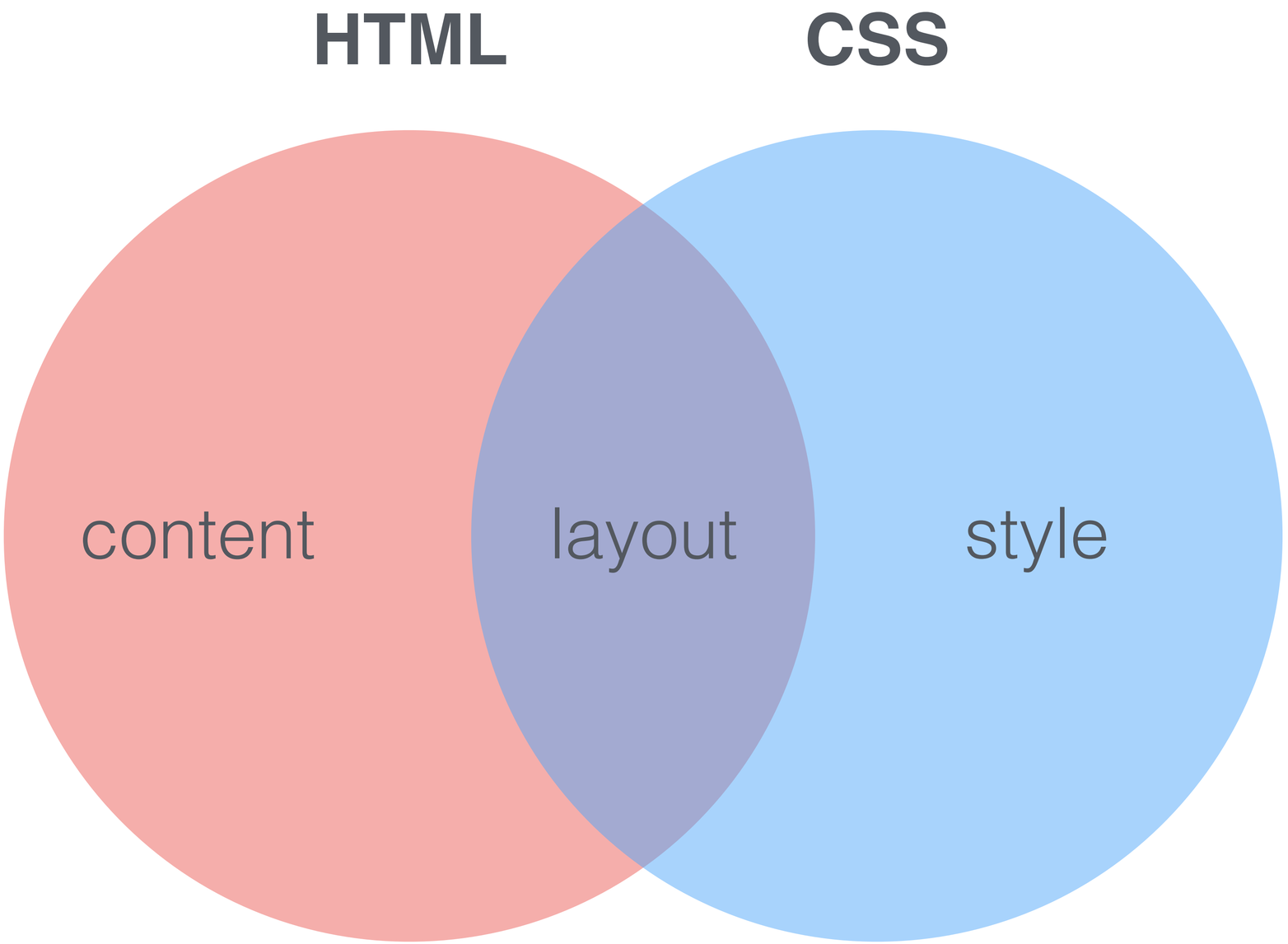
HTML & CSS

Layout laid out


HTML
CSS
content
layout
style

I Am Devloper
@iamdevloper

Following

CSS is easy. It's like riding a bike, which is on fire and the ground is on fire and everything is on fire because it is hell.

I Am Devloper
@iamdevloper

Following

"Learn CSS in 24 Hours"

After 3 hours, you've learnt all of CSS.

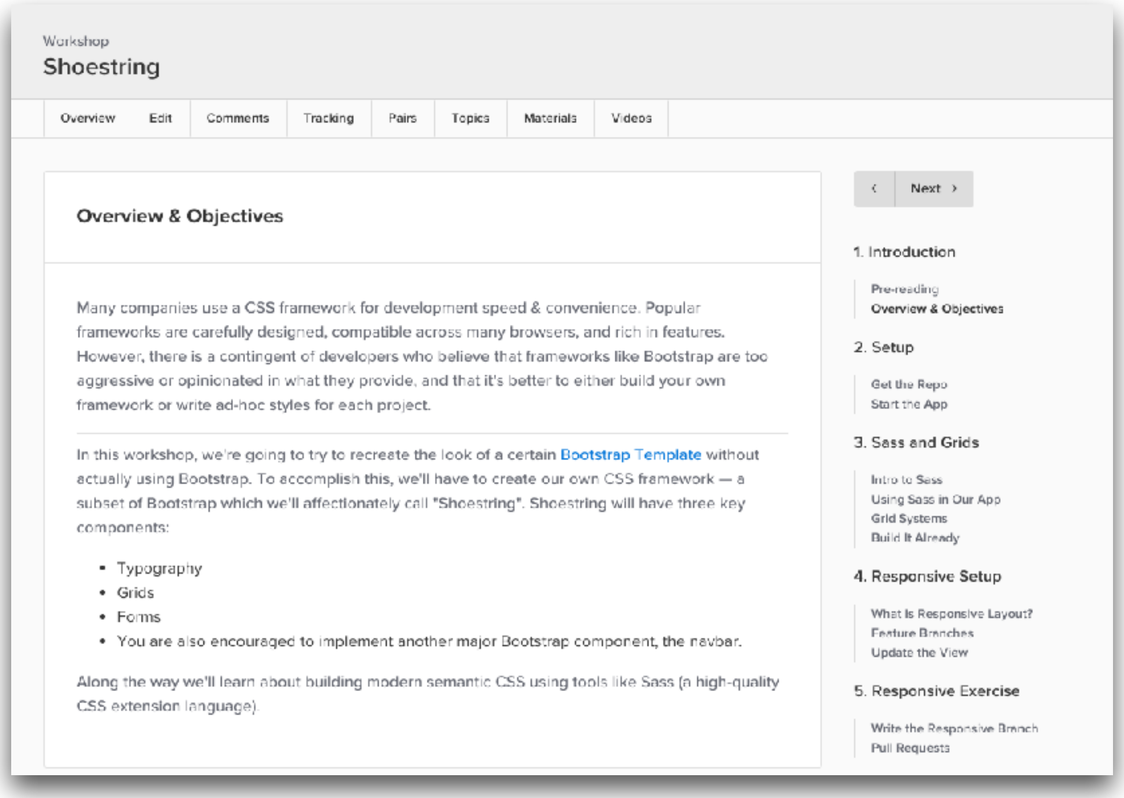
Hours 3 to eternity are spent trying to vertically center stuff.

