



CSS, ВЕМ и препроцессоры



01.

BEM

02.

SASS/SCSS, Less

03.

PostCSS

04.

CSS Custom Properties



ВЕМ

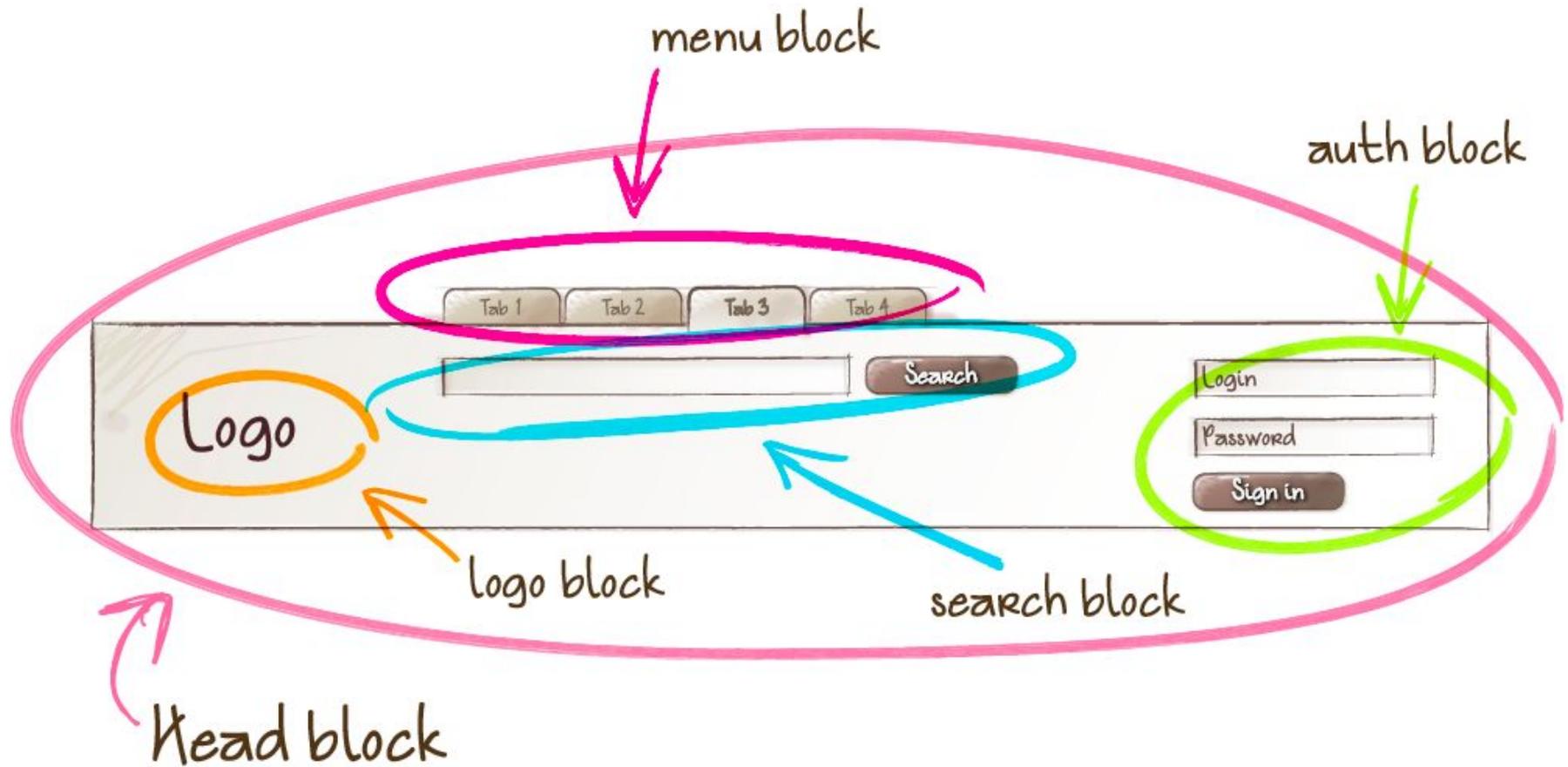
