

Atomic CSS_ В дикой природе_

Савва Михалевский





Савва Михалевский_

Front-End Архитектор @ Grabr.io
ex-dev @ MOEX, Sberbank CIB

Еще одна методология

Разве не достаточно OOCSS, BEM, SMACSS и пр.?

ЧТО МНЕ НЕ НРАВИТСЯ В CSS

1 HTML и CSS в отдельных файлах

Два контекста

Проект уже разделен на компоненты

А сколько у вас в проекте holder'ов и wrapper'ов?

Что мне не нравится в CSS

- 1 HTML и CSS в отдельных файлах
- 2 Надо придумывать названия
 - Коллизии имен
 - Специфичность селекторов

ЧТО МНЕ НЕ НРАВИТСЯ В CSS

- 1 HTML и CSS в отдельных файлах
- 2 Надо придумывать названия
- 3 **CSS правила для разных элементов повторяются**
Источник ошибок
Сборка CSS избыточна

ЧТО МНЕ НЕ НРАВИТСЯ В CSS

- 1 HTML и CSS в отдельных файлах
- 2 Надо придумывать названия
- 3 CSS правила для разных элементов повторяются
- 4 Media Queries и Responsiveness
>2 breakpoint'ов и верстка превращается в ад

Разметка

Стили

Media Queries

Палитры



```
$white: #fff;
```

```
$black: #000;
```

```
$brand: #fe0;
```

```
$cyan: #0ff;
```

Функциональные классы

```
$white: #fff;
```

```
$black: #000;
```

```
$brand: #fe0;
```

```
$cyan: #0ff;
```



```
.color-white { color: #fff; }
```

```
.color-black { color: #000; }
```

```
.color-brand { color: #fe0; }
```

```
.color-cyan { color: #0ff; }
```

Функциональные классы

```
$space-1: 5px;
```

```
$space-2: 10px;
```

```
.padding-5px { padding: 5px; }
```

```
.margin-5px { margin: 5px; }
```

```
.padding-10px { padding: 10px; }
```

```
.margin-10px { margin: 10px; }
```

Функциональные классы

```
.overflow-hidden { overflow: hidden; }  
.overflow-scroll { overflow: scroll; }
```

```
.border-color-white {  
    border-color: white;  
}
```

```
.border-style-solid {  
    border-style: solid;  
}
```

```
.border-width-1px {  
    border-width: 1px;  
}
```

Universal CSS

<https://github.com/marmelab/universal.css>

- 1 *Не нужно* писать новый код, он уже сгенерен
- 2 *Не нужно* придумывать имена
- 3 *Не нужно* переключаться между контекстами – вся разработка только в HTML
- 4 *Нет повторений* правил в коде

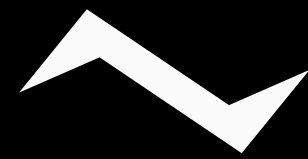
Universal CSS

<https://github.com/marmelab/universal.css>

```
<div class="border-style-solid border-color-red  
border-width-1px">  
  <!-- markup -->  
</div>
```



v4-alpha



Basscss

basscss.com

Tachyons

tachyons.io

■ border-right-0

■ border-right

■ br-0

■ rounded-right

■ rounded-right

■ br1

■ d-flex

■ flex

■ flex

■ mr-10

■ mr2

■ mr2

16

37

Проблемы Bascss

1 Не консистентный нейминг

В классе `.border-right-0` сторона называется `right`,
а `.mr-10` называется `r`.

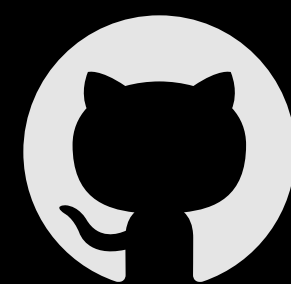
Проблемы Bascss

- 1 Не консистентный нейминг
- 2 Очень мало настроек – надо писать CSS

Проблемы Basscss

- 1 Не консистентный нейминг
- 2 Очень мало настроек – надо писать CSS
- 3 Слабая поддержка `media query`

Quantum CSS



github.com/grabr/quantum-css



Emmet

docs.emmet.io/cheat-sheet/

You can prefix abbreviations with hyphen to produce vendor-prefixed properties: `-foo`

