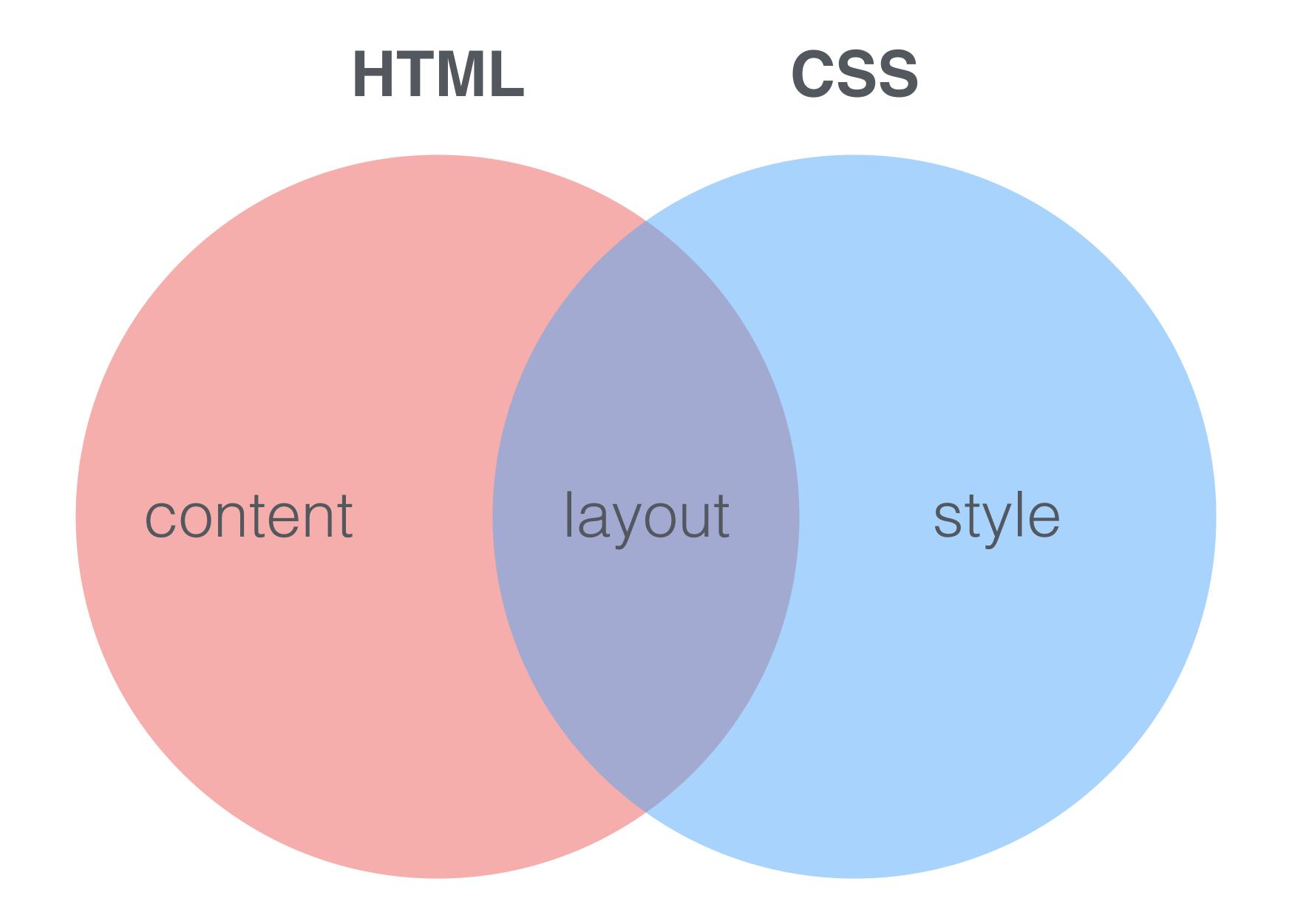
# HTML & CSS

Layout laid out

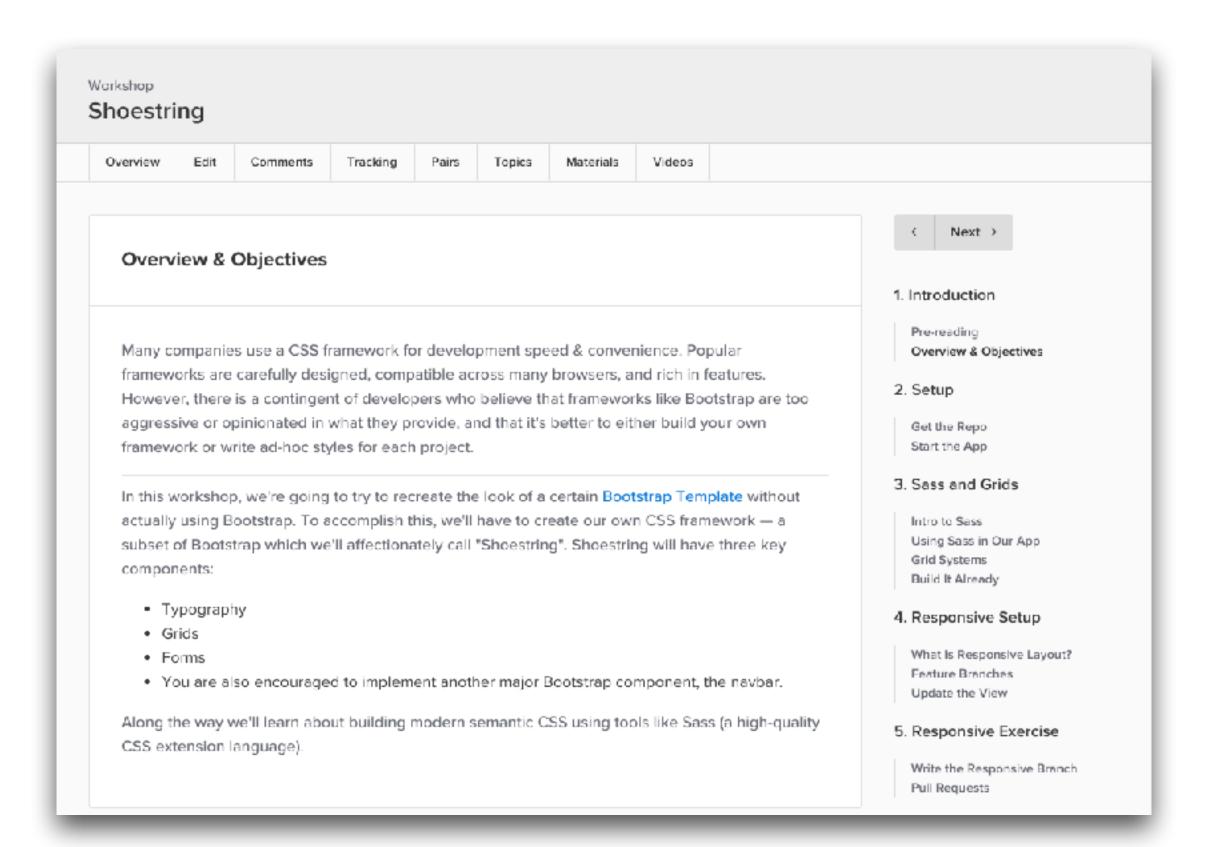


# WHY IS CSS IMPORTANT?

# CSS IS IMPORTANT

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

# WITH CSS



# WITHOUT CSS

#### Workshop Shoestring Overview Edit Comments Tracking Pairs Topics Materials Videos Overview & Objectives Many companies use a CSS framework for development speed & convenience. Popular frameworks are carefully designed, compatible across many browsers, and rich in features. However, there is a contingent of developers who believe that frameworks like Bootstrap are too aggressive or opinionated in what they provide, and that it's better to either build your own framework or write ad-hoc styles for each project. In this workshop, we're going to try to recreate the look of a certain Bootstrap Template without actually using Bootstrap. To accomplish this, we'll have to create our own CSS framework — a subset of Bootstrap which we'll affectionately call "Shoestring". Shoestring will have three key components: Typography Grids Forms • You are also encouraged to implement another major Bootstrap component, the navbar. Along the way we'll learn about building modern semantic CSS using tools like Sass (a high-quality CSS extension language). Select Cohort 1510FE 1511 1511JS 1511JS-MID 1601FE 1601 1601F 1601GH Next 1. Introduction Pre-reading Overview & Objectives

