CSS preprocessors to do the dirty work

Gunnar Bittersmann @g16n

Calculations

```
div {
        margin: (30em – 20px) / 2;
}
```



```
div {
        margin: 230px;
}
```

Calculations

```
div {
        margin: calc((30em – 20px) / 2);
}
```



```
div {
        margin: calc((30em – 20px) / 2);
}
```

Variables

```
$red: #c00;

.warning {
        border: 3px solid $red;
}
.warning h2 {
        color: $red;
}
button.danger {
        background: $red;
}
```



```
.warning {
        border: 3px solid #c00;
}
.warning h2 {
        color: #c00;
}
button.danger {
        background: #c00;
}
```

Variables

```
$warning: #c00;
```

```
.warning {
        border: 3px solid $warning;
}
.warning h2 {
        color: $warning;
}
button.danger {
        background: $warning;
}
```



```
.warning {
        border: 3px solid #c00;
}
.warning h2 {
        color: #c00;
}
button.danger {
        background: #c00;
}
```

Variables

```
$red: #c00;
$warning: $red;

.warning {
        border: 3px solid $warning;
}
.warning h2 {
        color: $warning;
}
button.danger {
        background: $warning;
}
```



```
.warning {
        border: 3px solid #c00;
}
.warning h2 {
        color: #c00;
}
button.danger {
        background: #c00;
}
```

Nesting

```
$red: #c00;
$warning: $red;

.warning {
        border: 3px solid $warning;

        h2 {
                color: $warning;
        }
}
```



```
.warning {
        border: 3px solid #c00;
}
.warning h2 {
        color: #c00;
}
```

Mixins

```
@mixin button($color: #666) {
    background: $color;
    border: none;
    border-radius: 3px;
    color: white;
}
button.escape {
    @include button;
}
button.submit {
    @include button(#3c3);
}
button.danger {
    @include button(#c00);
}
```



```
button.escape {
    background: #666;
    border: none;
    border-radius: 3px;
    color: white;
}
button.submit {
    background: #3c3;
    border: none;
    border-radius: 3px;
    color: white;
}
button.danger {
    background: #c00;
    border: none;
    border-radius: 3px;
    color: white;
}
```

Extensions

```
button.escape {
        background: #666;
        border: none;
        border-radius: 3px;
        color: white;
}
button.submit {
        @extend button.escape;
        background: #3c3;
}
button.danger {
        @extend button.escape;
        background: #c00;
}
```



```
button.escape, button.submit,
button.danger {
        background: #666;
        border: none;
        border-radius: 3px;
        color: white;
}
button.submit {
        background: #3c3;
}
button.danger {
        background: #c00;
}
```

Mixins vs. extensions

```
button.escape {
    background: #666;
    border: none;
    border-radius: 3px;
    color: white;
}
button.submit {
    background: #3c3;
    border: none;
    border-radius: 3px;
    color: white;
}
button.danger {
    background: #c00;
    border: none;
    border-radius: 3px;
    color: white;
}
```

```
button.escape, button.submit,
button.danger {
    background: #666;
    border: none;
    border-radius: 3px;
    color: white;
}
button.submit {
    background: #3c3;
}
button.danger {
    background: #c00;
}
```

Extensions!

```
button.escape {
    background-image: url(
    AAAAAwCAMAAABg3Am1AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlY
    WR5ccllPAAAAyJpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBhY2tldCBi
    ZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+IDx4OnhtcG1ld
    GEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBD
    b3JlIDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgI
    CAgIj4gPHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5Lz
    k7WnzkpVHvx59LTLPJqRFLcNEBC/T1oRVAVJezkCdjATJRSZHhDM8XKkM92S/Xeq
    T4LAzhomMjIUi1aSdONSiyRuZyaLyMAZaVQhof7E1WGdKdjiCQ8kFWdZCnrAudunL
    twjBnnSH8IRp2bvnKyRqdSQLDQz8vwADANNWPBg6OrUvAAAAAElFTkSuQmCC);
    border: none;
    border-radius: 3px;
    color: white;
}
```

Functions

```
$grid_base: 960px;
$grid_col: 60px;
$grid_gutter: 20px;

@function grid_width($cols) {
        @return $grid_base / 12 * $cols - $grid_gutter;
}

.secondary-navigation {
        width: grid_width(2);
}
.main {
        width: grid_width(10);
}
```

Control structures

@if, @else

@each

Dirty work?

