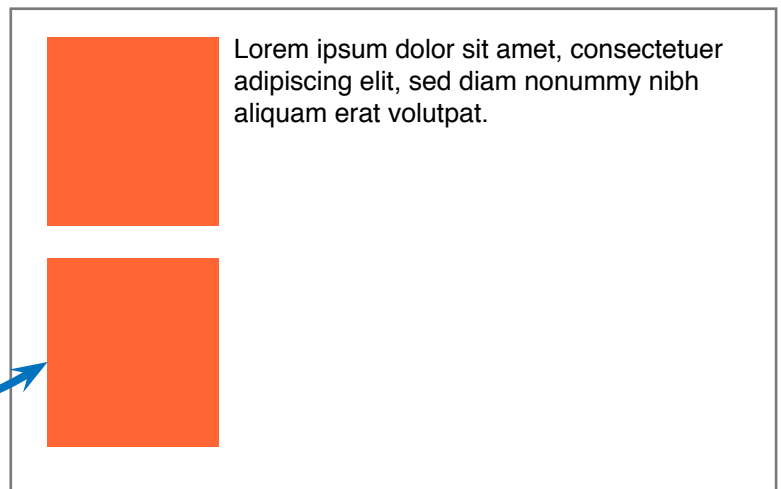
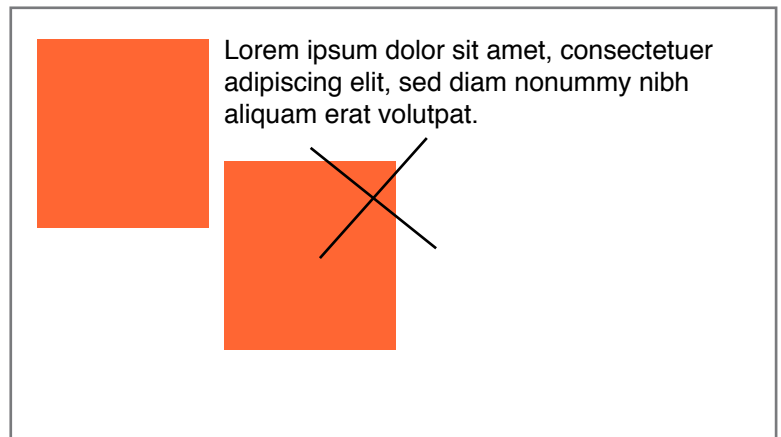
This is what the clear property is for. Clear just means to stop the floating and make the object go back down to where it was.

On that 2nd box, write **clear: left**;
(You'll want to give that box a class first in the HTML, then write clear:left in the CSS.)

The options are:

"clear: left", "clear: right" or "clear: both".

You pick which one to use based off of which direction the float was. If there was something floating left AND something floating right, you can stop the floating with "clear:both".

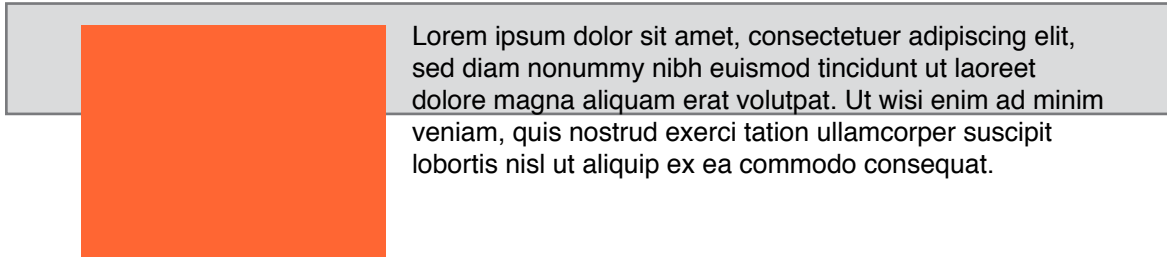


clear:left applied here

The parent that can't wrap around

You're probably already noticing that the container often has trouble continuing to wrap around the objects once something inside it is floating.