Блок__элемент_модификатор



АКАДЕМИЯ

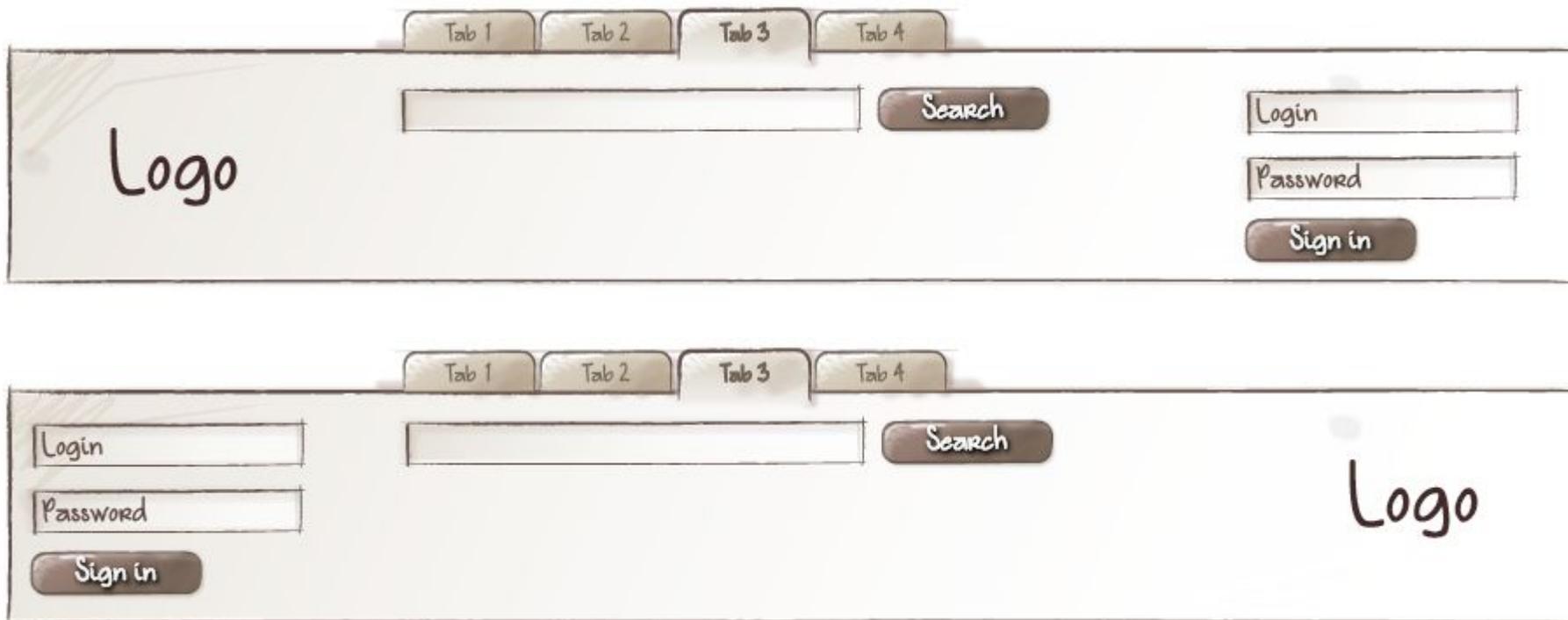
Блок

Логически и функционально независимый компонент страницы.