HTML & CSS

# TERMS

```
declaration border: 1px solid red;

font-style: italic;
}
```

### RULE EXAMPLE

```
apply these styles—border: 1px solid red;
font-style: italic;
}

to any elements matching this selector
even for any future changes declarative!
```

# SELECTORS

```
tag input
```

class .btn

id #upload

attribute [type="file"]

pseudo-element ::after

pseudo-class : hover

\* \*

# COMBINATORS

descendent (whitespace)

child >

next sibling +

later sibling ~

### BEWARE!

tag.class element with BOTH tag AND .class
tag .class element with .class whose ANCESTOR matches tag
tag, .class element with EITHER tag OR .class
tag+.class element with .class whose left SIBLING matches tag

# CASCADING STYLE SHEETS

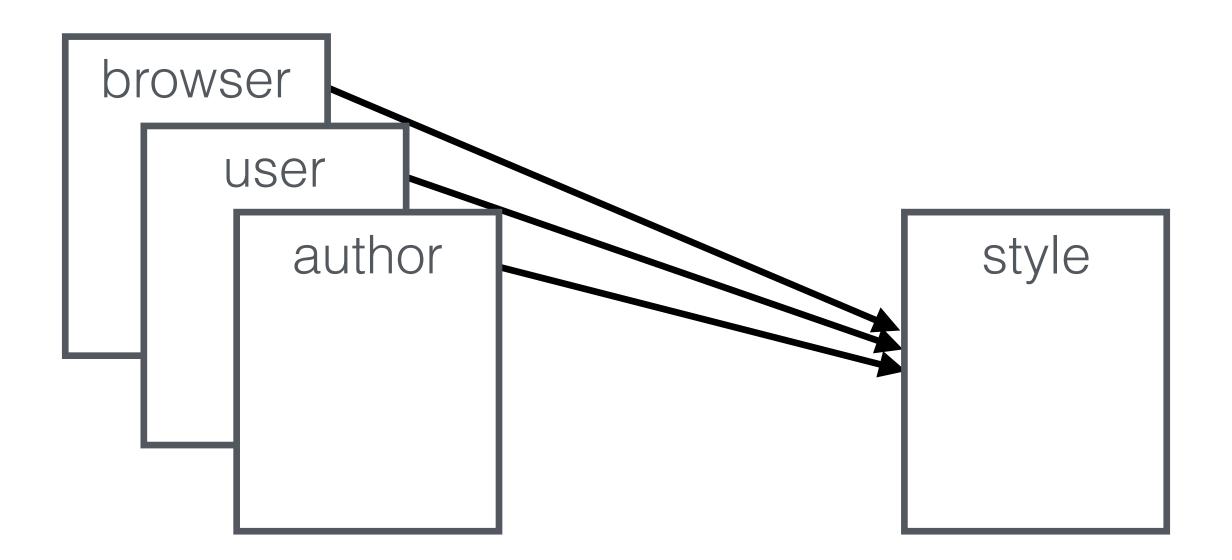
# CASCADING

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



```
styles-A.css
                                                                           styles-B.css
                   index.html
<head>
                                                                        li {
                                                     li {
 <link rel="stylesheet" href="styles-B.css" />
                                                                          font-size: 40px;