1. vendor prefixes
2. Supporting vintage IE
3. OOCSS

Vendor prefixes

Vendor prefixes

```
button {
        -moz-border-radius: 3px;
        -ms-border-radius: 3px;
        -o-border-radius: 3px;
        -webkit-border-radius: 3px;
        border-radius: 3px;
}
```

Vendor prefixes

```
$defaultPrefixes: -moz-, -ms-, -o-, -webkit-, '';

@mixin experimentalProperty($property, $value, $prefixes: $defaultPrefixes) {
	@each $prefix in $prefixes {
		#{$prefix}#{$property}: #{$value};
	}
}
@mixin experimentalValue($property, $value, $prefixes: $defaultPrefixes) {
	@each $prefix in $prefixes {
		#{$property}: #{$prefix}#{$value};
	}
}

button {
	@include experimentalProperty(border-radius, 3px);
}
```

Vendor prefixes

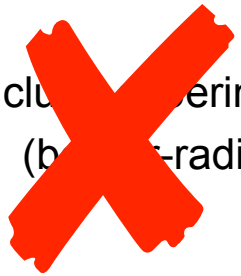
```
button {
    @include experimentalProperty
        (border-radius, 3px);
}
```



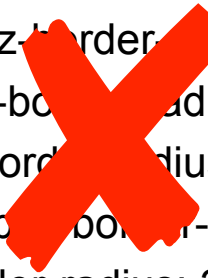
```
button {
    -moz-border-radius: 3px;
    -ms-border-radius: 3px;
    -o-border-radius: 3px;
    -webkit-border-radius: 3px;
    border-radius: 3px;
}
```

Vendor prefixes

```
button {
    @inclu    erimentalProperty
        (b    -radius, 3px);
}
```




```
button {
    -moz-    order-    dius: 3px;
    -ms-bo    adius: 3px;
    -o-bord    dius: 3px;
    -web    bor    -radius: 3px;
    border-radius: 3px;
}
```

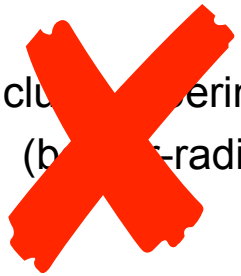


```
button {
    border-radius: 3px;
}
```

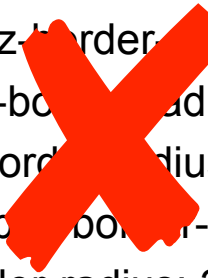
```
button {
    border-radius: 3px;
}
```

Vendor prefixes

```
button {
    @inclu    erimentalProperty
        (b    -radius, 3px);
}
```




```
button {
    -moz-    order-    dius: 3px;
    -ms-bo    adius: 3px;
    -o-bord    dius: 3px;
    -web    bo    -radius: 3px;
    border-radius: 3px;
}
```



```
button {
    border-radius: 3px;
}
```

```
button {
    border-radius: 3px;
}
```

```
button {
    @include experimentalProperty
        (border-radius, 3px,
        (-moz-, -webkit, ''));
}
```

```
button {
    -moz-border-radius: 3px;
    -webkit-border-radius: 3px;
    border-radius: 3px;
}
```

Vendor prefixes

```
button {
    @include experimentalValue(background, linear-gradient(top, #c00, #900));
}
```



```
button {
    background: -moz-linear-gradient(top, #c00, #900);
    background: -ms-linear-gradient(top, #c00, #900);
    background: -o-linear-gradient(top, #c00, #900);
    background: -webkit-linear-gradient(top, #c00, #900);
    background: linear-gradient(top, #c00, #900);
}
```

Vendor prefixes

```
button {
    @include experimentalValue(background, linear-gradient(top, #c00, #900));
}
```



```
button {
    background: -moz-linear-gradient(top, #c00, #900);
    background: -ms-linear-gradient(top, #c00, #900);
    background: -o-linear-gradient(top, #c00, #900);
    background: -webkit-linear-gradient(top, #c00, #900);
    background: linear-gradient(top, #c00, #900);          /* wrong syntax! */
}
```