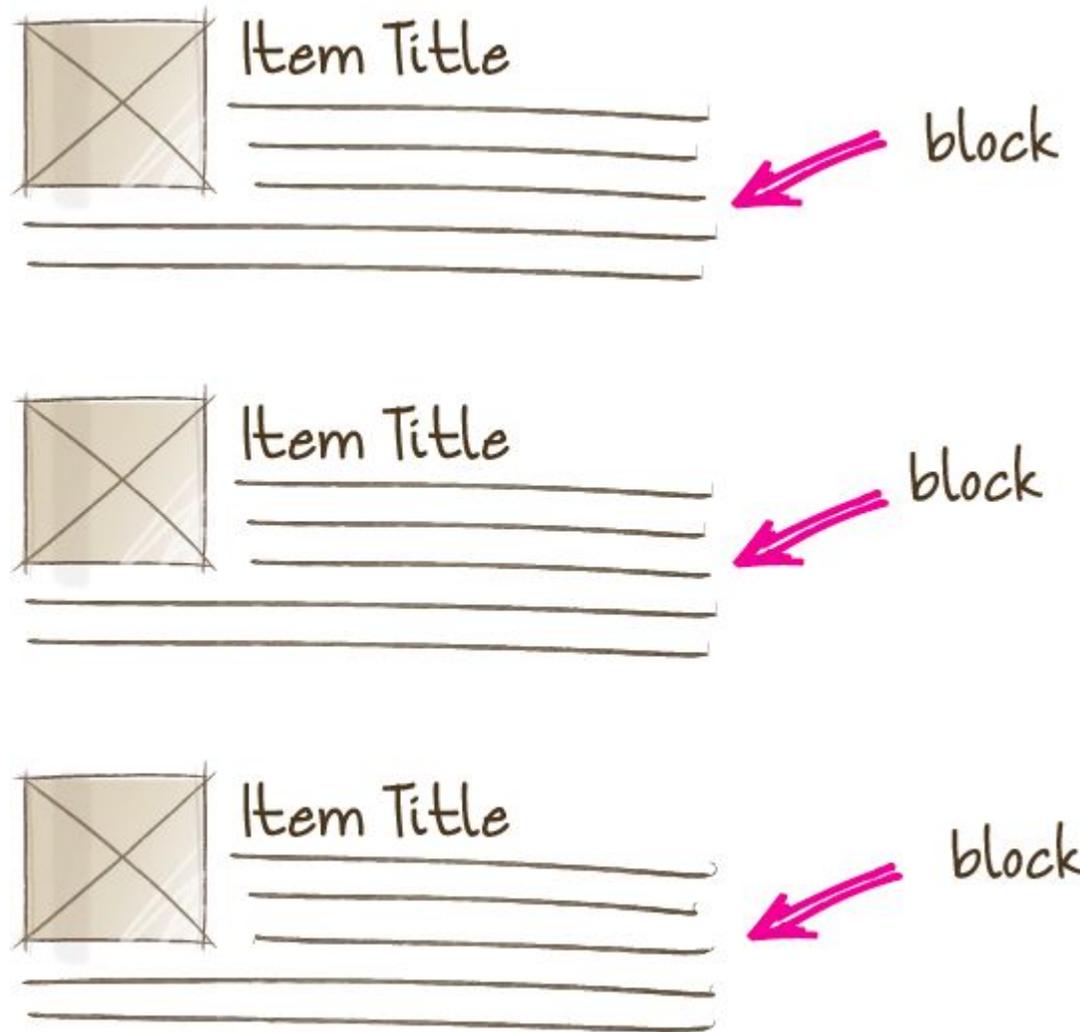
Свободное перемещение

Блоки можно перемещать в пределах одной страницы и разных проектов. Независимость блока позволяет изменять его положение на странице и обеспечивает корректную работу и внешний вид.