                                                       color: red;
  <link rel="stylesheet" href="styles-A.css" />
</head>
<body>
 <u1>
    style="background-color:blue;">A
  </body>
                                                                     style
                                                     element.style {
                                                       styles-A.css:1
                                                       color:  red;
                     view
                                                                               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>

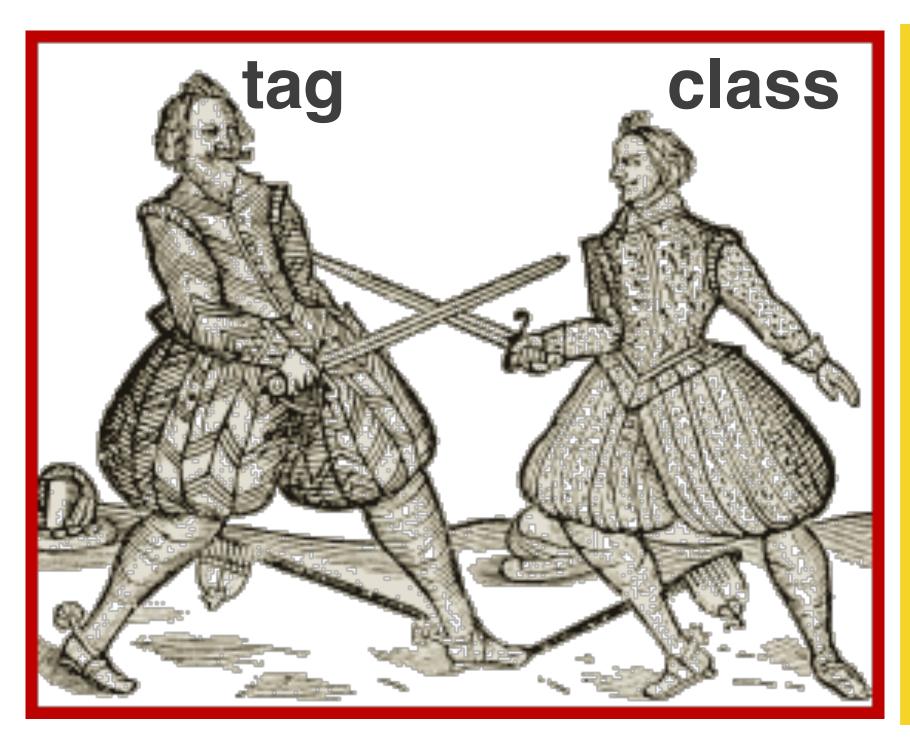
```
div {
  background: red;
}
```



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

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

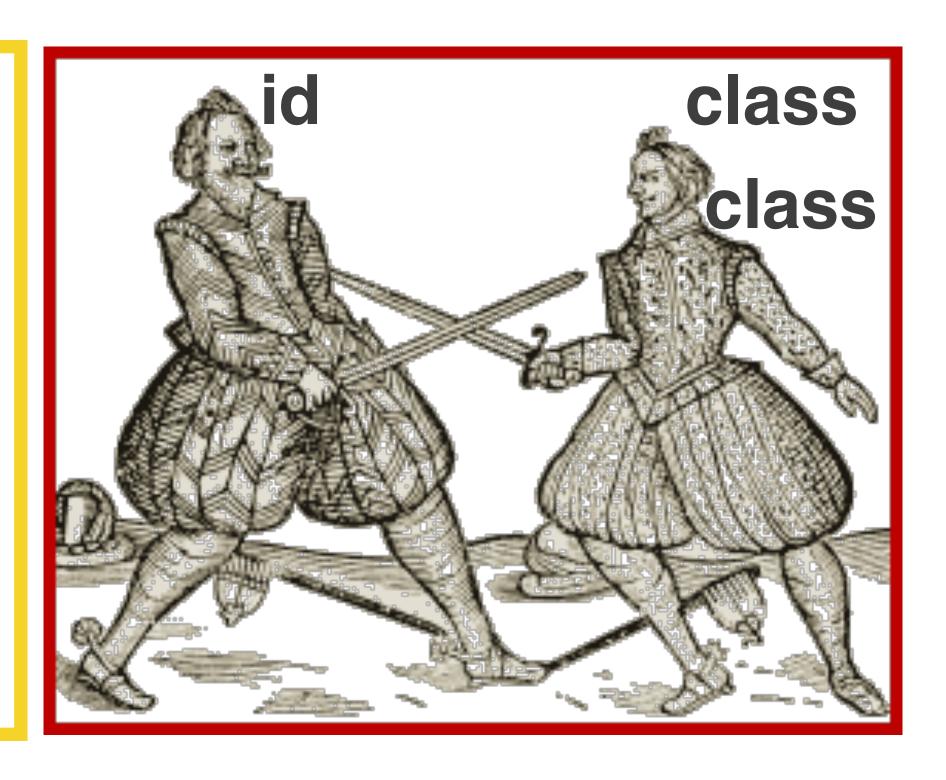
```
div {
  background: red;
}
```



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

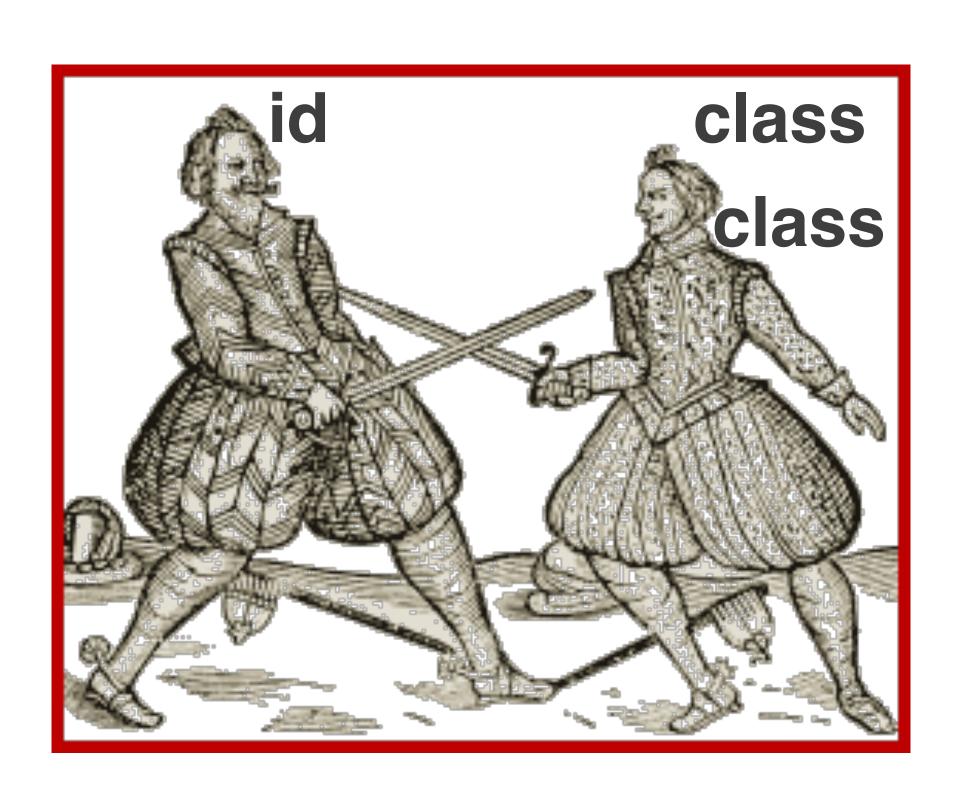
#### <div id="thing" class="foo bar"></div>

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



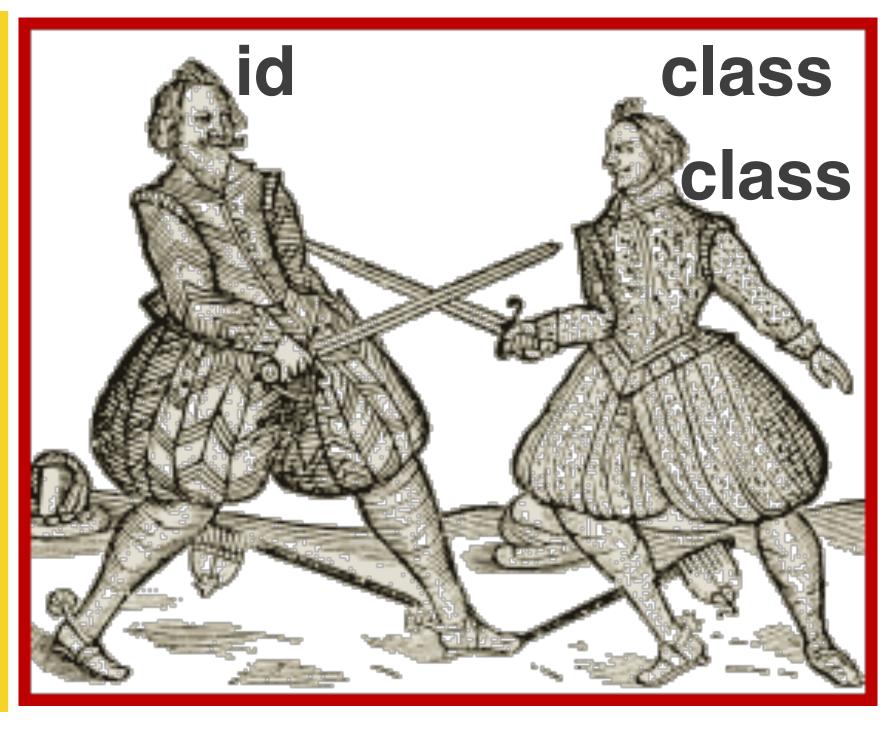
```
foo.bar {
  background: green;
}
```

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

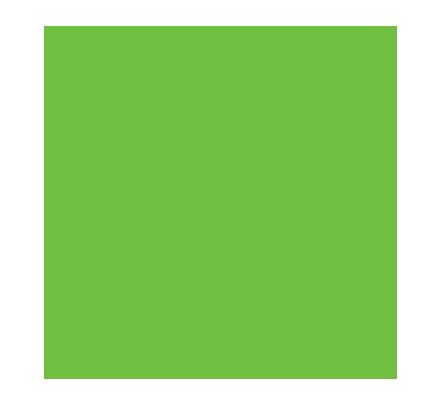