Vendor prefixes

```
button {
        background: -moz-linear-gradient(top, #c00, #900);
        background: -moz-linear-gradient(to bottom, #c00, #900);
        background: -o-linear-gradient(top, #c00, #900);
        background: -webkit-linear-gradient(top, #c00, #900);
        background: linear-gradient(to bottom, #c00, #900);
}
```

Vendor prefixes

```
button {
    @include experimentalValue(background, linear-gradient(top, #c00, #900),
        (-moz-, -webkit-, -o-));
    @include experimentalValue(background, linear-gradient(to bottom, #c00, #900),
        (-moz-, '));
}
```



```
button {
    background: -moz-linear-gradient(top, #c00, #900);
    background: -o-linear-gradient(top, #c00, #900);
    background: -webkit-linear-gradient(top, #c00, #900);
    background: -moz-linear-gradient(to bottom, #c00, #900);
    background: linear-gradient(to bottom, #c00, #900);
}
```

Vendor prefixes

```
button {
    @include experimentalValue(background, linear-gradient(#c00, #900));
}
```



```
button {
    background: -moz-linear-gradient(#c00, #900);
    background: -ms-linear-gradient(#c00, #900);
    background: -o-linear-gradient(#c00, #900);
    background: -webkit-linear-gradient(#c00, #900);
    background: linear-gradient(#c00, #900);
}
```

2

Supporting vintage IE

Separate stylesheets for IE

```
<!--[if IE lt 7]>
  <link rel="stylesheet" href="ie6.css"/>
<![endif]-->
<!--[if IE 7]>
  <link rel="stylesheet" href="ie7.css"/>
<![endif]-->
<!--[if IE 8]>
  <link rel="stylesheet" href="ie8.css"/>
<![endif]-->
<!--[if lte IE 9]><!-->
  <link rel="stylesheet" href="standard.css"/>
<!--<![endif]-->
```

standard.css:

```
#foo {
        background: rgba(255, 255, 255, .2);
}
```

ie8.css:

```
#foo {
        background: url(bg-transparent.png);
}
```

ie7.css:

```
#foo {
        background: url(bg-transparent.png);
}
```

ie6.css:

```
#foo {
        background: url(bg-foo.jpg);
}
```

IE hacks

```
<link rel="stylesheet" href="standard.css"/>
```

standard.css:

```
#foo {
        background: rgba(255, 255, 255, .2);
}
*+html #foo {
        background: url(bg-transparent.png);
}
* html #foo {
        background: url(bg-foo.jpg);
}
```

IE hacks

```
<link rel="stylesheet" href="standard.css"/>
```

standard.css:

```
#foo {
        background: url(bg-transparent.png);
        background: rgba(255, 255, 255, .2);
}


* html #foo {
        background: url(bg-foo.jpg);
}
```

Separate CSS vs. hacks

Benefits

- relevant code only
- possible for IE 6, 7, 8, 9
- valid standard.css

Drawbacks

- more code
- dirty hacks for IE 8, 9
- possibly invalid CSS

Drawbacks

- split to several files: hard to maintain

Benefits

- according rules close to each other in stylesheet: maintainable

Separate CSS vs. hacks

Benefits

- relevant code only
- possible for IE 6, 7, 8, 9
- valid standard.css

Drawbacks

- more code
- dirty hacks for IE 8, 9
- possibly invalid CSS

Drawbacks

- split to several files: hard to maintain

Benefits

- according rules close to each other in stylesheet:

maintainable

The best of both worlds

Benefits

- relevant code only
- possible for IE 6, 7, 8, 9
- valid standard.css

Benefits

- according rules close to each other in stylesheet:

maintainable

The best of both worlds

standard.scss:

```
$ua: standard !default;
#foo {
    @if $ua==standard {
        background: rgba(255, 255, 255, .2);
    }
    @else if $ua==ie8 or $ua==ie7 {
        background: url(bg-transparent.png);
    }
    @else {
        background: url(bg-foo.jpg);
    }
}
```

ie8.scss:

ie7.scss:

ie6.scss:

```
$ua: ie6;
@import standard;
```

standard.css:

```
#foo {
        background: rgba(255, 255, 255, .2);
}
```

ie8.css:

```
#foo {
        background: url(bg-transparent.png);
}
```

ie7.css:

```
#foo {
        background: url(bg-transparent.png);
}
```

ie6.css:

```
#foo {
        background: url(bg-foo.jpg);
}
```

Relevant code only

```
h1 {
        font-size: 24px;
        font-size: 1.5rem;
}
```