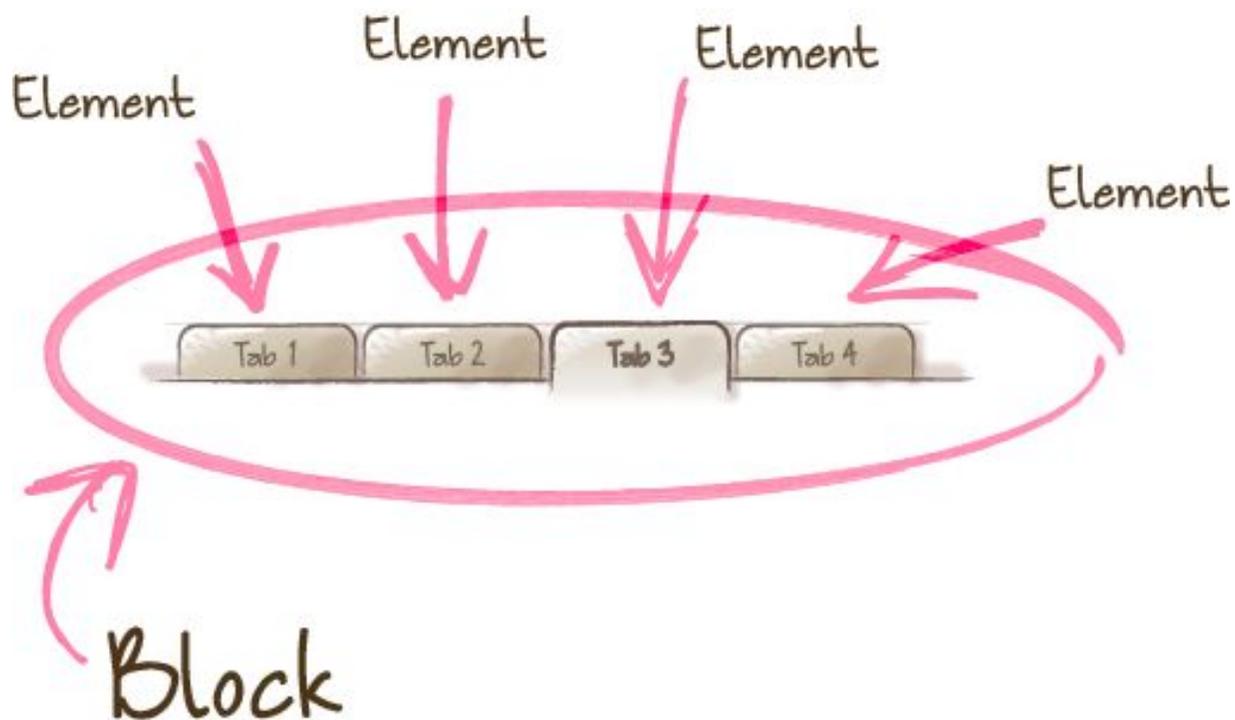
Повторное использование

В интерфейсе может одновременно присутствовать несколько экземпляров одного и того же блока.



Элемент

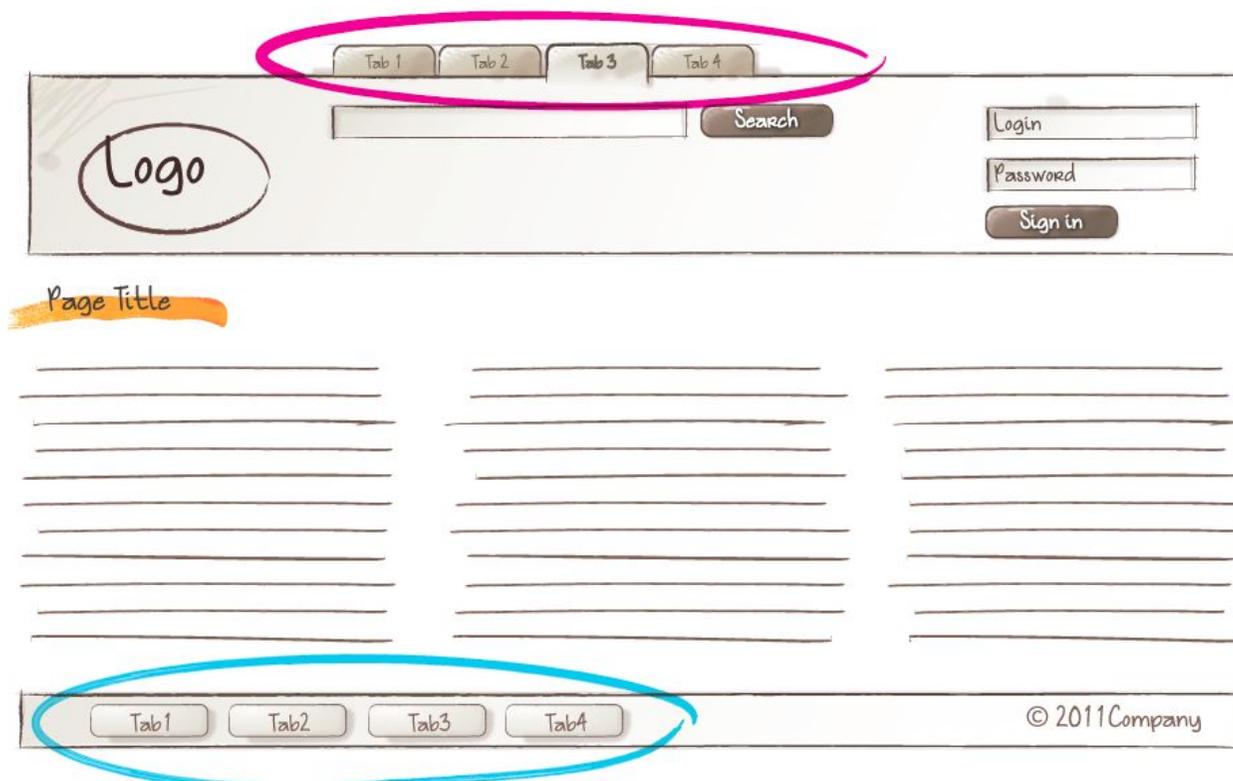
Составная часть блока, которая не может использоваться в отрыве от него. Элементы не существуют вне блока. Каждый элемент может принадлежать только одному блоку.



