THE BOX MODEL
Works great...

- Don’t apply left, or right padding to elements with defined widths
- Floated elements with borders are wider than you think they are.
- Old IE has its own idea of a box model that is unlike any other with many bugs
CSS3 can normalize

.selector {
  box-sizing: border-box;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
}

Gets the box model to be more like IE6.
This is better...

- Borders, and horizontal paddings all contained within the width.
- Supported by a majority of browsers, including the lovely IE8.
FLOATS
A basic float

.float {
  float: left;
  margin-right: 20px;
  margin-bottom: 20px;
}

.content
Another Common float

.float

.content

.float {
  float: left;
  width: 150px;
  margin-right: 20px;
  margin-bottom: 20px;
}

.content {
  margin-left: 170px;
}
Some caveats...

- Float without a width and the width of the interior content will be the width.
- Long text within a float requires that you set a width. Otherwise it’s 100%.
- A tall floated item must be cleared, or you will end up with floats within floats.
Uncleared floats

.content

.float

.float {
    float: right;
    margin-left: 20px;
    margin-bottom: 20px;
}
Clear your floats...

- No need to use extra markup to clear your floats. Self-clearing is the only way.
Self Clearing Floats

.selector {
overflow: auto;
*zoom: 1;
}

.selector:after {
content: ".";
display: block;
height: 0;
clear: both;
visibility: hidden;
}
Don’t use clearfix as a mixin in SASS or LESS. Use it as a class name and save.
Crazy floats

.float

.float {
    float: left;
    margin-left: -50px;
    margin-right: 20px;
    margin-bottom: 20px;
}
Positioning
Things to remember...

- Absolutely positioned elements are positioned relative to the nearest relatively positioned element, or the body element if none.

- Fixed position elements are positioned relative to the body element every time.
Fixed Positioning

.parent {
  position: relative;
}

.child {
  position: fixed;
  top: 0;
  right: 0;
  left: 0;
}
Absolute Positioning

```css
.parent {
    position: relative;
}

.child {
    position: absolute;
    top: 0;
    left: 0;
}
```
Z-index

- Z-index defines the layer that you want an element to appear on.
- IE6 & 7 have issues with z-index. Everyone else does it properly.
Using z-index

(selector:.selector:before {
  position: absolute;
  z-index: -1;
  top: -20px;
  opacity: 50;
  font-size: 75%;
  content: “Z-INDEX”
})

(selector:.selector:after {
  position: absolute;
  z-index: -2;
  top: -40px;
  opacity: 25;
  font-size: 50%;
  content: “Z-INDEX”
})
Fin.