```
•outer •foo {
  background: green;
```

```
#thing {
  background: red !important;
}
```



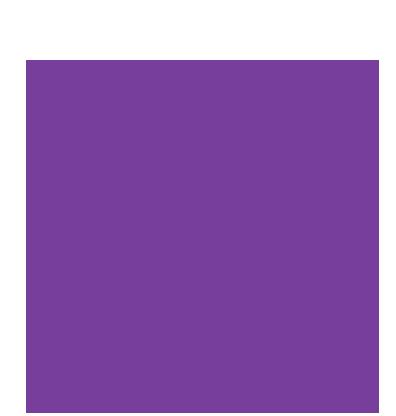
```
.outer .foo {
  background: green;
}
```

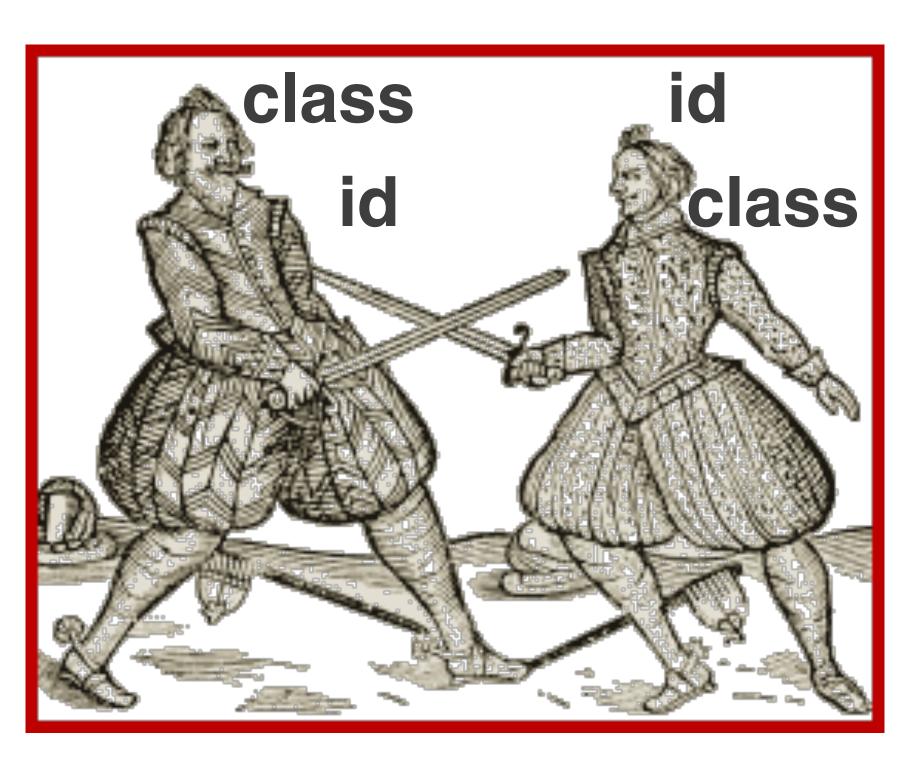


### depends on which rule gets defined last

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

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

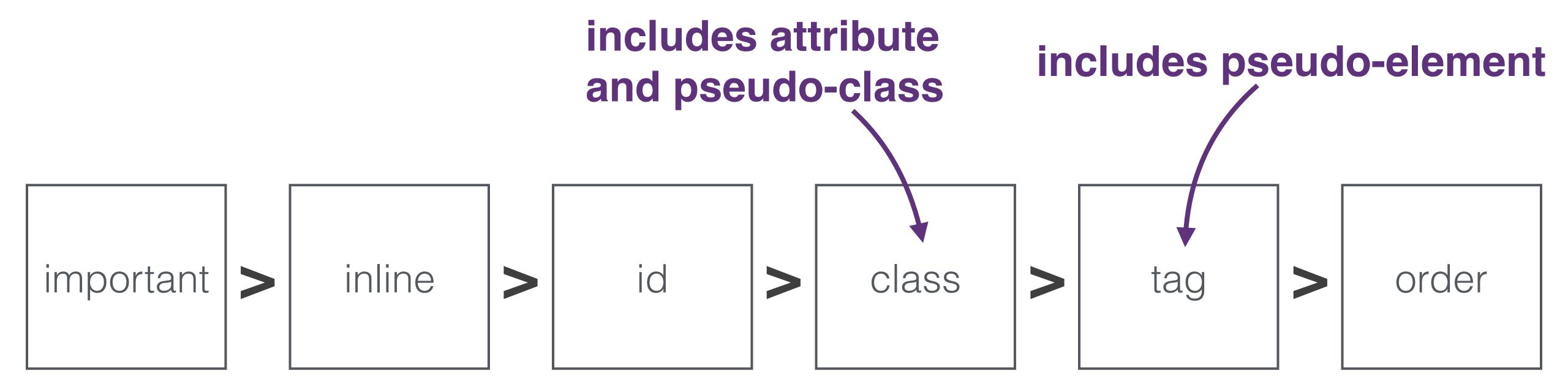




