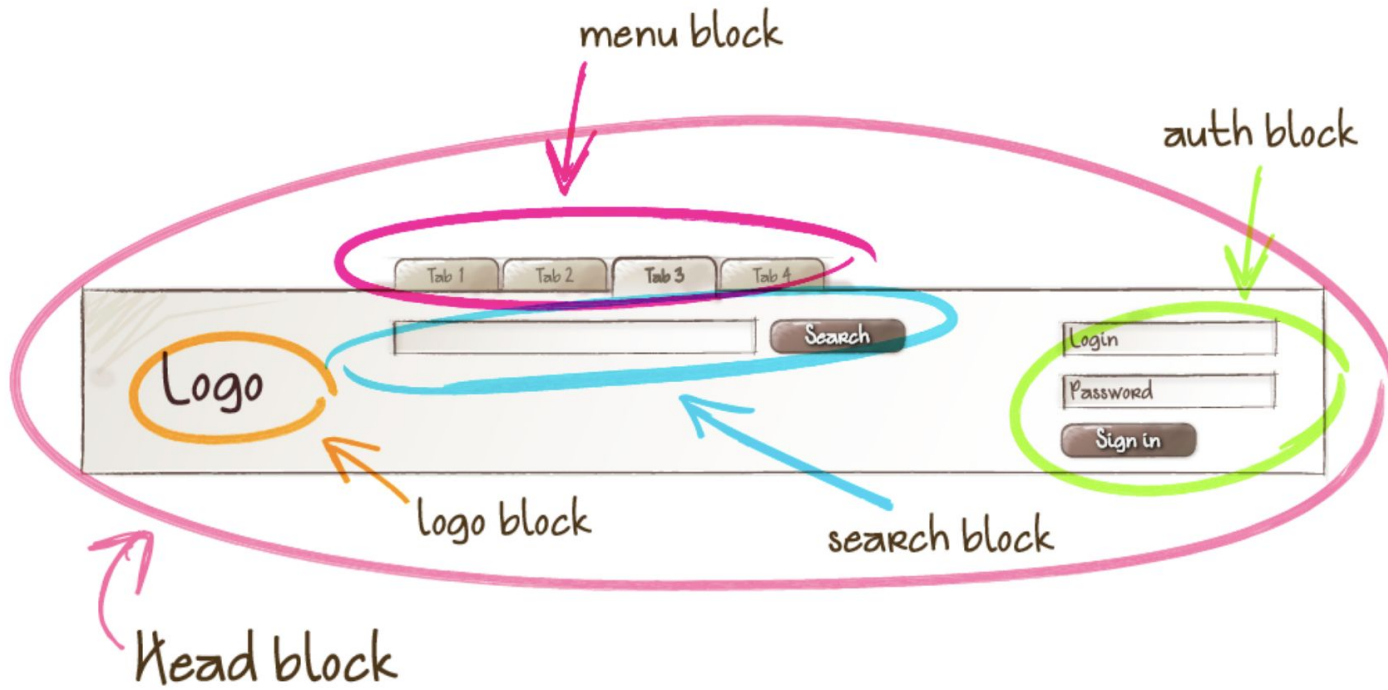




Лекция №10

CSS МЕТОДОЛОГИИ
BEM





Tab 1

Tab 2

Tab 3

Tab 4

Logo

Search

Sign in

Tab 1

Tab 2

Tab 3

Tab 4

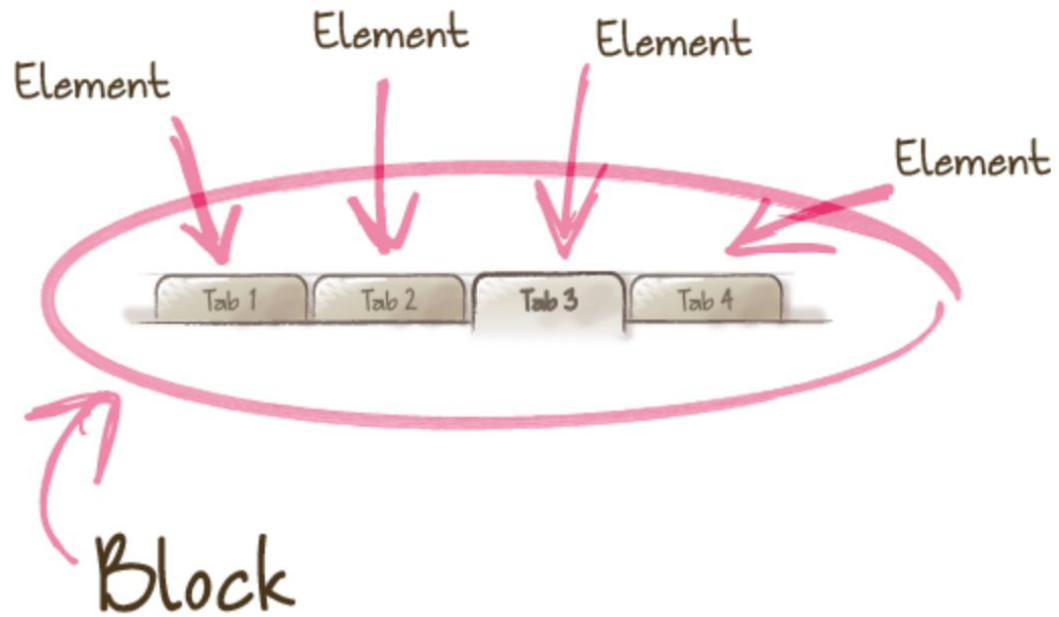
Sign in

Search

Logo









Введение

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».