WHY IS CSS IMPORTANT?

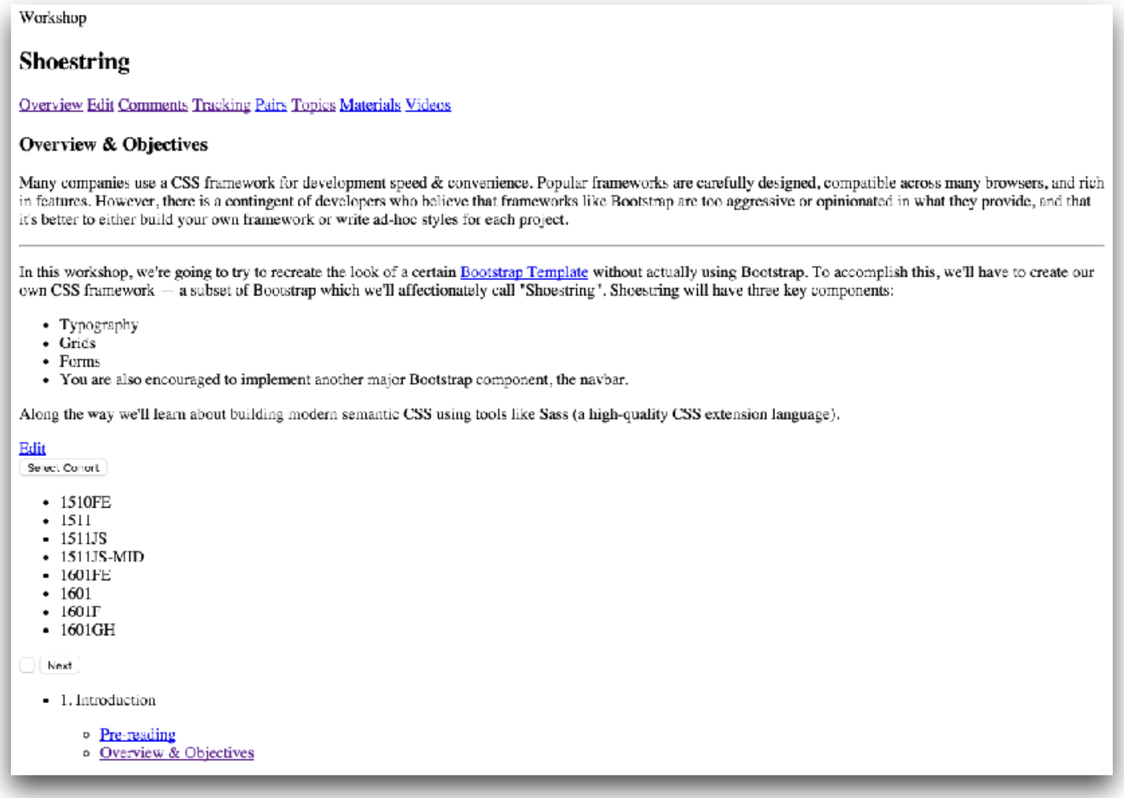
CSS IS IMPORTANT

- The web needs to look nice.
- It’s the only game in town.
- Be a triple threat.

WITH CSS



WITHOUT CSS



TERMS


selector
property
keyword
declaration
rule
article li > a:hover {
border: 1px solid red;
font-style: italic;
}

RULE EXAMPLE

SELECTORS

tag	`input`
class	`.btn`
id	`#upload`
attribute	`[type="file"]`
pseudo-element	`::after`
pseudo-class	`:hover`
*	*

COMBINATORS

descendent	(whitespace)	div li
child	>	div > li
next sibling	+	img + span
later sibling	~	img ~ span

BEWARE!

`tag.class`	element with BOTH `tag` AND `.class`
`tag .class`	element that is a descendent of `tag` and has `.class`
`tag, .class`	element with EITHER `tag` OR `.class`