Модификатор

БЭМ-сущность, определяющая внешний вид блока или элемента.

Использование модификаторов опционально, количество — не ограничено. Блоку или элементу нельзя одновременно присвоить разные значения модификатора.



Пример DOM-дерева

```
<header class="header">
  <img class="logo">
  <form class="search-form">
    <input class="input">
    <button class="button">Submit</button>
  </form>
  <ul class="lang-switcher">
    <li class="lang-switcher__item">
      <a class="lang-switcher__link" href="/en">en</a>
    </li>
    <li class="lang-switcher__item">
      <a class="lang-switcher__link lang-switcher__link_active" href="/ru">ru</a>
    </li>
  </ul>
</header>
<main class="main">
  <div class="main__container">
    <!-- Content -->
  </div>
</main>
<footer class="footer">
  <span class="footer__copyright">
    &copy;CompanyName, 2021
  </span>
</footer>
```

SASS / SCSS, Less



SASS/SCSS Nesting

SASS - это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей. Имеет два синтаксиса SASS и SCSS

```
1 // style.sass
2 .header
3   display: flex
4   justify-content: space-between
5
6 .lang-switcher
7   display: flex
8   align-items: center
9   list-style: none
10  margin: 0
11
12  &_items
13    padding: 0 4px
14
15  &_link
16    color: black
17
18  &_active
19    color: blue
20
21 .main
22   display: flex
23   flex-direction: column
24
25   &_container
26     align-self: center
27     width: 100%
28     max-width: 1280px
29     padding: 0 16px
30
```

```
1 // style.scss
2 .header {
3   display: flex;
4   justify-content: space-between;
5 }
6
7 .lang-switcher {
8   display: flex;
9   align-items: center;
10  list-style: none;
11  margin: 0;
12
13  &_items {
14    padding: 0 4px;
15  }
16
17  &_link {
18    color: black;
19  }
20
21  &_active {
22    color: blue;
23  }
24 }
25
26 .main {
27   display: flex;
28   flex-direction: column;
29
30   &_container {
31     align-self: center;
32     width: 100%;
33     max-width: 1280px;
34     padding: 0 16px;
35   }
36 }
```

```
1 /* style.css */
2 .header {
3   display: flex;
4   justify-content: space-between;
5 }
6
7 .lang-switcher {
8   display: flex;
9   align-items: center;
10  list-style: none;
11  margin: 0;
12 }
13
14 .lang-switcher__items {
15   padding: 0 4px;
16 }
17
18 .lang-switcher__link {
19   color: black;
20 }
21
22 .lang-switcher__link_active {
23   color: blue;
24 }
25
26 .main {
27   display: flex;
28   flex-direction: column;
29 }
30
31 .main__container {
32   align-self: center;
33   width: 100%;
34   max-width: 1280px;
35   padding: 0 16px;
36 }
```

