Содержание:

Введение

БЭМ (аббревиатура от слов — Блок, Элемент и Модификатор) — это методология разработки программ и интерфейсов, способ описания сущностей, не привязанный к конкретным технологиям реализации.

- Блок — это отдельный компонент приложения. Он независим от других блоков и может содержать в себе другие блоки и элементы.
- Элемент — это часть блока, отвечающая за отдельную функцию. Он не имеет смысла в отрыве от блока.
- Модификатор — это свойство блока или элемента, отвечающее за его внешний вид или поведение. Модификаторы описывают состояние блока или элемента.

БЭМ предоставляет абстракцию над DOM-деревом. Блоки независимы друг от друга и инкапсулируют в себе всю функциональность и элементы. Не важно, какими HTML-тегами будет реализован блок — div или form, вы всегда можете изменить это или добавить дополнительные обёртки. Любые изменения не должны оказывать влияние на остальные блоки. Мы описываем приложение компонентами интерфейса, а не HTML-тегами.

Каждый блок лежит в своей папке в файловой системе, в которой сложены все технологии, описывающие блок, его элементы и модификаторы.

Основные понятия

Основные понятия

«Блок», «элемент» и «модификатор» — основные термины БЭМ. Это необходимые и достаточные понятия для описания интерфейса любой сложности.

Блок

Блок — это независимый интерфейсный компонент. Блок может быть простым или составным (содержать другие блоки). При создании блока нужно обеспечивать возможность его использования в любом месте web-страницы, а также повторения в том же самом месте страницы (родительском элементе). Блок должен включать в себя всю реализацию, необходимую для представления части интерфейса, которую он выражает.

Элемент

Элемент — это составная часть блока. Элементы контекстно-зависимы: они имеют смысл только в рамках своего блока. Элемент — не обязательная составляющая блока, небольшие блоки обходятся без элементов.

Модификатор

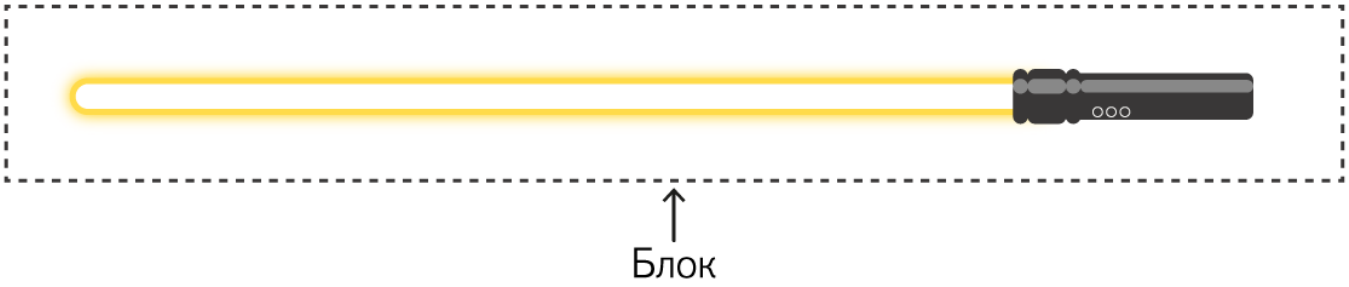
Модификатор — это свойство блока или элемента, задающее изменения в их внешнем виде или поведении. Модификатор может быть булевым (например, button_big) или парой ключ-значение (например, menu_type_bullet, menu_type_numbers). У блока или элемента может быть несколько модификаторов одновременно.

1.1 Блок

Логически и функционально независимый компонент страницы. Блок полностью самодостаточен: у него может быть свое поведение, шаблоны, стили, документация и не только. Блоки могут использоваться в любом месте страницы, повторно, даже в другом проекте.

Возможности блоков:

- Вложенная структура
- Свободное перемещение
- Повторное использование



Вложенная структура

Блоки можно вкладывать в любые другие блоки.

Например, блок head может содержать логотип (logo), форму поиска (search) и блок авторизации (auth).

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.

Так, например, логотип и форму авторизации можно поменять местами. При этом вносить изменения в CSS или JavaScript-код блоков не нужно.

Повторное использование

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

1.2 Элемент

Элемент — это составная часть блока. Элементы контекстно-зависимы: они имеют смысл только в рамках своего блока. Элемент — не обязательная составляющая блока, небольшие блоки обходятся без элементов.

Часть блока, которая не может использоваться в отрыве от него и имеет смысл только в рамках своего родителя. Элементы могут быть обязательными и опциональными.

Работая с элементами, важно помнить правило: не рекомендуется создавать элементы элементов. Если вложить один элемент в другой, будет невозможно изменить внутреннюю структуру блока: элементы нельзя будет поменять местами, удалить или добавить без корректировки существующего кода.