```
#profile .foo {
  background: brown;
}
```

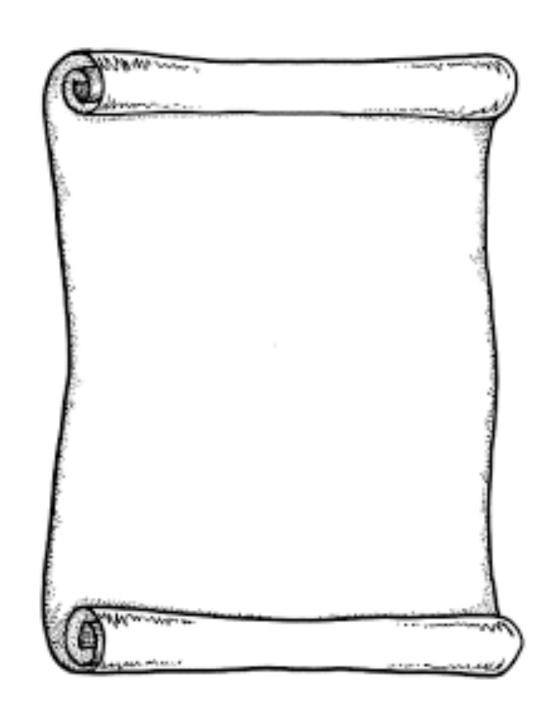
# THAT WAS CSS SPECIFICITY

# DECLARATION SPECIFICITY

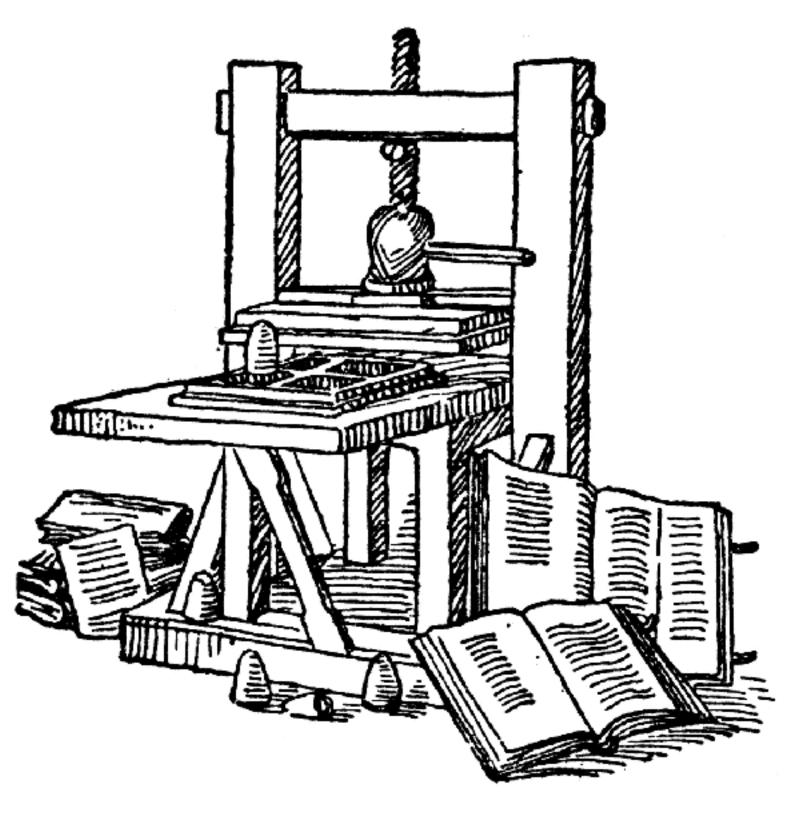


Combinators don't affect specificity!

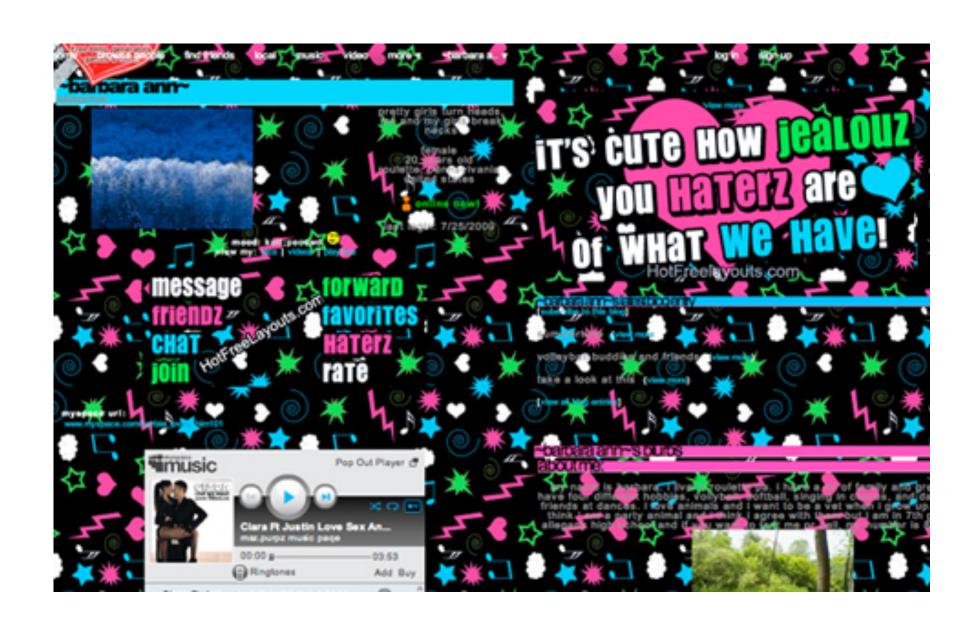
# LAYOUT







HTTP://ETC.USF.EDU/CLIPART/44800/44880/44880 GUTEN PRESS LG.GIF



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



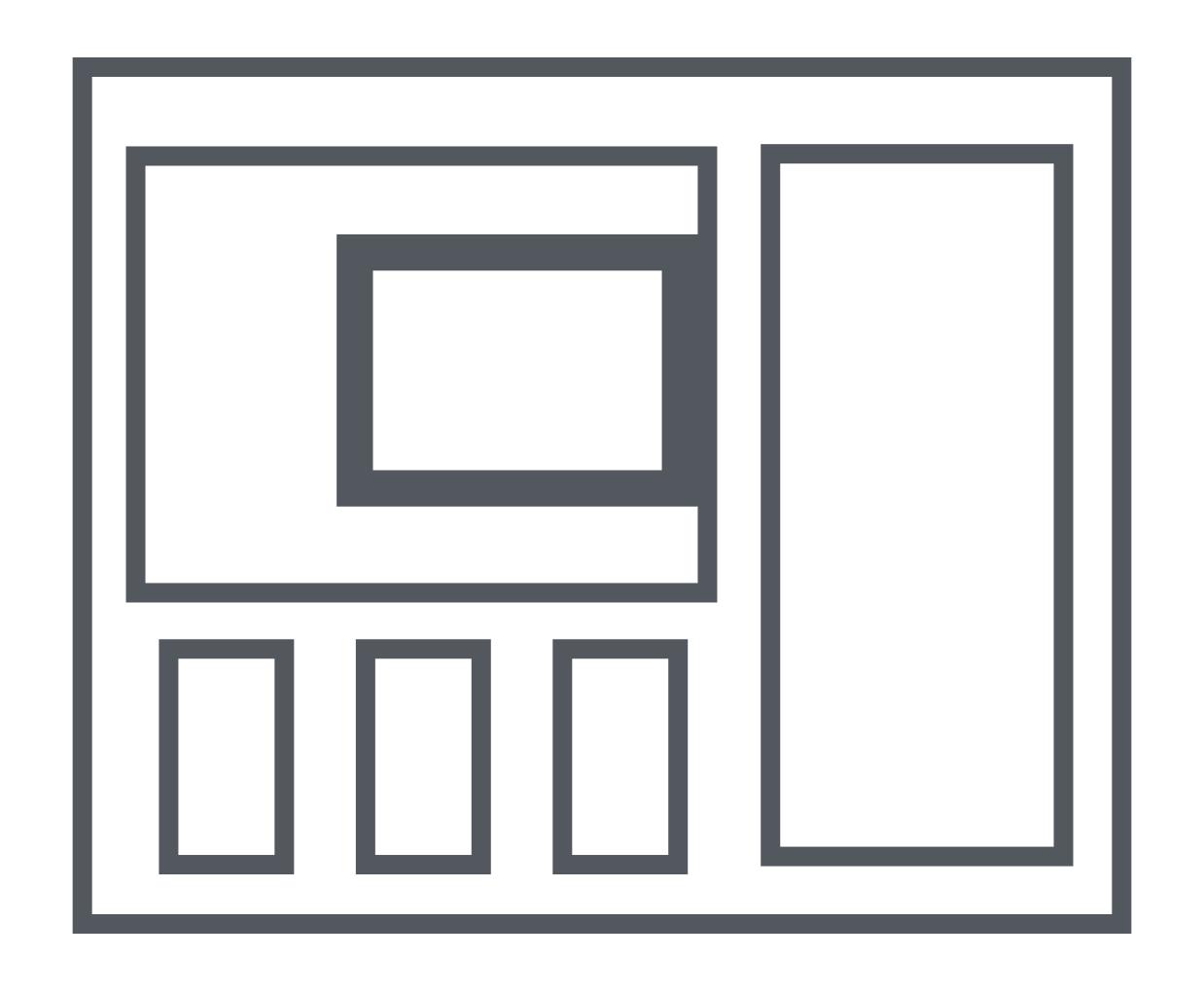
HTML & CSS

# THE BOX MODEL

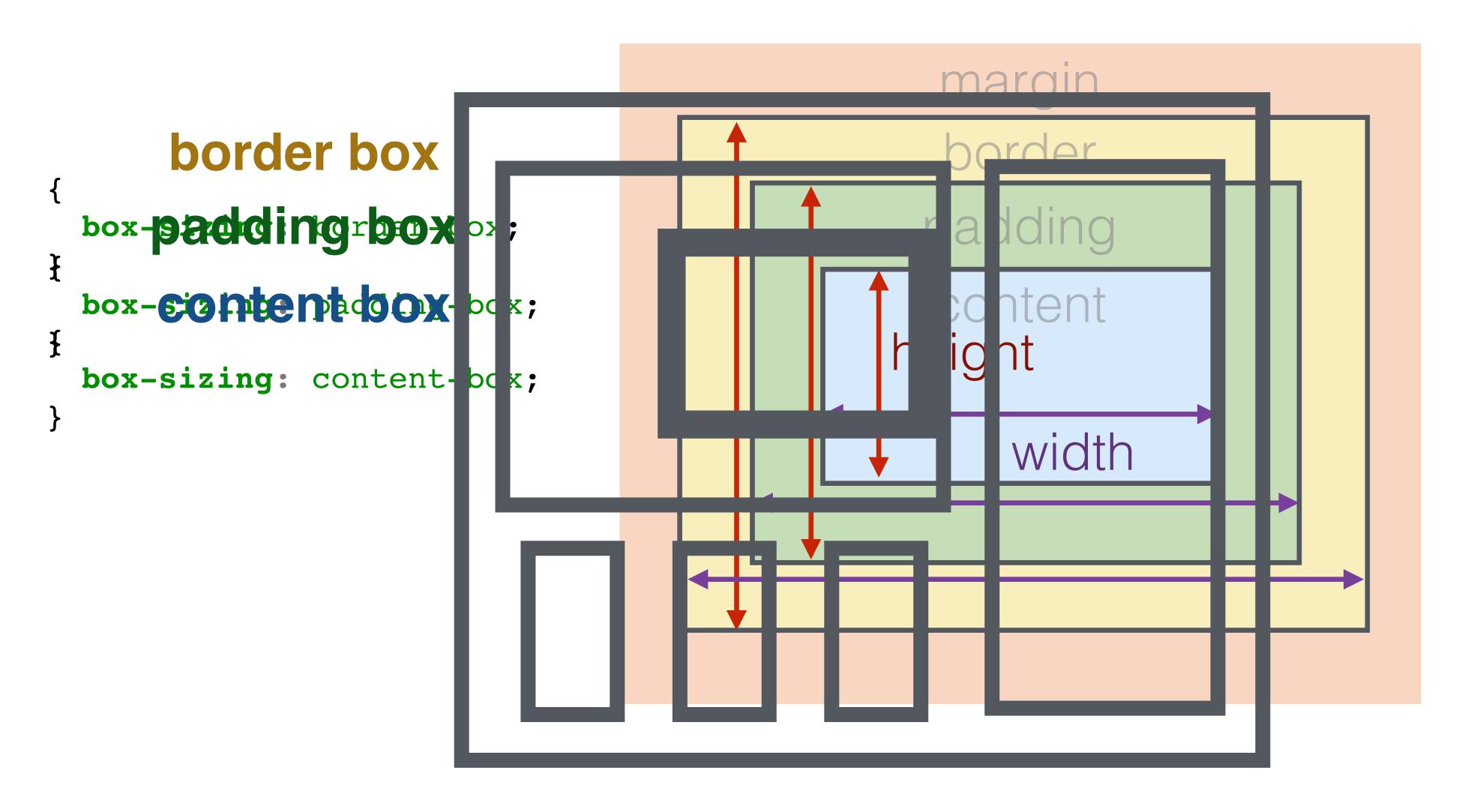
"Everything on the DOM is a rectangle"