CASCADING STYLE SHEETS

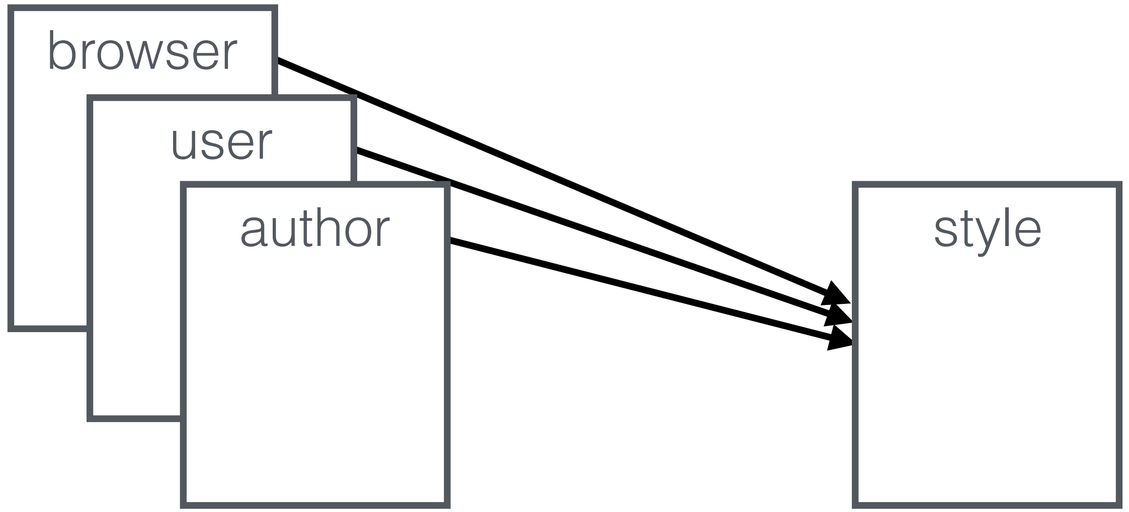
CASCADING

In ~1994... *CSS had one feature that distinguished it from all the [competing style languages]: it took into account that on the Web the style of a document couldn't be designed by either the author or the reader on their own, but that their wishes had to be combined, or "cascaded," in some way.*

CASCADING STYLE SHEETS, DESIGNING FOR THE WEB, BY HÅKON WIUM LIE AND BERT BOS (1999) - CHAPTER 20

CASCADING

An element's style is a merge of every rule whose selector matches



index.html

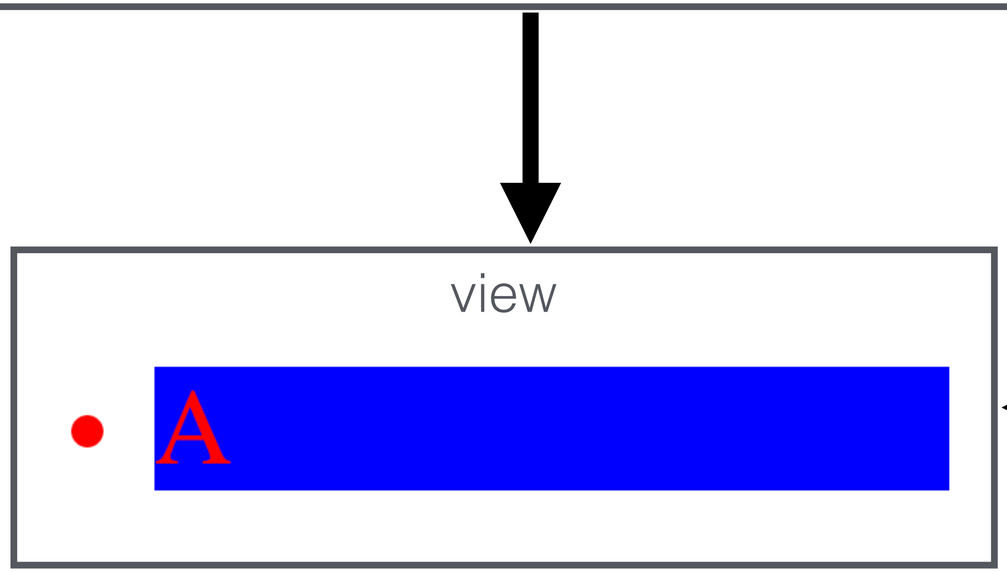
```
<head>
  <link rel="stylesheet" href="styles-B.css" />
  <link rel="stylesheet" href="styles-A.css" />
</head>
<body>
  <ul>
    <li style="background-color:blue;">A</li>
  </ul>
</body>
```

styles-A.css

```
li {
  color: red;
}
```

styles-B.css

```
li {
  font-size: 40px;
}
```



style

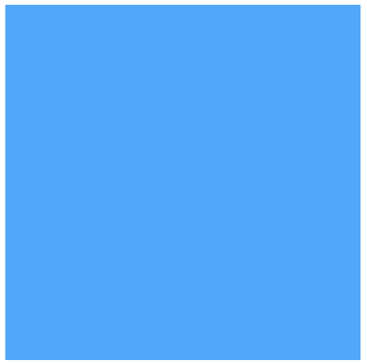
```
element.style {
  background-color: blue;
}

li {                                    styles-A.css:1
  color: red;
}

li {                                    styles-B.css:1
  font-size: 40px;
}

li {                                    user agent stylesheet
  display: list-item;
  text-align: -webkit-match-parent;
}
```

What happens when declarations conflict?

```
<div id="thing"></div>
```

```
div {
  background: red;
}
```


```
#thing {
  background: blue;
}
```