SASS/SCSS Миксины

Миксины (часто используется название примеси) позволяют определить стили, которые могут быть использованы повторно в любом месте документа без необходимости прибегать к несемантическим классам вроде `.flex-column`. Миксины также могут содержать целые CSS правила или что-либо другое, разрешённое в Sass документе. Они даже могут принимать аргументы, что позволяет создавать большое разнообразие стилей при помощи небольшого количества миксинов.

```
1 // style.scss
2 @mixin flex-column {
3   display: flex;
4   flex-direction: column;
5 }
6
7 @mixin border ($color, $width: 1px) {
8   border: {
9     color: $color;
10    width: $width;
11    style: dashed;
12  }
13 }
14
15 .block-1 {
16   @include flex-column;
17   @include border(■ red);
18 }
19
20 .block-2 {
21   @include border(■ green, 2px);
22 }
```

```
1 /* style.css */
2 .block-1 {
3   display: flex;
4   flex-direction: column;
5   border-color: ■ red;
6   border-width: 1px;
7   border-style: dashed;
8 }
9
10 .block-2 {
11   border-color: ■ green;
12   border-width: 2px;
13   border-style: dashed;
14 }
```

SASS/SCSS Переменные

```
1 // style.scss
2 $module-name: block;
3 $base-indent: 4px;
4
5 .#{ $module-name } {
6     display: flex;
7     flex-direction: column;
8     padding-bottom: $base-indent * 4;
9
10     &:hover {
11         .#{ $module-name } {
12             &__image {
13                 transform: scale(1.1);
14             }
15         }
16     }
17
18     &__image {
19         width: 100%;
20         transform: scale(1);
21         transition: transform 0.3s ease;
22     }
23
24     &__title,
25     &__description {
26         margin-top: $base-indent * 4;
27         padding: 0 $base-indent * 3;
28     }
29 }
```

```
1 /* style.css */
2 .block {
3     display: flex;
4     flex-direction: column;
5     padding-bottom: 16px;
6 }
7 .block:hover .block__image {
8     transform: scale(1.1);
9 }
10
11 .block__image {
12     width: 100%;
13     transform: scale(1);
14     transition: transform 0.3s ease;
15 }
16
17 .block__title, .block__description {
18     margin-top: 16px;
19     padding: 0 12px;
20 }
```

PostCSS



АКАДЕМИЯ

PostCSS

Инструмент для автоматизации рутинных операции в CSS, с помощью плагинов написанных на JavaScript.

```
1 // header.css
2 .header {
3     display: flex;
4     align-items: center;
5     justify-content: center;
6     color: #d2d2d2;
7     background-color: #2d2d2d;
8     padding: 8px 16px 12px;
9
10    &__title {
11        font-size: 1.2em;
12        font-weight: bold;
13        text-align: center;
14
15        @media all and (min-width: 1000px) {
16            font-size: 1.5em;
17        }
18
19        @media all and (min-width: 540px) {
20            font-size: 1.3em;
21        }
22    }
23 }
```

```
1 /* header.css */
2 .header {
3     display: -webkit-flex;
4     display: -ms-flexbox;
5     display: flex;
6     -webkit-align-items: center;
7     -ms-flex-align: center;
8     align-items: center;
9     -webkit-justify-content: center;
10    -ms-flex-pack: center;
11    justify-content: center;
12    color: #d2d2d2;
13    background-color: #2d2d2d;
14    padding: 8px 16px 12px
15 }
16
17 .header__title {
18     font-size: 1.2em;
19     font-weight: 700;
20     text-align: center
21 }
22
23 @media (min-width: 540px) {
24     .header__title {
25         font-size: 1.3em
26     }
27 }
28
29 @media (min-width: 1000px) {
30     .header__title {
31         font-size: 1.5em
32     }
33 }
```

PostCSS популярные плагины

1. **autoprefixer** – автоматически добавляет вендорные префиксы в соответствии с требованиями конфигурации
2. **postcss-preset-env** - преобразует современный CSS в то, что может понять большинство браузеров, определяя полифиллы, которые необходимы, на основе целевых браузеров или сред выполнения
3. **postcss-nested** - плагин для развертывания вложенных правил, аналогично Sass nesting
4. **postcss-import** - добавляет поддержку @import, для подключения локальных файлов при сборке проекта
5. **postcss-flexbugs-fixes** - исправляет известные баги при использовании flex
6. **postcss-animation** - позволяет использовать отдельные анимации из библиотеки animate.css без подключения библиотеки целиком
7. **postcss-sort-media-queries** - сортирует медиа выражения в соответствии с требованиями mobile first или desktop first
8. **cssnano** – оптимизирует размер CSS файла