Pssst: the below layout is what you are going to do in your project.



We have a simple fix for making the container (or the “parent”) wrap around its floated children. It’s called the “clearfix”. I would have called it the “hold onto your kids!” fix, but I guess I wasn’t consulted.

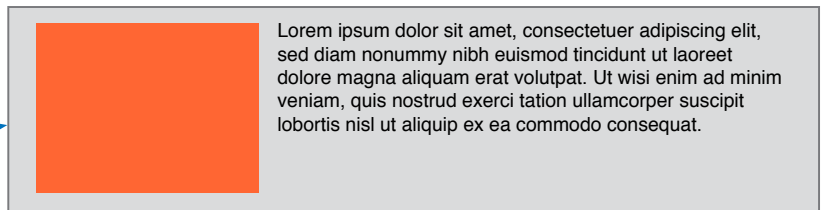
It’s just a little chunk of code that we all just copy and paste (no one really tries to understand it—don’t worry). We apply that class to the wonky container and it makes the container, well, be a better parent and hold onto the kids.

```
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```



First, write all of this in your CSS. The top of the page is a good place to put it.

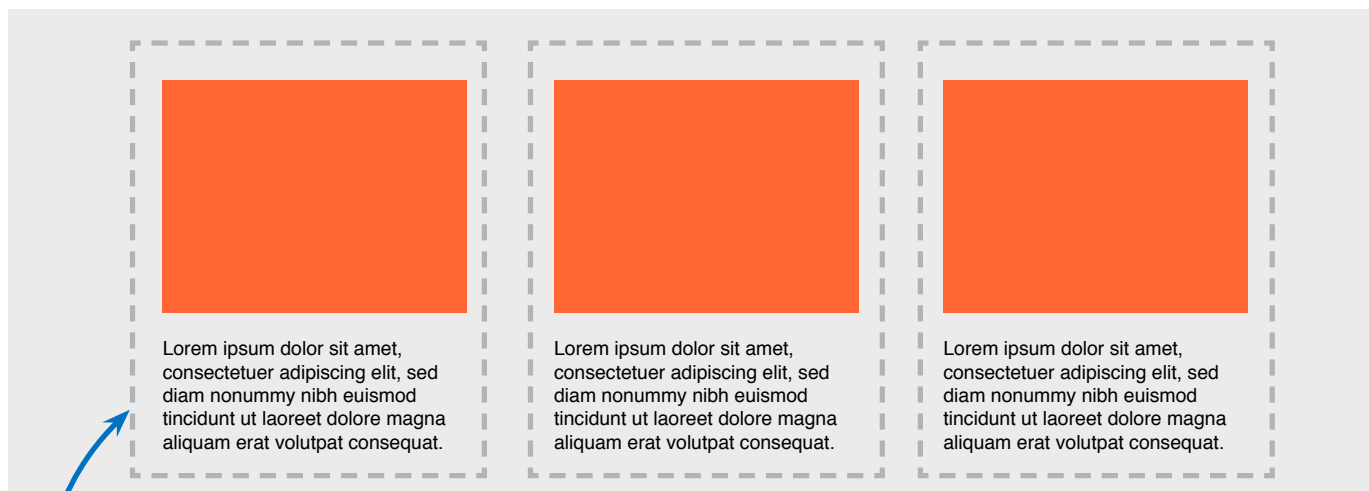
Then on the container (in the HTML) add `<div class="clearfix">...`



A common design and how to do it

Some of you have designed something like this. Here is how you'd code it"

Note: you do NOT have to do this layout. You are allowed to simplify it by putting an image on the left and it's text on the right.



First, make sure you've placing a `<p>` tag or heading tag around each chunk of text. This way the text doesn't site right next to the image, but goes down to the next line.

Then, in the HTML, place a new div around each set of image + text, which is where I've placed dashed lines. Give each div a class, like "columns", for instance.

Then, in the CSS, apply each a `float:left` to ".columns" or `display:inline`. (Both do the same thing.)

Spread out the columns by giving them some margins.