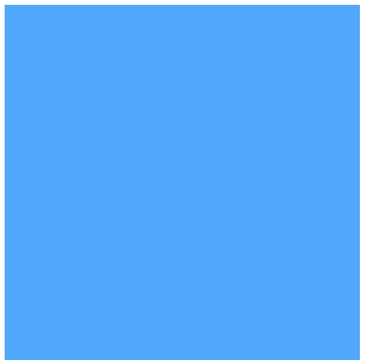
```
<div class="foo"></div>
```

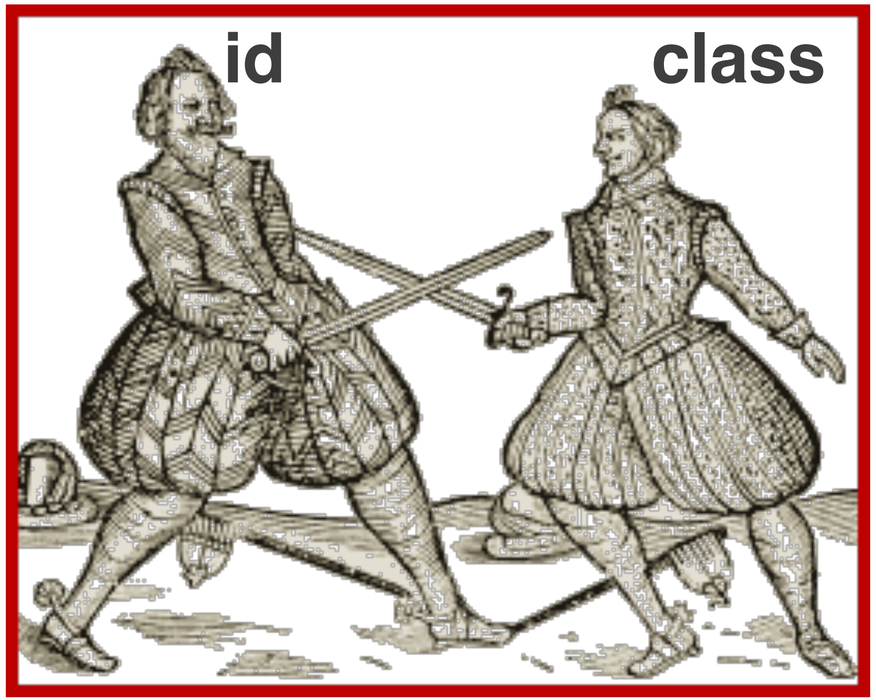
```
div {
  background: red;
}
```


```
.foo {
  background: green;
}
```

```
<div id="thing" class="foo"></div>
```

```
#thing {
  background: blue;
}
```

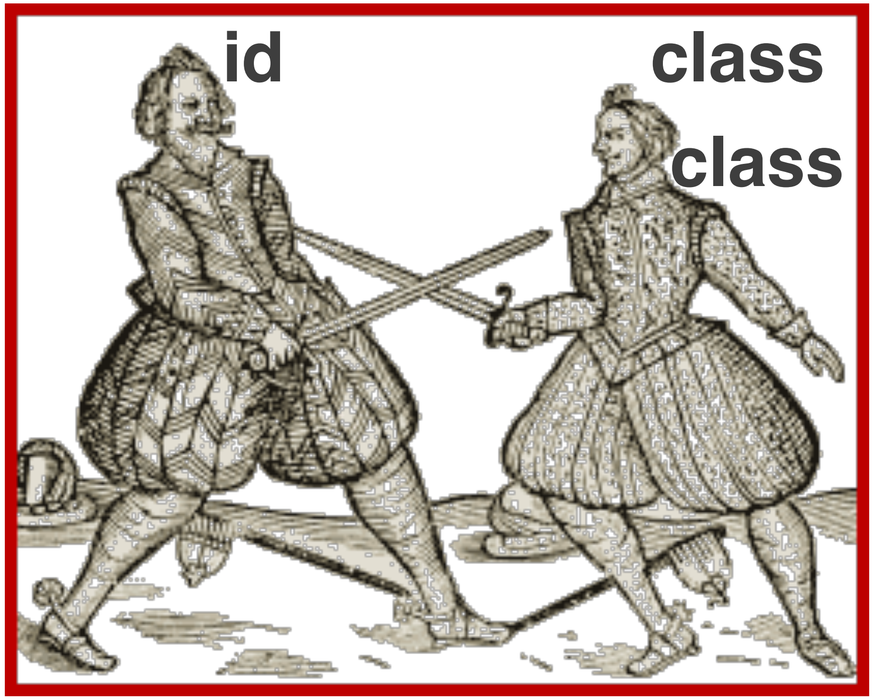





```
.foo {
  background: green;
}
```

```
<div id="thing" class="foo bar"></div>
```

```
#thing {
  background: blue;
}
```




```
.foo.bar {
  background: green;
}
```

```
<div class="outer">
  <div id="thing" class="foo" style="background:orange;"></div>
</div>
```

```
#thing {
  background: blue;
}
```

```
.outer .foo {
  background: green;
}
```

```
<div class="outer">
  <div id="thing" class="foo" style="background:orange;"></div>
</div>
```

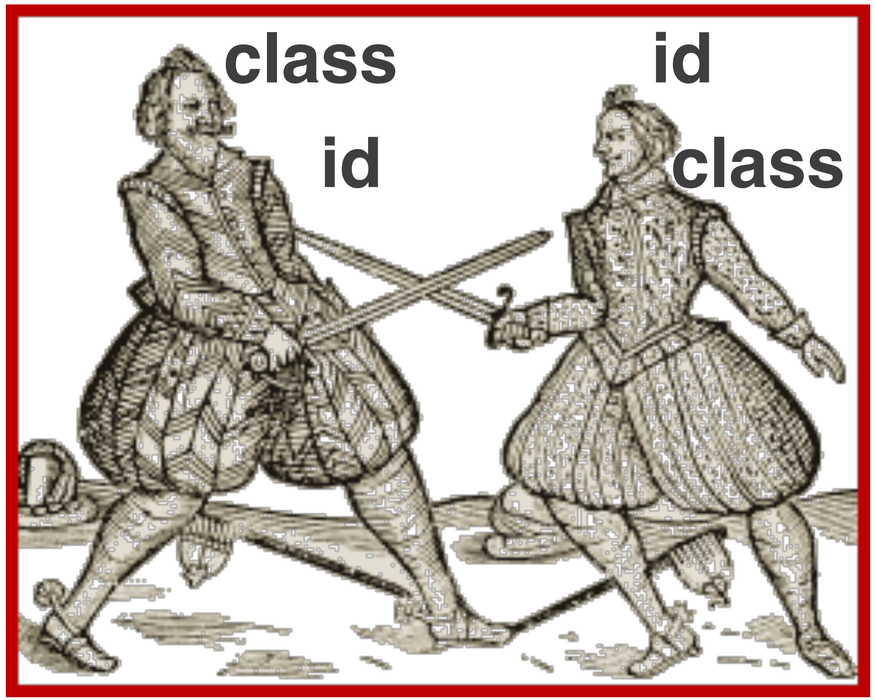
```
div {
  background: red !important;
}
```

```
.outer .foo {
  background: green;
}
```

depends on which rule gets defined *last*

```
<div id="profile" class="outer">
  <div id="thing" class="foo"></div>
</div>
```