Visual Formatting

<code>pos</code>	<code>position: relative;</code>
<code>pos:s</code>	<code>position: static;</code>
<code>pos:a</code>	<code>position: absolute;</code>
<code>pos:r</code>	<code>position: relative;</code>
<code>pos:f</code>	<code>position: fixed;</code>
<code>t</code>	<code>top: ;</code>
<code>t:a</code>	<code>top: auto;</code>
<code>r</code>	<code>right: ;</code>
<code>r:a</code>	<code>right: auto;</code>
<code>b</code>	<code>bottom: ;</code>
<code>b:a</code>	<code>bottom: auto;</code>
<code>l</code>	<code>left: ;</code>
<code>l:a</code>	<code>left: auto;</code>
<code>z</code>	<code>z-index: ;</code>

Outline

<code>ol</code>	<code>outline: ;</code>
<code>ol:n</code>	<code>outline: none;</code>
<code>olo</code>	<code>outline-offset: ;</code>
<code>olw</code>	<code>outline-width: ;</code>
<code>olw:tn</code>	<code>outline-width: thin;</code>
<code>olw:m</code>	<code>outline-width: medium;</code>
<code>olw:tc</code>	<code>outline-width: thick;</code>
<code>ols</code>	<code>outline-style: ;</code>
<code>ols:n</code>	<code>outline-style: none;</code>
<code>ols:dt</code>	<code>outline-style: dotted;</code>
<code>ols:ds</code>	<code>outline-style: dashed;</code>
<code>ols:s</code>	<code>outline-style: solid;</code>
<code>ols:db</code>	<code>outline-style: double;</code>

fw:b  font-weight: bold;
d:i  display: inline;

p → %

e → em

x → ex

pb100p tab → padding-bottom: 100%;

l-42 → left: -42px;

Литеральные $\{\text{property}\}:\{\text{value}\}$

Числовые $\{\text{property}\}\{\text{value}\}$

Литеральные `■ {property}:{value}`

```
■ bgc\:red {  
    background-color: red;  
}
```

Числовые `■ {property}{value}`

```
■ pb100p {  
    padding-bottom: 100%;  
}
```

Литеральные `■ {property}–{value}`

```
■ bgc–red {  
    background-color: red;  
}
```

Числовые `■ {property}{value}`

```
■ pb100p {  
    padding-bottom: 100%;  
}
```

▪ {property}–{value}–{suffix}

```
▪ bgc-red-h:hover {  
    background-color: red;  
}
```

```
▪ bgc-red-d[disabled] {  
    background-color: red;  
}
```

.{media}_{property}-{value}-{suffix}

```
@media (min-width: 960px) {  
  .LG_bgc-red-h:hover {  
    background-color: red;  
  }  
}
```

```
<div class="d-fx fxd-c LG_fxd-r LG_bgc-red-h">  
  <div class="LG_fxg1">  
    Grow on large viewport  
  </div>  
  <div class="fxs0">  
    Prevent shrink  
  </div>  
</div>
```

КОМПОЗИЦИЯ КЛАССОВ

Flex row `.fx-r {
 @extend .d-fx, .fxd-r;
 }`

Flex column `.fx-c {
 @extend .d-fx, .fxd-c;
 }`



```
.ellipsis {  
  &:extend(.ws-nw, .tov-e, .ov-h);  
}
```



```
.ellipsis {  
  @extend .ws-nw, .tov-e, .ov-h;  
}
```



w/ postcss-extend

```
.ellipsis {  
  @extend .ws-nw;  
  @extend .tov-e;  
  @extend .ov-h;  
}
```



```
.link {  
    @extend .c-blue, .td-n, .td-n-h;  
    @extend .MD_c-red-h, .MD_td-u-h;  
}
```

```
.link, .c-blue {
  color: blue;
}

.link, .td-n {
  text-decoration: none;
}

.link:hover, .td-n-h:hover {
  text-decoration: none;
}

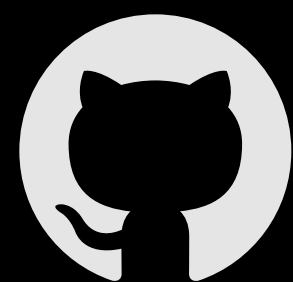
@media (min-width: 100px) {
  .link:hover, .MD_td-u-h:hover {
    text-decoration: underline;
  }

  .link:hover, .MD_c-red-h:hover {
    color: red;
  }
}
```

Что в результате?

- 1 Время на верстку сократилось в 2х
- 2 Легче поддерживать консистентность дизайн-системы
- 3 30 строк специфичного CSS на 50'000 SLOC на JS
- 4 Ограничение на использование свойств CSS и сборка для нескольких таргетов: web, email и т.д.
- 5 Возможность значительно оптимизировать результат сборки CSS

Спасибо за внимание!



github.com/grabr/quantum-css

github.com/smikhalevski



facebook.com/smikhalevski