CSS Custom Properties



CSS Custom Properties примеры использования

```
1 /* style.css */
2 .block {
3   --color-bg: red;
4   background-color: var(--color-bg);
5 }
```

Объявление и использование

```
1 /* style.css */
2 :root {
3   --color-bg: red;
4 }
5
6 .block {
7   background-color: var(--color-bg);
8 }
```

Глобальные переменные

```
1 /* style.css */
2 :root {
3   --color-bg: red;
4 }
5
6 .block {
7   color: var(--color-text, black);
8 }
```

Значение по умолчанию

```
1 /* style.css */
2 :root {
3   --hello: 'Hello';
4 }
5
6 .block::before {
7   content: var(--hello)' world!';
8 }
```

Конкатенация строк

CSS Custom Properties примеры использования

```
1  /* style.css */
2  :root {
3    --color-button: #f00;
4  }
5
6  .button {
7    background-color: var(--color-button);
8  }
9
10 .button:hover {
11   --color-button: #00f;
12 }
```

Смена цвет кнопки при наведении мышью

```
1  /* style.css */
2  :root {
3    --default-font-size: 16px;
4  }
5
6  .header {
7    font-size: var(--default-font-size);
8  }
9
10 @media all and (min-width: 640px) {
11   .header {
12     font-size: calc(var(--default-font-size) + 1vw);
13   }
14 }
```

Можно использовать в вычислениях с использованием **calc**

```
1  /* style.css */
2  :root {
3    --color-brand: #d88509;
4  }
```

Использование переменных в SVG

```
1  <!-- index.html -->
2  <svg>
3    <path fill="var(--color-brand)"/>
4  </svg>
```

CSS Custom Properties доступ из JS

```
1  /* style.css */
2  body {
3    --color-brand: #d88509;
4  }
5
6  .block {
7    background-color: var(--color-brand);
8  }
```

Определяем переменную и используем ее в CSS

```
1  // main.js
2  const block = document.querySelector('.block');
3
4  // получение текущего значения
5  const propValue = getComputedStyle(block).getPropertyValue('--color-brand');
6
7  // Установка нового значения
8  block.style.setProperty('--color-brand', '#f00');
9
```

Получений значения переменной из JS и изменение

CSS Custom Properties поддержка

Can I use custom properties ? [Settings](#)

2 results found

Caniuse (1) MDN (1)

CSS Variables (Custom Properties) CR

★ Permits the declaration and usage of cascading variables in stylesheets.

Usage % of all users

- Russian Federation 82.88% + 0.2% = 83.08%
- Global 94.85% + 0.21% = 95.06%

Current aligned Usage relative Date relative Filtered All

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	12-14		4-47	3.1-9	10-34	3.2-9.2										
	15	2-30	48	9.1	35	9.3							4			
6-10	16-87	31-84	49-88	10-13.1	36-72	10-13.7		2.1-4.4.4	12-12.1				5-12.0			
11	88	85	89	14	73	14.4	all	81	59	88	85	12.12	13.0	10.4	7.12	2.5
		86-87	90-92	TP												

Notes Test on a real browser Known issues (4) Resources (9) Feedback

¹ Enabled through the "Experimental Web Platform features" flag in `chrome://flags`

² Partial support is due to bugs present (see known issues)

Необходимо выполнить

1. Найти информацию и ознакомиться с методологиями: CSS modules, Atomic, OOCSS, SMACSS
2. Сверстать страницу с использованием BEM и CSS custom properties



Ссылки

1. BEM
<https://ru.bem.info/methodology/>
2. SASS документация
<https://sass-lang.com/>
<https://sass-scss.ru/>
3. Less
<http://lesscss.org/>
4. PostCSS
<https://postcss.org/>
<https://www.postcss.parts/>



Всем спасибо!

Лига –
лучший
старт
карьеры!

Мы в Лиге!

Умножай
знания –
верь в мечту!

Каждый
день – новый
челлендж!