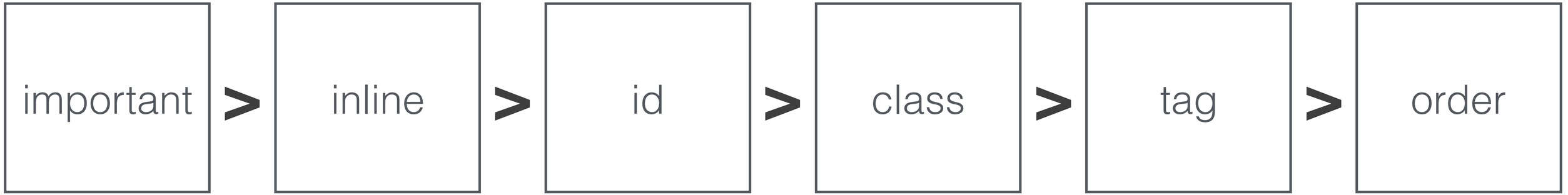
```
.outer #thing {
  background: purple;
}
```




```
#profile .foo {
  background: brown;
}
```

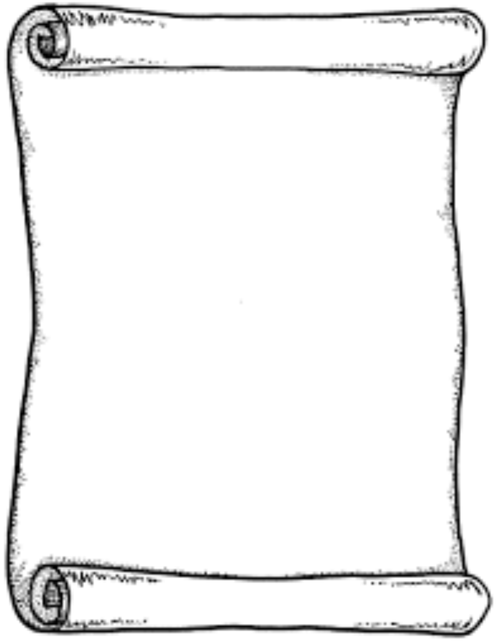
THAT WAS CSS SPECIFICITY

DECLARATION SPECIFICITY

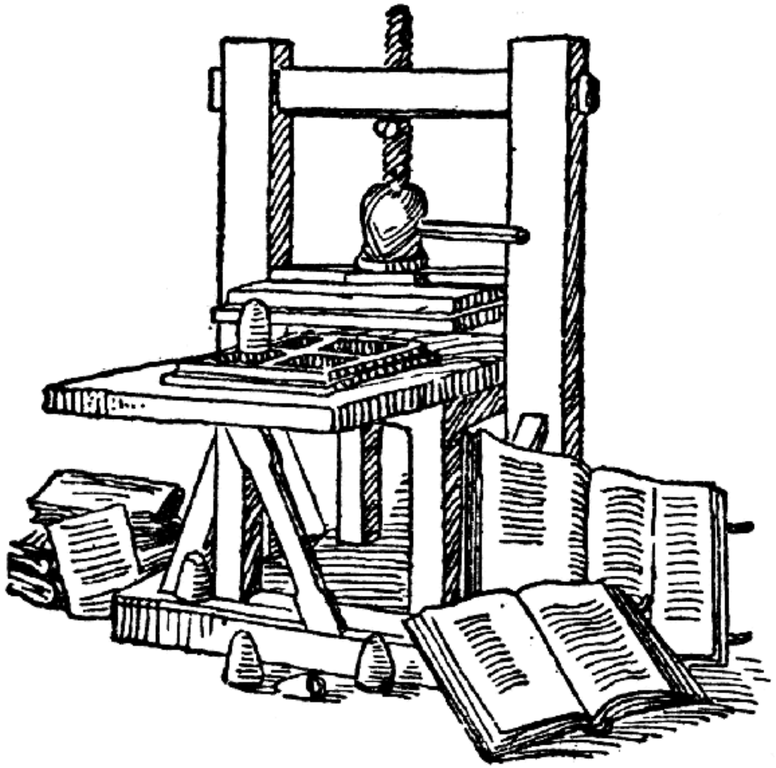


Combinators don’t affect specificity!