Relevant code only

```
$fontbase: 16px;

@function remify($px) {
    @if $ua == standard {
        @return #{$px/$fontbase}rem;
    }
    @else {
        @return #{$px};
    }
}

h1 {
        font-size: remify(24px);
}
```

standard.css:

```
h1 {
        font-size: 1.5rem;
}
```

ie8.css:

```
h1 {
        font-size: 24px;
}
```

ie7.css:

```
h1 {
        font-size: 24px;
}
```

ie6.css:

```
h1 {
        font-size: 24px;
}
```

Relevant code only

```
$fontbase: 16px;

@function remlength($rem) {
    @if $ua == standard {
        @return #{$rem};
    }
    @else {
        @return #{$rem/1rem * $fontbase};
    }
}

h1 {
        font-size: remlength(1.5rem);
}
```

standard.css:

```
h1 {
        font-size: 1.5rem;
}
```

ie8.css:

```
h1 {
        font-size: 24px;
}
```

ie7.css:

```
h1 {
        font-size: 24px;
}
```

ie6.css:

```
h1 {
        font-size: 24px;
}
```

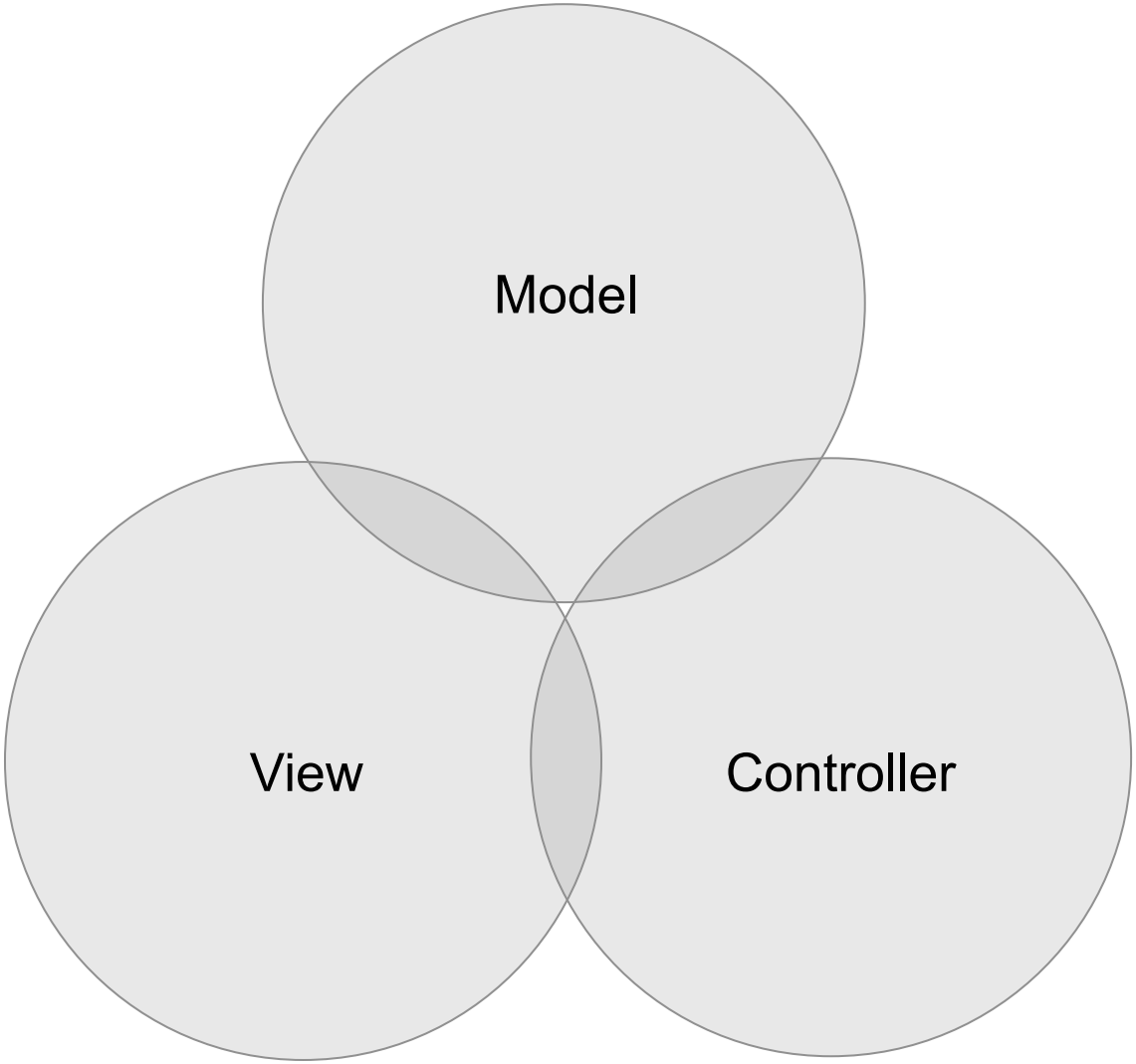
Relevant code only

```
#foo {
   @if $ua==standard {
      background-image: url(
      AAAAAwCAMAAABg3Am1AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlY
      WR5ccllPAAAAyJpVFh0WE1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBhY2tldCBi
      ZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlklj8+IDx4OnhtcG1ld
      GEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBD
                                    ⋮
      k7WnzkpVHvx59LTLPJqRFLcNEBC/T1oRVAVJezkCdjATJRSZHhDM8XKkM92S/Xeq
      T4LAzhomMjIUi1aSdONSiyRuZyaLyMAZaVQhof7E1WGdKdjiCQ8kFWdZCnrAudunL
      twjBnnSH8IRp2bvnKyRqdSQLDQz8vwADANNWPBg6OrUvAAAAAElFTkSuQmCC);
   }
   @else
   {
      background-image: url(bg-foo.png);
   }
}
```

