

Методология БЭМ

Цели занятия

- узнать что такое БЭМ и для чего он нужен;
- разобрать основные принципы и правила БЭМ;

Полезные материалы

- [Справочник CSS](#)
- [Методология БЭМ](#)
- [Часто задаваемые вопросы по БЭМ](#)

Что такое БЭМ?

Методология БЭМ создана в Яндексе для разработки сайтов, которые надо делать быстро, а поддерживать долгие годы. Она позволяет создавать расширяемые и повторно используемые компоненты интерфейса.

БЭМ (Блок, Элемент, Модификатор) — компонентный подход к веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки. Он позволяет легко и быстро разрабатывать интерфейсы любой сложности и повторно использовать существующий код, избегая «Copy-Paste».

Особенно важно знать и использовать БЭМ в командной разработке, чтобы разные разработчики говорили “на одном языке”, работали с классами одинаково и могли быстро разобраться в коде своего коллеги.

Блок

Функционально независимый компонент страницы, который может быть повторно использован в разных её частях или легко перенесён с одного места на другое. В HTML блоки представлены атрибутом `class`.

Особенности:

- Название блока характеризует смысл («что это?» — «меню»: `menu`, «кнопка»: `button`), а не состояние («какой, как выглядит?» — «красный»: `red`, «большой»: `big`).

```
<!-- Верно. Семантически осмысленный блок `error` -->
<div class="error"></div>
```

```
<!-- Неверно. Описывается внешний вид -->
<div class="red-text"></div>
```

- Блок не должен влиять на свое окружение, т. е. блоку не следует задавать внешнюю геометрию (в виде отступов, границ, влияющих на размеры) и позиционирование.
- В CSS по БЭМ также не рекомендуется использовать селекторы по тегам или `id`.
- Блоки можно вкладывать друг в друга, причём допускается любой уровень вложенности блоков

```
<!-- Блок `header` →
<header class="header">
  <!-- Вложенный блок `logo` →
  <div class="logo"></div>

  <!-- Вложенный блок `search-form` →
  <form class="search-form"></form>
</header>
```

Элемент

Составная часть блока, которая **не может** использоваться в отрыве от него.

Особенности:

- Название элемента характеризует смысл («что это?» — «пункт»: `item`, «текст»: `text`), а не состояние («какой, как выглядит?» — «красный»: `red`, «большой»: `big`).
- Структура полного имени элемента соответствует схеме: `имя-блока__имя-элемента`. Имя элемента отделяется от имени блока двумя подчеркиваниями (`__`).

```
<!-- Блок `search-form` →  
<form class="search-form">  
  <!-- Элемент `input` блока `search-form` →  
  <input class="search-form__input">  
  
  <!-- Элемент `button` блока `search-form` →  
  <button class="search-form__button">Найти</button>  
</form>
```

Правила работы с элементами

Вложенность

Элементы можно вкладывать друг в друга, причём допускается любая вложенность элементов. Но элемент — всегда часть блока, а

не другого элемента. Это означает, что в названии элементов нельзя прописывать иерархию вида `block__elem1__elem2`.

`<!-- Верно. Структура полного имени элементов соответствует схеме: `имя-блока__имя-элемента` -->`

```
<form class="search-form">
  <div class="search-form__content">
    <input class="search-form__input">
    <button class="search-form__button">Найти</button>
  </div>
</form>
```

`<!-- Неверно. Структура полного имени элементов не соответствует схеме: `имя-блока__имя-элемента` -->`

```
<form class="search-form">
  <div class="search-form__content">
    <!-- Рекомендуется: `search-form__input` -->
    <input class="search-form__content__input">

    <!-- Рекомендуется: `search-form__button` -->
    <button
class="search-form__content__button">Найти</button>
  </div>
</form>
```

Блок может иметь вложенную структуру элементов в DOM-дереве:

```
<div class="block">
  <div class="block__elem1">
    <div class="block__elem2">
      <div class="block__elem3"></div>
    </div>
  </div>
</div>
```

Однако эта же структура блока в методологии БЭМ всегда будет представлена плоским списком элементов:

```
.block {}
.block__elem1 {}
.block__elem2 {}
.block__elem3 {}
```

Это позволяет изменять DOM-структуру блока без внесения правок в код каждого отдельного элемента:

```
<div class="block">
  <div class="block__elem1">
    <div class="block__elem2"></div>
  </div>

  <div class="block__elem3"></div>
</div>
```

Структура блока меняется, а правила для элементов и их названия остаются прежними.

Принадлежность

Элемент — всегда часть блока и не должен использоваться отдельно от него.

```
<!-- Верно. Элементы лежат внутри блока `search-form` -->
<!-- Блок `search-form` -->
<form class="search-form">
  <!-- Элемент `input` блока `search-form` -->
  <input class="search-form__input">

  <!-- Элемент `button` блока `search-form` -->
  <button class="search-form__button">Найти</button>
</form>
```

```
<!-- Неверно. Элементы лежат вне контекста блока `search-form` -->
<!-- Блок `search-form` -->
<form class="search-form"></form>

<!-- Элемент `input` блока `search-form` -->
<input class="search-form__input">

<!-- Элемент `button` блока `search-form` -->
<button class="search-form__button">Найти</button>
```

Необязательность

Элемент — необязательный компонент блока. Не у всех блоков должны быть элементы.

```
<!-- Блок `search-form` -->
<div class="search-form">
  <!-- Блок `input` -->
  <input class="input">

  <!-- Блок `button` -->
  <button class="button">Найти</button>
</div>
```

Модификатор

Сущность, определяющая внешний вид, состояние или поведение блока либо элемента.

Особенности:

- Название модификатора характеризует внешний вид («какой размер?», «какая тема?» и т. п. — «размер»: `size_s`, «тема»: `theme_islands`), состояние («чем отличается от прочих?» — «отключен»: `disabled`, «фокусированный»: `focused`) и поведение («как ведет себя?», «как взаимодействует с пользователем?» — «направление»: `directions_left-top`).
- Имя модификатора отделяется от имени блока или элемента одним подчеркиванием (`_`).

Булевы модификаторы

Используют, когда важно только наличие или отсутствие модификатора, а его значение несущественно. Например, «отключен»: `disabled`. Считается, что при наличии булевого модификатора у сущности его значение равно `true`.

Структура полного имени модификатора соответствует схеме:

- имя-блока_имя-модификатора;
- имя-блока__имя-элемента_имя-модификатора.

```
<!-- Блок `search-form` имеет булевый модификатор `focused` -->
<form class="search-form search-form_focused">
  <input class="search-form__input">

  <!-- Элемент `button` имеет булевый модификатор `disabled` -->
  <button class="search-form__button
search-form__button_disabled">Найти</button>
</form>
```

Модификаторы ключ-значение

Используют, когда важно значение модификатора. Например, «меню с темой оформления islands»: menu_theme_islands.

Структура полного имени модификатора соответствует схеме:

- имя-блока_имя-модификатора_значение-модификатора;
- имя-блока__имя-элемента_имя-модификатора_значение-модификатора.

```
<!-- Блок `search-form` имеет модификатор `theme` со значением `islands` -->
<form class="search-form search-form_theme_islands">
  <input class="search-form__input">

  <!-- Элемент `button` имеет модификатор `size` со значением `m` -->
  <button class="search-form__button
search-form__button_size_m">Найти</button>
</form>
```