LAYOUT



HTTP://WWW.CLIPARTBEST.COM/CLIPART-9I4O55P6T



HTTP://ETC.USF.EDU/CLIPART/44800/44880/44880_GUTEN_PRESS_LG.GIF

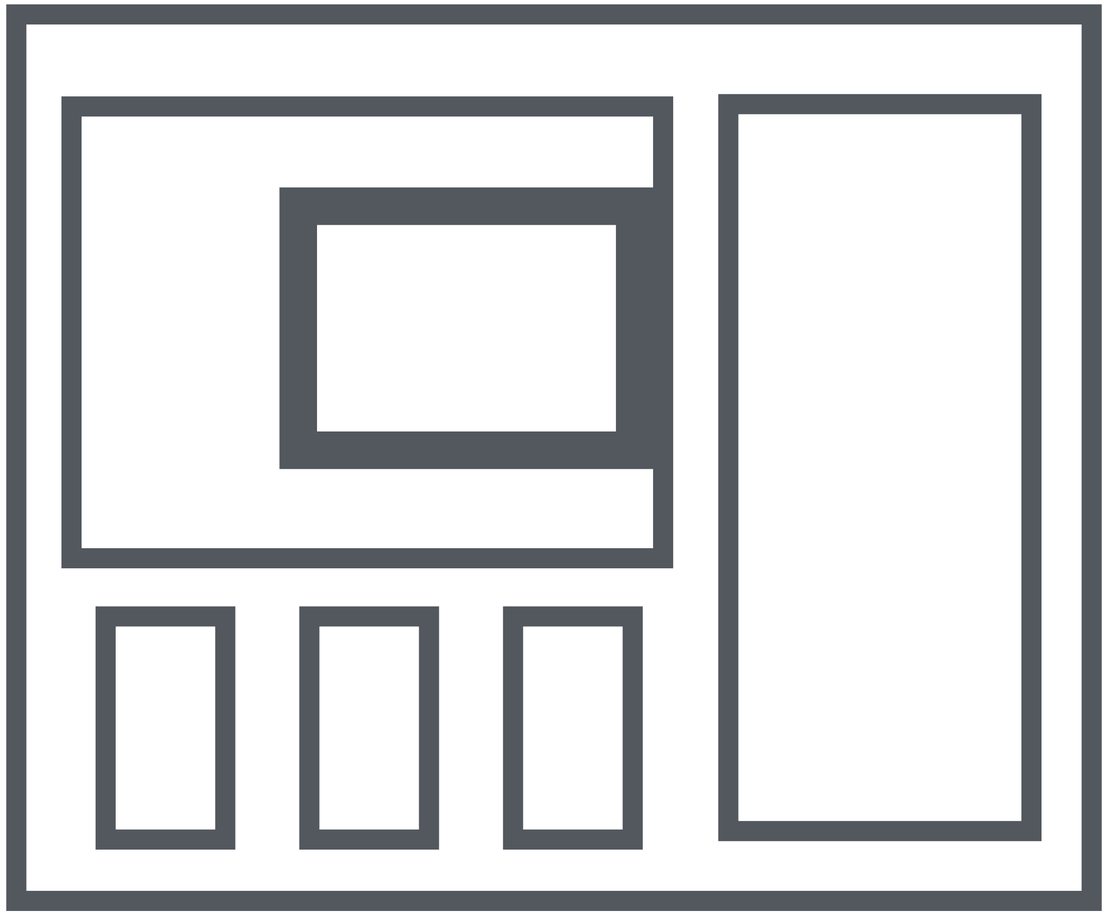
HTTP://ASSETS.UXBOOTH.COM/UPLOADS/2009/11/MYSPACE.PNG

THE BOX MODEL

“Everything on the DOM is a rectangle”

- JOHN AND TOM

BOX MODEL



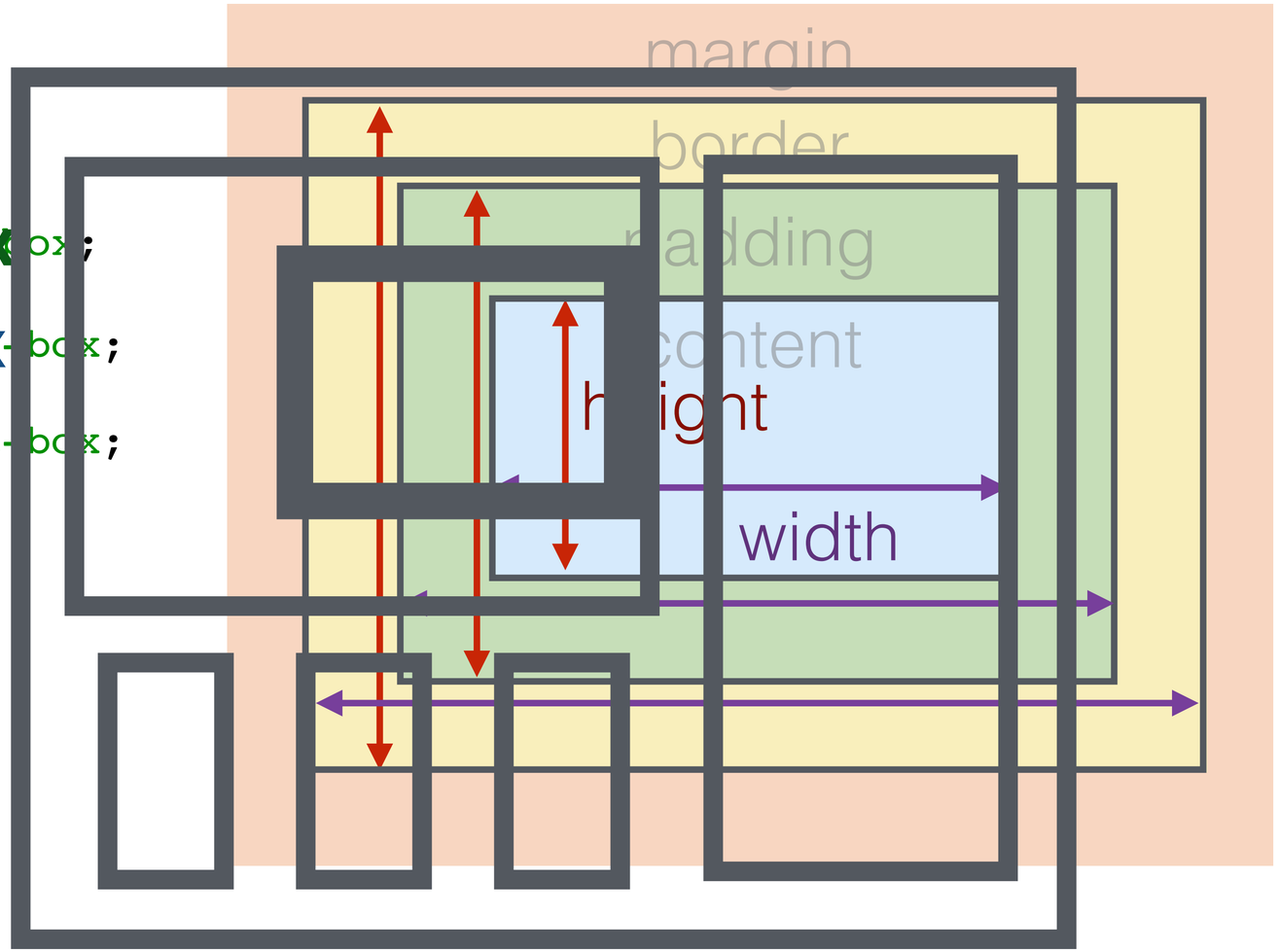
BOX MODEL

```
{
  box-sizing: border-box;
}
{
  box-sizing: padding-box;
}
{
  box-sizing: content-box;
}
```