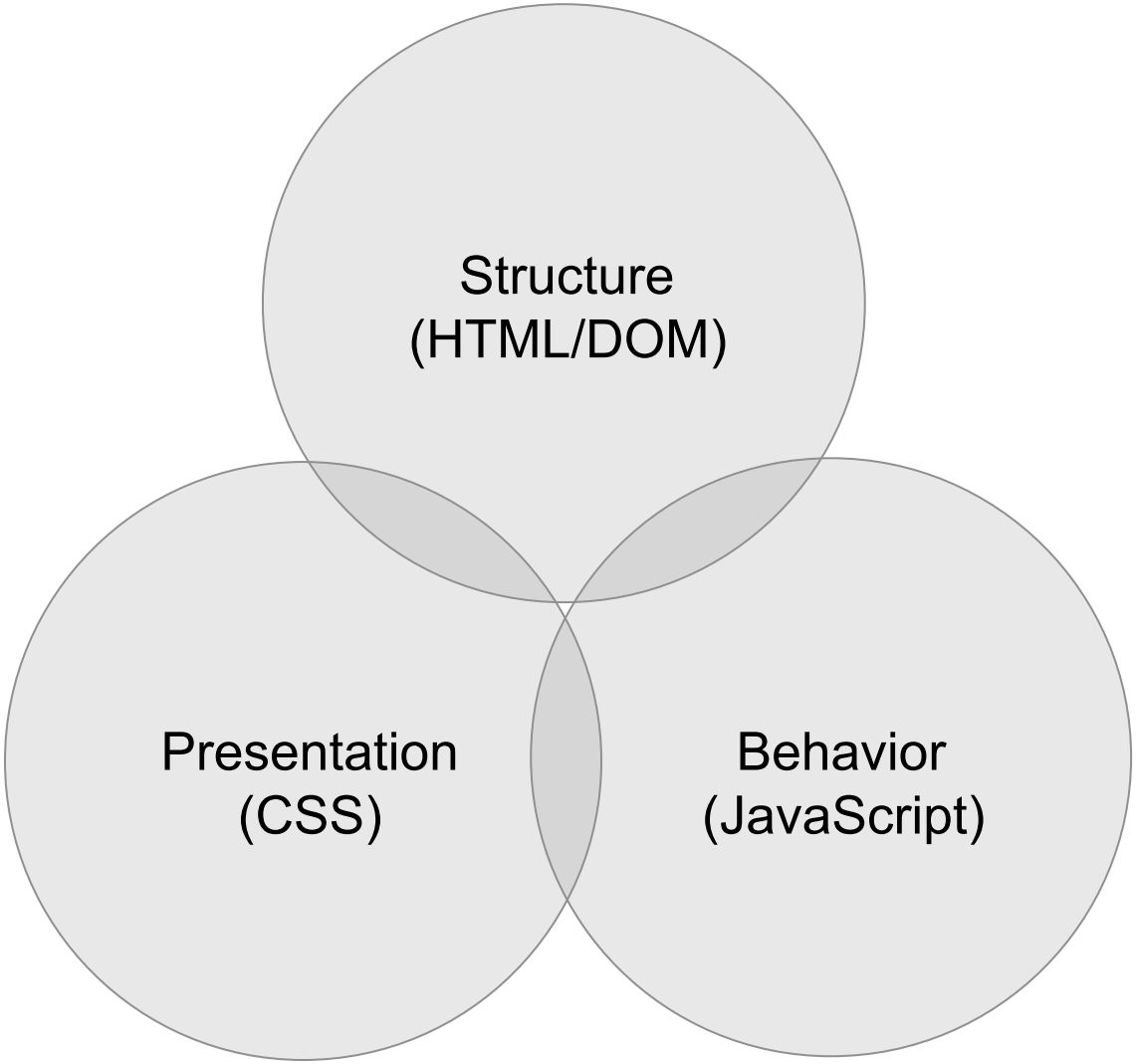
3

OOCSS

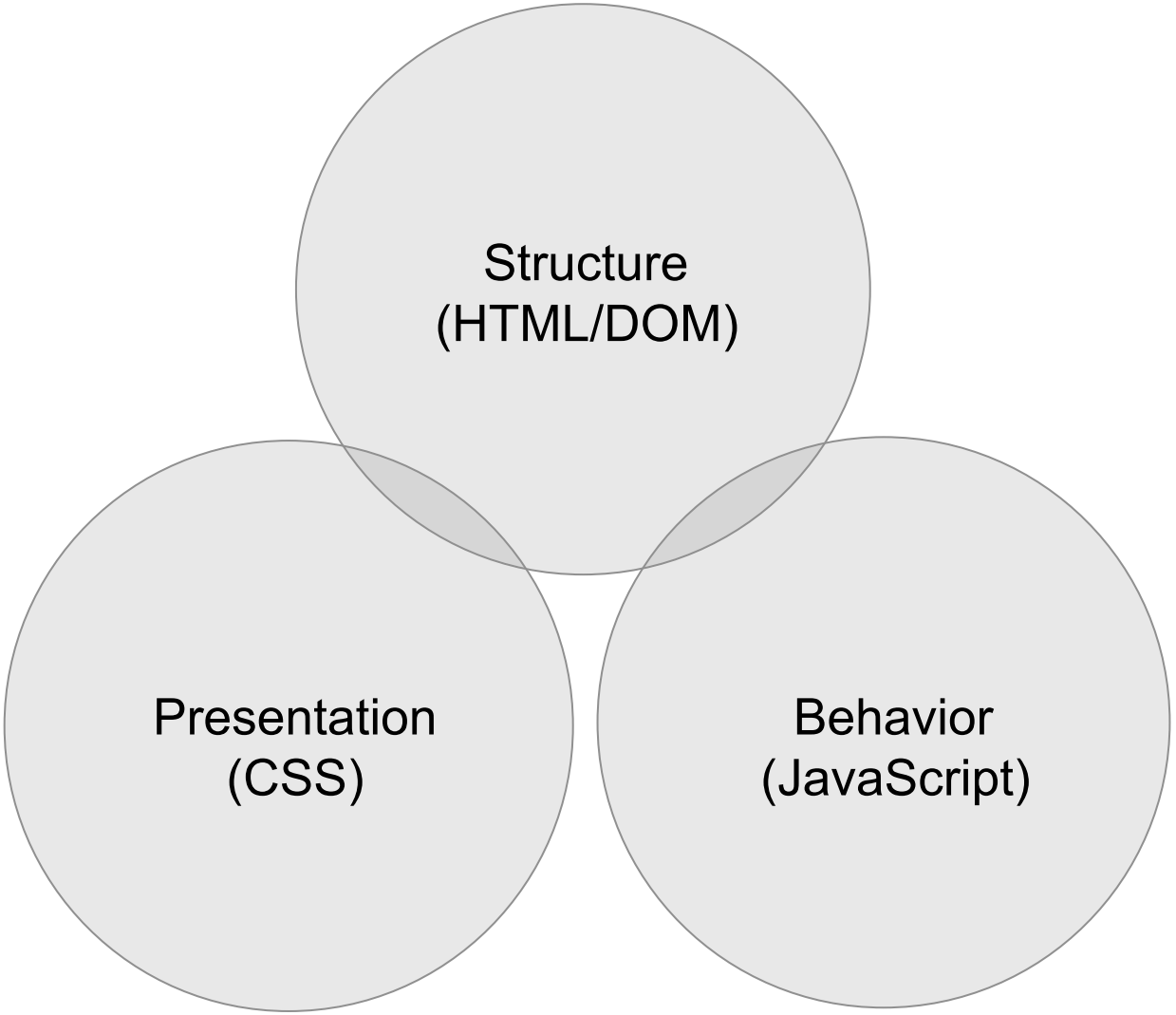
Separation of concerns



Separation of concerns



Separation of concerns



What's this OOCSS?

Object Oriented CSS

CSS "object": a repeating visual pattern, which can be abstracted into an independent snippet of HTML, CSS, and possibly JavaScript.

Goal: performance, maintainability

Principles:

1. Separate structure and skin