Блок

Функционально независимый компонент страницы, который может быть повторно использован. 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`).






Пример

```
<!-- 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`.

Таким образом обеспечивается независимость, при которой возможно повторное использование или перенос блоков с места на место.



Вложенность

- Блоки можно вкладывать друг в друга.
- Допустима любая вложенность блоков.

Пример

```
<!-- 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 (`__`).



Пример

```
<!-- 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>
```





Вложенность

- Элементы можно вкладывать друг в друга.
- Допустима любая вложенность элементов.
- 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` или `search-form__content-input`

-->

<input class="search-form__content__input">

<!--

Рекомендуется:

`search-form__button` или `search-form__content-button`

-->

<button class="search-form__content__button">Найти</button>

</div>

</form>





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.

Пример

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



Когда создавать блок, когда — элемент?

Создавайте блок

Если фрагмент кода может использоваться повторно и не зависит от реализации других компонентов страницы.

Создавайте элемент

Если фрагмент кода не может использоваться самостоятельно, без родительской сущности (блока).

Исключение составляют элементы, реализация которых для упрощения разработки требует разделения на более мелкие части — подэлементы. В БЭМ-методологии [нельзя создавать элементы элементов](#). В подобном случае вместо элемента необходимо создавать служебный блок.



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 (`_`).



Типы модификаторов

Булевый

- 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>
```



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>
```





Модификатор нельзя использовать самостоятельно

С точки зрения БЭМ-методологии модификатор не может использоваться в отрыве от модифицируемого блока или элемента. Модификатор должен изменять вид, поведение или состояние сущности, а не заменять ее.



```
<!-- Верно. Блок `search-form` имеет модификатор `theme` со значением `islands`-->  
<form class="search-form search-form_theme_islands">  
  <input class="search-form__input">  
  
  <button class="search-form__button">Найти</button>  
</form>
```

```
<!-- Неверно. Отсутствует модифицируемый класс `search-form` -->  
<form class="search-form_theme_islands">  
  <input class="search-form__input">  
  
  <button class="search-form__button">Найти</button>  
</form>
```



Микс

Прием, позволяющий использовать разные БЭМ-сущности на одном DOM-узле.

Миксы позволяют:

- совмещать поведение и стили нескольких сущностей без дублирования кода;
- создавать семантически новые компоненты интерфейса на основе имеющихся.

Пример

```
<!-- Блок `header` -->
<div class="header">
  <!-- К блоку `search-form` примиксован элемент `search-form` блока `header`-->
  <div class="search-form header__search-form"></div>
</div>
```



Микс

Прием, позволяющий использовать разные БЭМ-сущности на одном DOM-узле.

Миксы позволяют:

- совмещать поведение и стили нескольких сущностей без дублирования кода;
- создавать семантически новые компоненты интерфейса на основе имеющихся.

Пример

```
<!-- Блок `header` -->  
<div class="header">  
  <!-- К блоку `search-form` примиксован элемент `search-form` блока `header`-->  
  <div class="search-form header__search-form"></div>  
</div>
```





Соглашение по именованию

Имя БЭМ-сущности уникально. Во всех технологиях (CSS, JavaScript, HTML) одна и та же БЭМ-сущность всегда называется одинаково. Основная идея соглашения по именованию — вложить смысл в имена и сделать их максимально информативными для разработчика.



Можно сравнить одно и тоже имя CSS-селектора, написанное разными способами:

- `menuitemvisible`
- `menu-item-visible`
- `menuItemVisible`

Чтобы понять смысл первого имени, нужно вчитаться в каждое слово. В последних двух примерах имя явно разделяется на логические части. Но ни одно из имен пока не помогает точно определить, что `menu` — это блок, `item` — элемент, а `visible` — модификатор. Чтобы имена сущностей были однозначными и понятными, в БЭМ были разработаны правила формирования имен БЭМ-сущностей.



Правила формирования имен

block-name__elem-name_mod-name_mod-val

- Имена записываются латиницей в нижнем регистре.
- Для разделения слов в именах используется дефис (-).
- Имя блока задает **пространство имен** для его элементов и модификаторов.
- Имя элемента отделяется от имени блока двумя подчеркиваниями (__).
- Имя модификатора отделяется от имени блока или элемента одним подчеркиванием (_).
- Значение модификатора отделяется от имени модификатора одним подчеркиванием (_).
- Значение булевых модификаторов в имени не указывается.



Стиль Two Dashes

`block-name__elem-name--mod-name--mod-val`

- Имена записываются латиницей в нижнем регистре.
- Для разделения слов в именах БЭМ-сущностей используется дефис (-).
- Имя элемента отделяется от имени блока двумя подчеркиваниями (__).
- Булевые модификаторы отделяются от имени блока или элемента двумя дефисами (--).
- Значение модификатора отделяется от его имени двумя дефисами (--).



Стиль CamelCase

```
blockName__elemName_modName_modVal
```

- Имена записываются латиницей.
- Каждое слово внутри имени пишется с заглавной буквы.
- Разделители элементов и модификаторов совпадают с классической схемой.





<https://ru.bem.info/methodology/>





Спасибо!