**JOE ALVES** 

# BOX MODEL

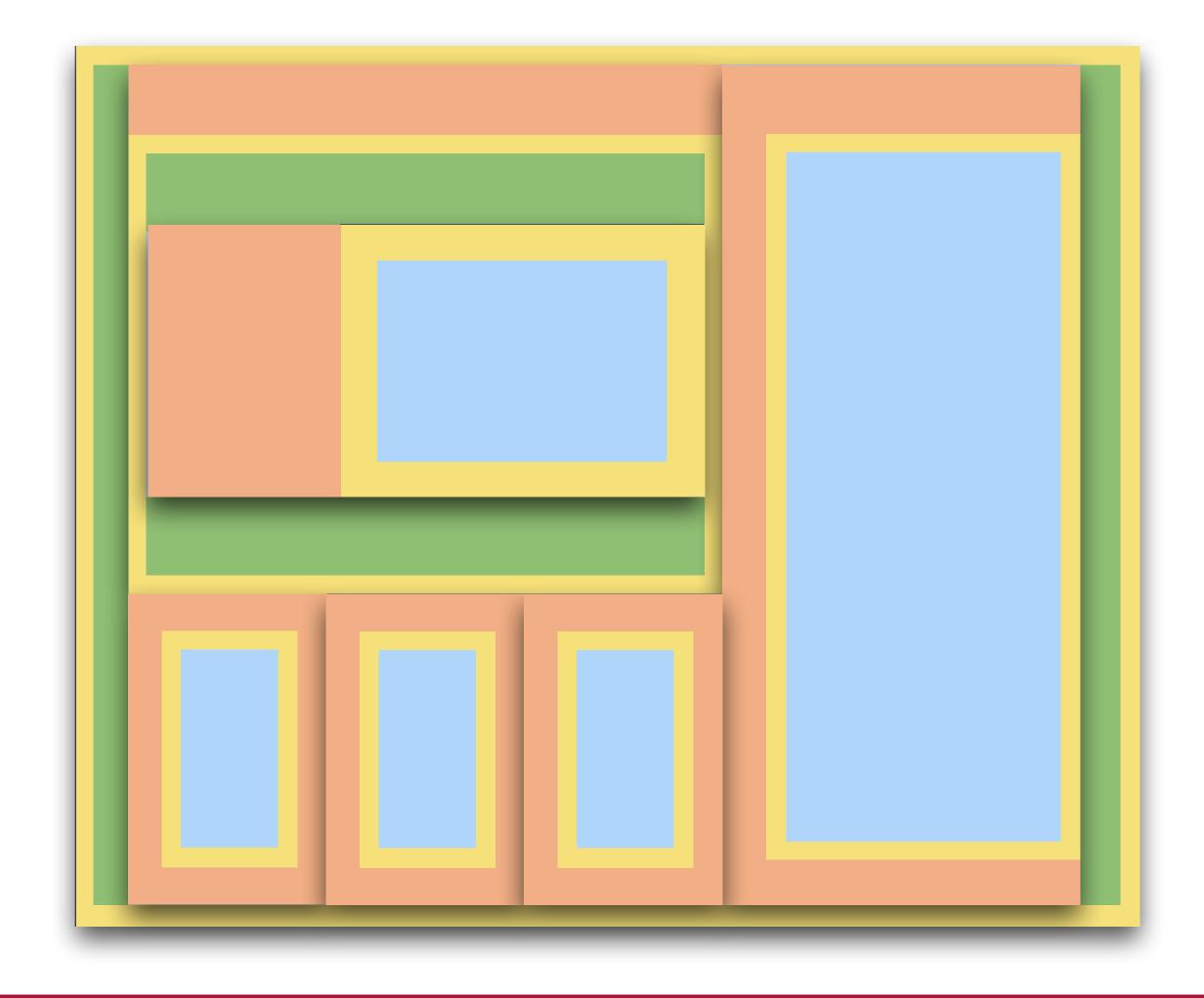


# BOX MODEL



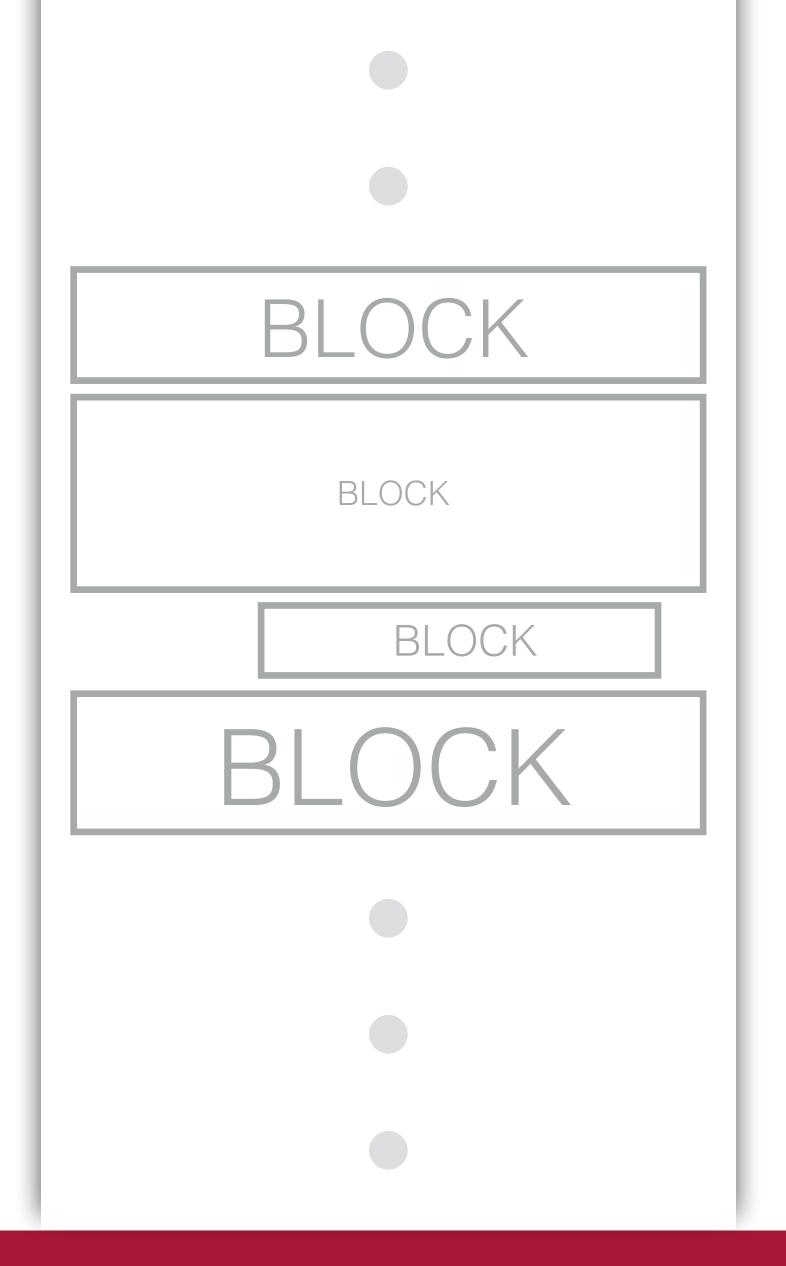
# BOX MODEL

fractal!



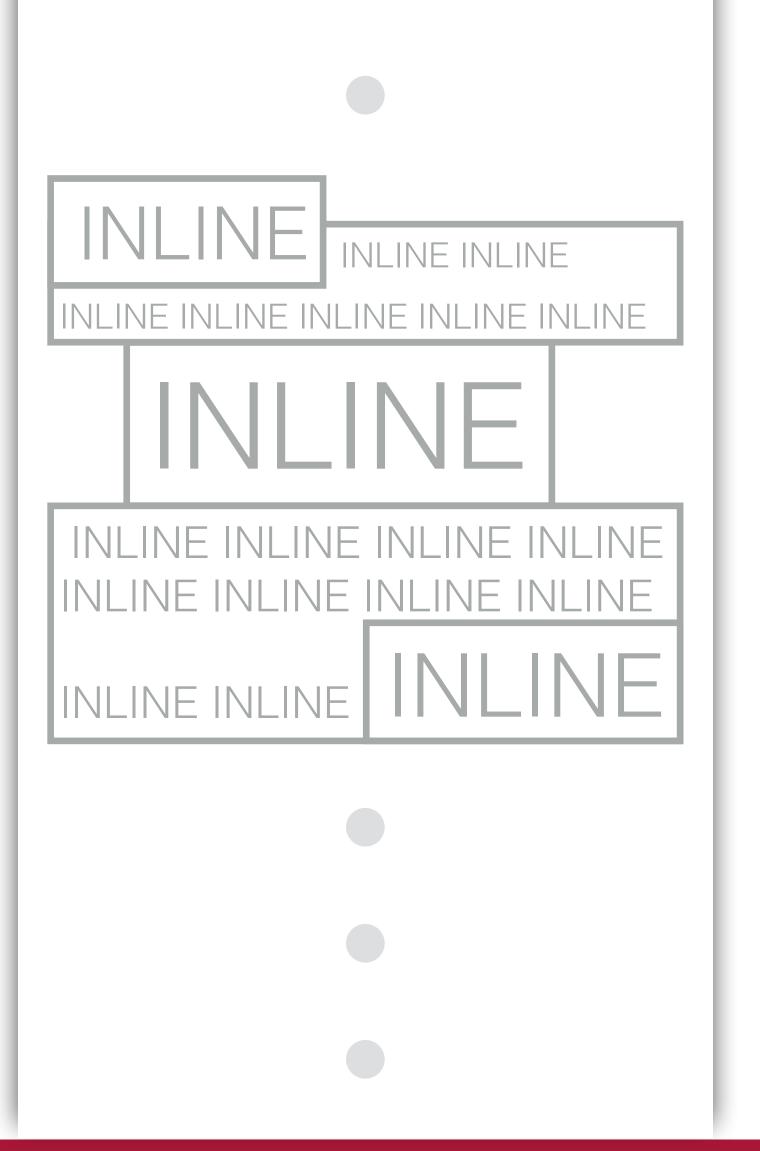
# BLOCK-LEVEL

- 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



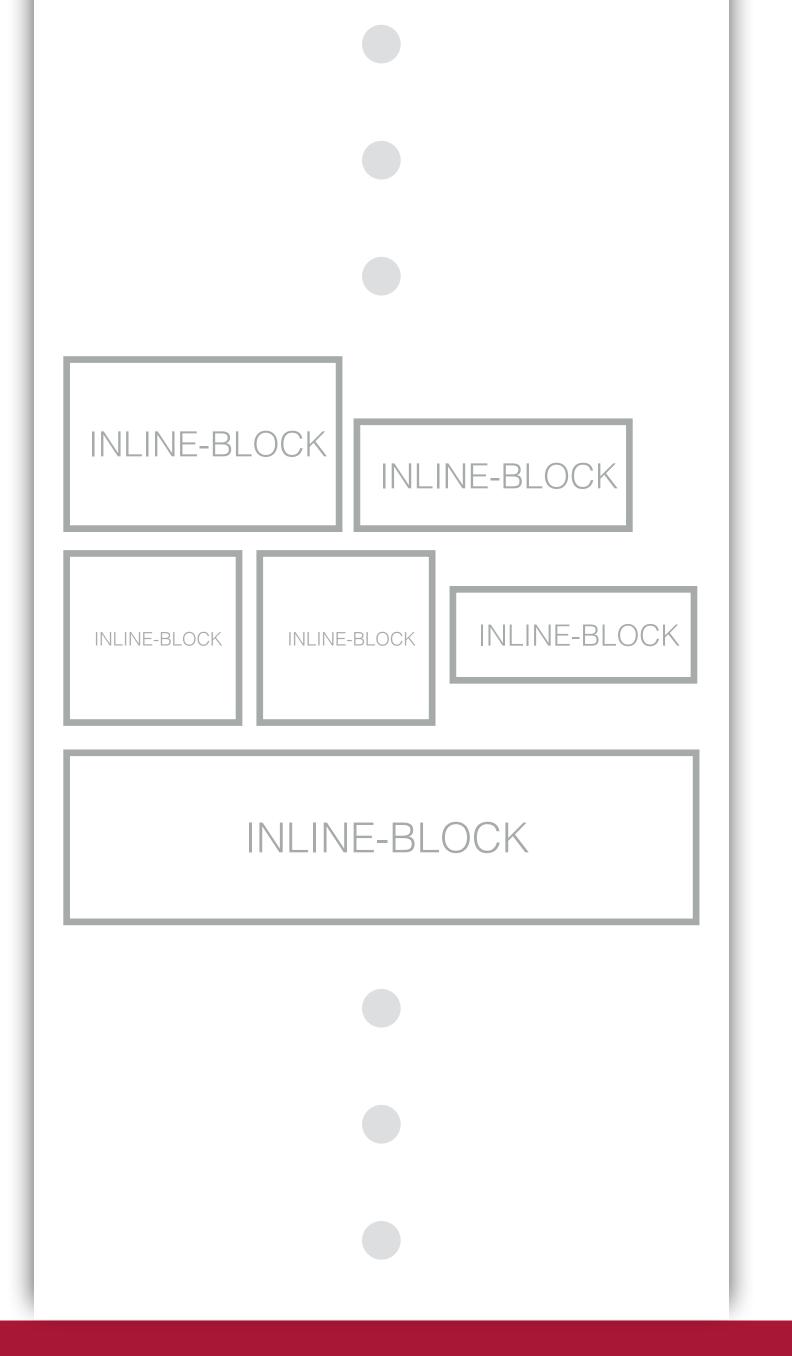
# INLINE-LEVEL

- 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

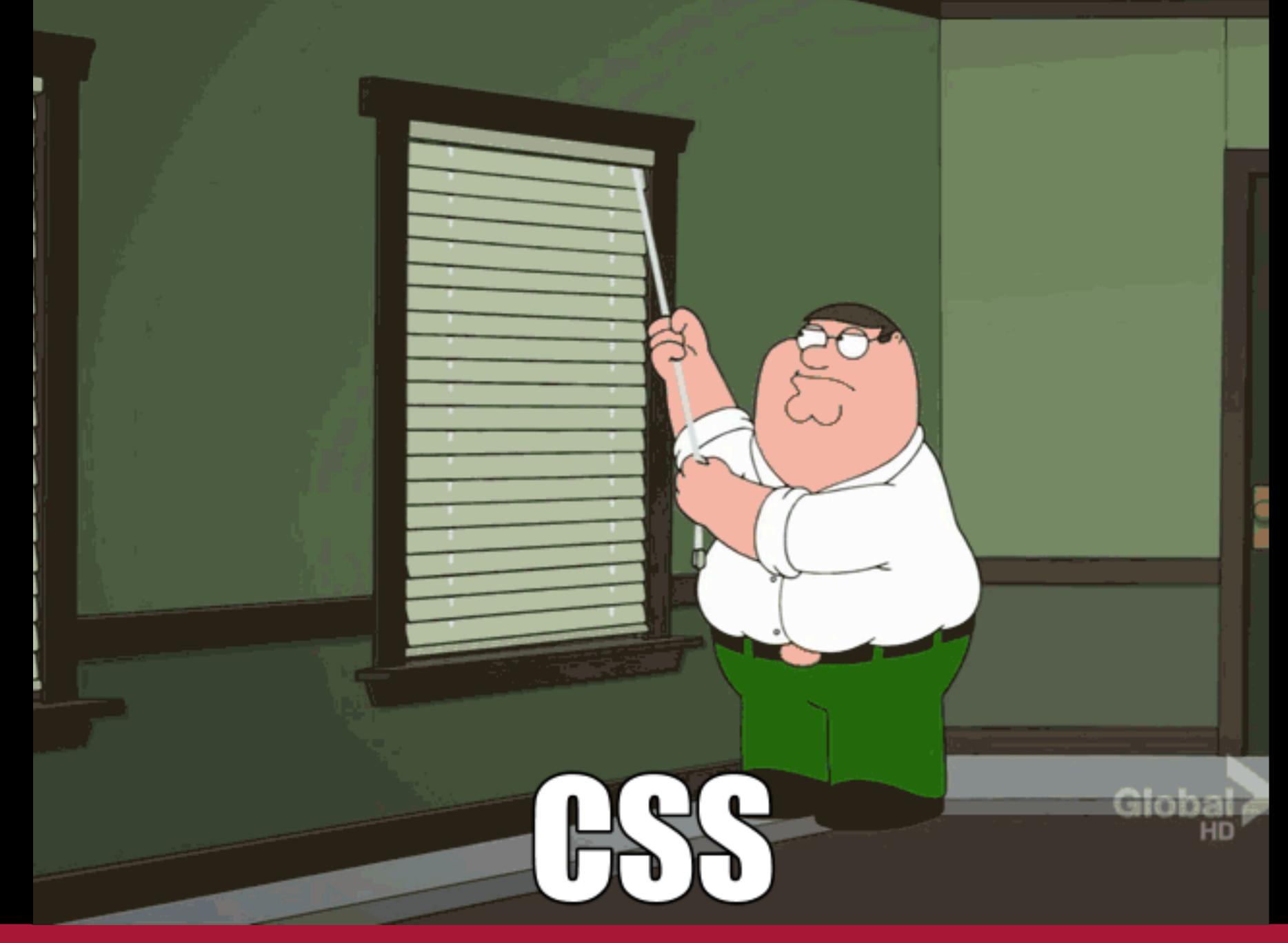


# BLOCK vs INLINE vs INLINE-BLOCK

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

- <a>>
- <span>
- <strong>, <em>

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



S T B A C T I C A L Y AMESOME S B Y E E SEETS YAET

# SCSS

- "nesting"
- "variables"
- "loops"
- "functions"
- "modules"

# HOW DOES SCSS WORK?

# SCSS COMPILES TO CSS

# NESTING

# SCSS

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

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

### VARIABLES

#### SCSS

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

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

# LOOPS

# SCSS

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

```
h1 {
    font-size: 10px;
h2 {
    font-size: 20px;
h3 {
    font-size: 30px;
```

# MIXINS

#### SCSS

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

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

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

# MODULES

### SCSS

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

```
* 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; }
 * Remove the gray background color from active links in IE 10.
*/
   background: transparent; }
/* ... */
```