2. Separate container and content

Means:

- class selectors
- no type selectors
- no ID selectors
- no descendant combinators
- lots of presentational classes in the mark-up

OOCSS

◂ Back Cancel × Next ▸

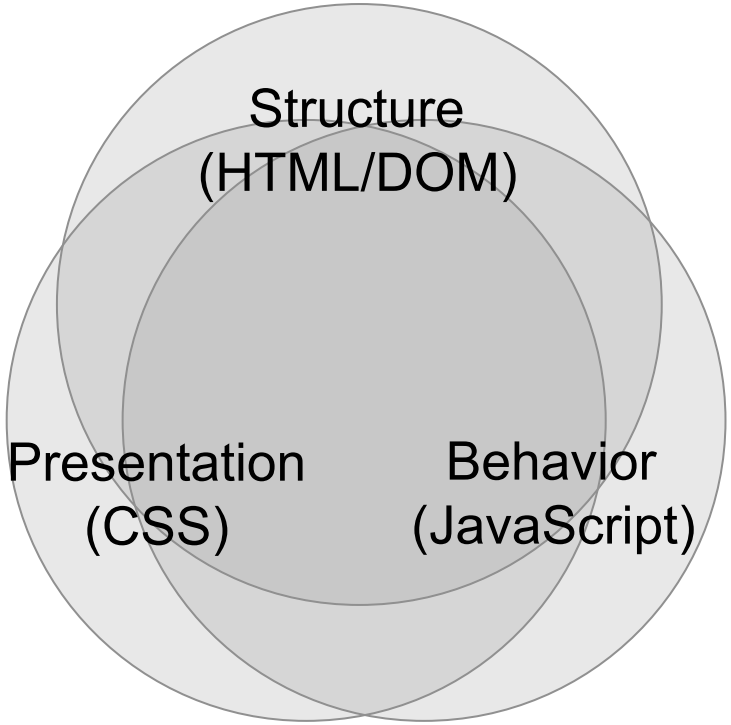
```
<button type="button” class="btn-medium btn-gray btn-arrow-left">Back</button>
<button type="button” class="btn-medium btn-gray btn-cross">Cancel</button>
<button type="submit” class="btn-large btn-red btn-arrow-right">Next</button>
```

```
.btn-medium              { font-size: 1.2em }
.btn-large               { font-size: 1.8em }
.btn-gray                { background: gray; color: black }
.btn-red                 { background: red; color: white }
.btn-arrow-left::before  { content: '◂ ' }
.btn-arrow-right::after  { content: ' ▸' }
.btn-cross::after        { content: ' ×' }
```