border box

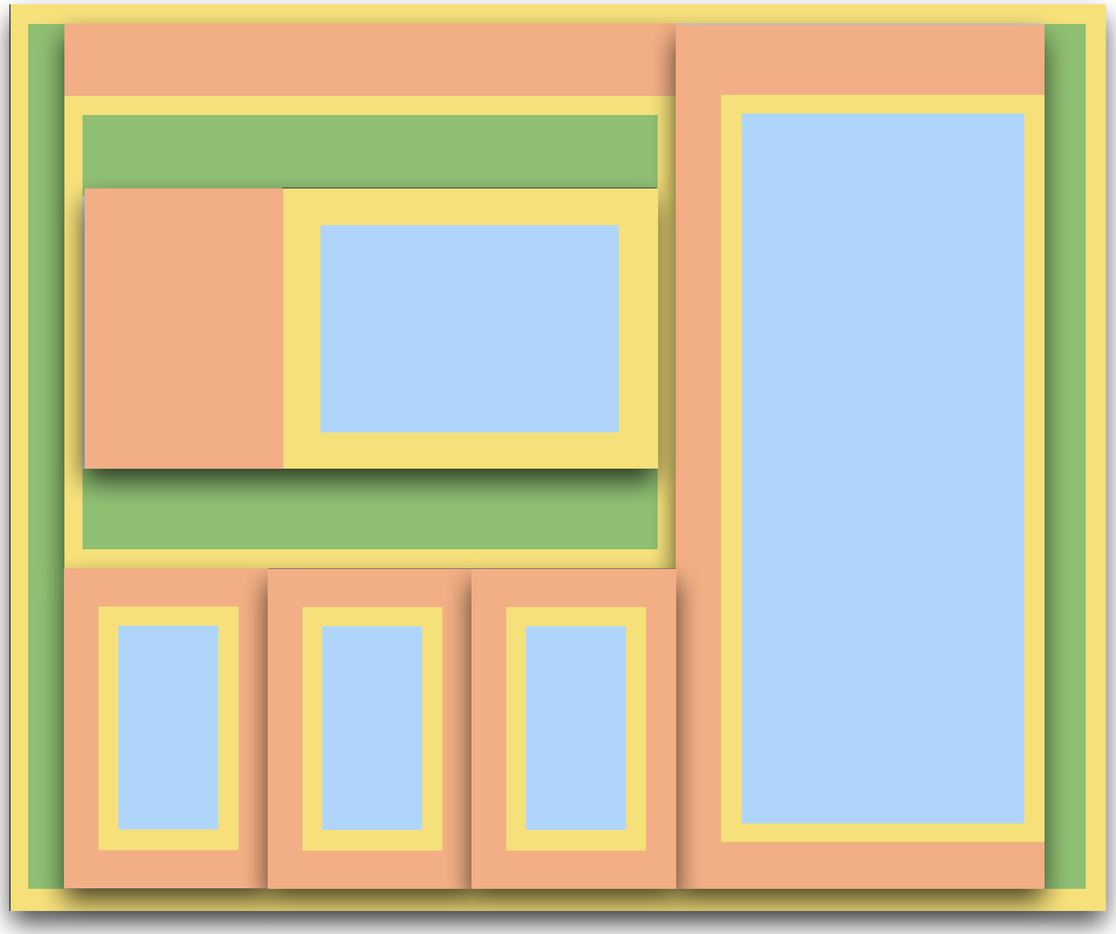
padding box

content box



BOX MODEL

fractal!



BLOCK vs INLINE vs INLINE-BLOCK

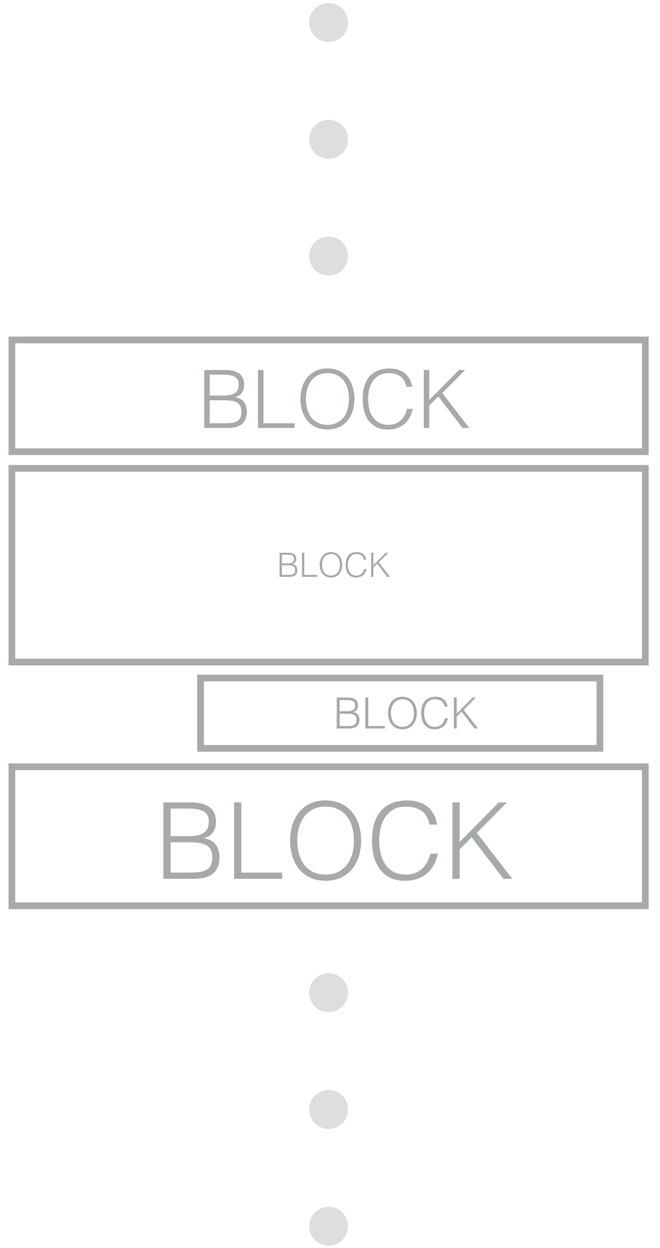
-
- <h1>, <h2>, etc.
- <p>
- <form>
- <header>, <footer>, <main>, <section>, <nav>