’Nuff said


Gunnar Bittersmann
@g16n
OOCSS is about pure CSS —@stubbornella
Agreed. But it is about shitty HTML.
#smashingconf
Antworten
Löschen
Favorisieren

OOCSS



“Semantic” Mark-up


```
<button type="button” class="back">Back</button>
<button type="button” class="cancel">Cancel</button>
<button type="submit” class="forward”>Next</button>
```

```
button                  { font-size: 1.2em }
button[type='submit']   { font-size: 1.8em }
button                  { background: gray; color: black }
button[type='submit']   { background: red; color: white }
button.back::before     { content: '◂ ' }
button.forward::after   { content: ' ▸' }
button.cancel::after    { content: ' ×' }
```

“Semantic” Mark-up


```
<button type="button” class="back">Back</button>
<button type="button” class="cancel">Cancel</button>
<button type="submit” class="forward">Next</button>
```

```
button                  { background: gray; color: black; font-size: 1.2em }
button[type='submit']   { background: red; color: white; font-size: 1.8em }
button.back::before     { content: '◂ ' }
button.forward::after   { content: ' ▸' }
button.cancel::after    { content: ' ×' }
```

OOCSS vs. „Sem.“ Markup

Benefits

- reusable units
- slightly shorter CSS code
- slightly more performant CSS code

Drawbacks

- presentational mark-up, hard to maintain
- blown-up mark-up

Drawbacks

- usage of the cascade: less performant selectors
- selector specifity (be aware of)
- longer CSS code

Benefits

- separation of mark-up and presentaion: better maintainable
- shorter HTML code

OOCSS vs. „Sem.“ Markup

Benefits

- reusable units
- slightly more performant CSS code

Drawbacks

- presentational mark-up, hard to maintain
- blown-up mark-up

Drawbacks

- usage of the cascade: less performant selectors
- selector specifity (be aware of)

Benefits

- separation of mark-up and presentaion: better maintainable
- shorter HTML code

OOCSS vs. „Sem.“ Markup

Benefits

- reusable units

Drawbacks

- presentational mark-up, hard to maintain
- blown-up mark-up

Drawbacks

- selector specifity (be aware of)

Benefits

- separation of mark-up and presentaion: better maintainable
- shorter HTML code

OOCSS vs. „Sem.“ Markup

Benefits

reusable units

Drawbacks

- presentational mark-up, hard to maintain
- blown-up mark-up

Drawbacks

- selector specifity (be aware of)

Benefits

- separation of mark-up and presentaion:

  better maintainable

- shorter HTML code

intermediate preprocessor layer

```
%btn-medium                { font-size: 1.2em }
%btn-large                 { font-size: 1.8em }
%btn-gray                  { background: gray; color: black }
%btn-red                   { background: red; color: white }
%btn-arrow-left::before    { content: '◂ ' }
%btn-arrow-right::after    { content: ' ▸' }
%btn-cross::after          { content: ' ×' }

button                     { @extend %btn-medium; @extend %btn-gray }
button[type='submit']      { @extend %btn-large; @extend %btn-red }
button.back                { @extend %btn-arrow-left }
button.forward             { @extend %btn-arrow-right }
button.cancel              { @extend %btn-cross }
```

intermediate preprocessor layer

Benefits:

- reusable “objects” in intermediate preprocessor layer
- separation of structure and presentation (maintainability)

Drawbacks:

- possibly long lists of selectors in generated CSS

TL;DR

CSS preprocessors make life easier

- vendor prefixes

and make things possible that you would not want to do by hand (the **dirty work**):

- separate IE stylesheets generated from one source
- intermediate layer instead of OOCSS