- <a>
-
- ,

- <img>
- <input>
- <textarea>

BLOCK-LEVEL

- By default will try to **clear** their own line
- Default width is 100% of parent
- Default height will expand to fit all children
- Can have margins on all sides
- Can set height and width

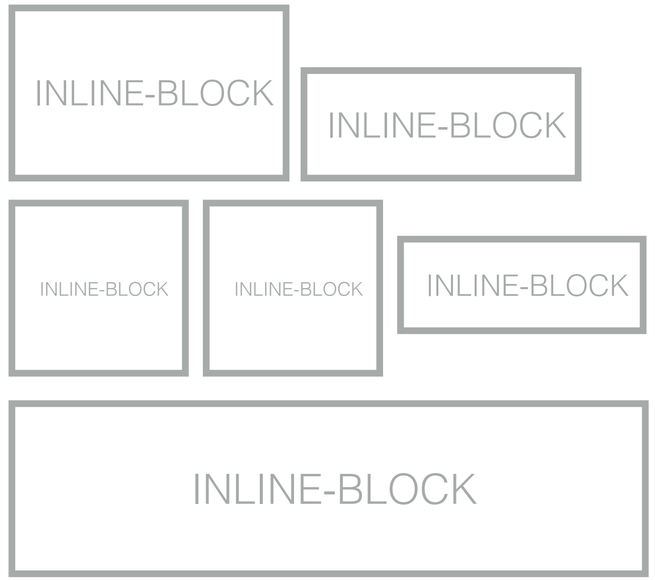


INLINE-LEVEL

- Flows with content (does **not** clear line)
- Ignores top and bottom margins
- Height and width fit content
- Cannot set height or width

INLINE-BLOCK

- Flows with content (does **not** clear line)
- Default width will expand to fit all children
- Default height will expand to fit all children
- Can have margins on all sides
- Can set height and width



FLOATS

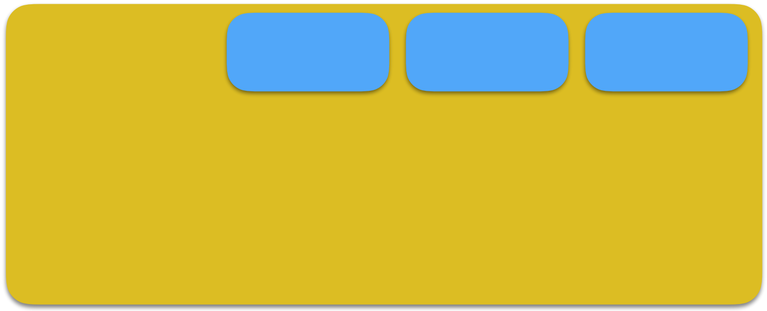
"I understand floats."

–Liar McPantsonfire

```
<body>
  <div>
    <div style="width:50px;">…</div>
    <div style="width:50px;">…</div>
    <div style="width:50px;">…</div>
  </div>
</body>
```

```
<body>
  <div>
    <div style="width:50px; float:right;">…</div>
    <div style="width:50px; float:right;">…</div
    <div style="width:50px; float:right;">…</div>
  </div>
</body>
```

```
<body>
  <div style=overflow:auto;>
    <div style="width:50px; float:right;">…</div>
    <div style="width:50px; float:right;">…</div
    <div style="width:50px; float:right;">…</div>
  </div>
</body>
```



'CLEARFIX' HACK?

- putting *overflow: auto* on the parent will work a lot of the time
- But it can have unpredictable effects, so we use something called the ‘clearfix hack’
- give the parent element a class of *clearfix*
- In your CSS, write the following clearfix rule:

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


CSS
Global
HD


SYNTACTICALLY
AWESOME
STYLESHEETS
SASSY
STYLESHEETS
SHADING

SCSS

- nesting
- variables
- loops
- functions
- modules

HOW DOES SCSS WORK?

SCSS COMPILES TO CSS

NESTING

SCSS

```scss
article {
    border: 1px solid red;
    li {
        background: gray;
    }
}
```

CSS

```css
article {
    border: 1px solid red;
}
article li {
    background: gray;
}
```

VARIABLES

SCSS

```scss
$deep-red: #990000;
a {
    color: $deep-red;
}
.warning {
    border-color: $deep-red;
}
```

CSS

```css
a {
    color: #990000;
}
.warning {
    border-color: #990000;
}
```

LOOPS

SCSS

```
@for $i from 1 through 3 {
    size#{$i} {
        width: $i * 10px;
    }
}
```

CSS

```
size1 {
    width: 10px;
}
size2 {
    width: 20px;
}
size3 {
    width: 30px;
}
```

MIXINS

SCSS

```scss
@mixin border-radius ($r) {
    -webkit-border-radius: $r;
    -moz-border-radius: $r;
    border-radius: $r;
}

.thing {
    @include border-radius(10px);
}
```

CSS

```css
.thing {
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    border-radius: 10px;
}
```

MODULES

SCSS

```scss
/* pulls in normalize.scss */
@import 'normalize';
```

CSS

```css
/* ... */
/**
 * 1. Set default font family to sans-serif
 * 2. Prevent iOS text size adjust after orientation change, without disabling
 *    user zoom.
 */
html {
    font-family: sans-serif;
    /* 1 */
    -ms-text-size-adjust: 100%;
    /* 2 */
    -webkit-text-size-adjust: 100%;
    /* 2 */ }
/**
 * Remove default margin.
 */
body {
    margin: 0; }
/* ==========================================================================
   Links
   ========================================================================== */
/**
 * Remove the gray background color from active links in IE 10.
 */
a {
    background: transparent; }
/* ... */
```