1.3 Модификатор

Модификатор — это свойство блока или элемента, задающее изменения в их внешнем виде или поведении. Модификатор может быть булевым (например, button_big) или парой ключ-значение (например, menu_type_bullet, menu_type_numbers). У блока или элемента может быть несколько модификаторов одновременно.

Свойство блока или элемента, которое меняет их внешний вид, состояние или поведение.
Модификатор имеет имя и может иметь значение. Использование модификаторов опционально. У блока/элемента может быть несколько разных модификаторов одновременно.

Так, например, с помощью модификатора можно изменить не только цвет меча, но и его функциональность (как показано в случае с красным мечом):

Применение БЭМ в различных web-технологиях

2.1 HTML/CSS

В HTML/CSS блоки, элементы и модификаторы представлены в виде CSS-классов, названных согласно правилам именования (naming convention). Несколько блоков могут быть расположены на одном и том же DOM-узле, в этом случае DOM-узлу назначается 2 CSS-класса. На одном DOM-узле также могут быть одновременно расположены блок и элемент другого блока.

2.1.2 Правила именования БЭМ-классов от Яндекса

CSS-класс блока соответствует имени блока. Для разделения слов в сложных именах блоков используется дефис.

...

<ul class="menu">...

...

...

CSS-класс элемента содержит имя блока и имя элемента, разделённые двумя знаками underscore.

```
<div class="header">
<div class="header__bottom">...</div>
</div>
<ul class="menu">
<li class="menu__item">...</li>
</ul>
<span class="button">
<input class="button__control">...</input>
</span>
<div class="tabbed-pane">
<div class="tabbed-pane__panel">...</div>
</div>
```

CSS-класс модификатора содержит имя блока и имя модификатора, разделённые одним знаком underscore. В том случае, если модификатор — это пара ключ-значение, они тоже разделяются знаком underscore. Для модификатора элемента в CSS-классе сохраняются и имя блока, и имя элемента. CSS-класс модификатора используется в паре с классом своего блока (или элемента).

```
<div class="header header_christmas">...</div> <!-- Christmas edition of the header -->
<ul class="menu">
<li class="menu__item menu__item_current">...</li>
</ul>
<span class="button button_theme_night">...</span>
<div class="tabbed-pane tabbed-pane_disabled">...</div>
```

2.1.2 Правила именования БЭМ-классов от Гарри Робертса

Альтернативные правила именования были предложены Гарри Робертсом. Он советует использовать 2 дефиса для разделения имён блока и модификатора.

... <!-- Christmas edition of the header -->

<ul class="menu">

<li class="menu__item menu__item--current">...

...

...

2.2 JavaScript

В БЭМ JavaScript работает с абстрактной структурой блоков-элементов и модификаторов, не обращаясь к лежащим за ним DOM-узлам и их CSS-классам напрямую. Кроме того, для идентификации DOM-узлов не используются дополнительные CSS-классы "специально для JavaScript". Для обеспечения такой возможности используется фреймворк или собственный набор хелперов.

2.2.1 Хелперы для работы с БЭМ-структурой

Так, если каждому блоку с JavaScript-функциональностью соответствует объект, его методы позволяют:

обращаться к вложенным элементам:

// предположим, что blockObj указывает на объект блока

blockObj.elem('panel'); // возвращает элементы

работать с модификаторами

// предположим, что blockObj указывает на объект блока

blockObj.setMod('disabled'); // устанавливает модификатор

blockObj.delMod('disabled'); // удаляет модификатор

2.2.2 Реакция на установку/удаление модификаторов

Поскольку модификатор отражает состояние блока, при назначении модификатора блок или элемент должен быть приведён в соответствующее состояние. Для изменения внешнего вида достаточно назначения CSS-класса модификатора. В более сложных случаях приведение блока в нужное состояние требует JavaScript-функциональности. Поэтому у используемого JavaScript-фреймворка должна быть возможность декларировать список действий, соответствующий модификатору.

BlockObj.on({

active: function() {

// do smth when active

},

disabled: function() {

// do something when disabled

}

});

2.3 Файловая структура проекта

На файловой системе блоки, элементы и модификаторы представлены в виде файлов своих реализаций в различных web-технологиях. Файлы, относящиеся к одному блоку, объединяют в один каталог.

2.3.1 Плоская структура

Самая простая структура проекта не предполагает вложенности в каталоге блоков:

button/

button.css

button.js

button.tpl

button__control.css

header/

header.css

header.tpl

header_christmas.css

tabbed-pane/

tabbed-pane.css

tabbed-pane.js

tabbed-pane.tpl

2.3.2 Вложенная структура

В больших проектах или библиотеках удобно использовать вложенную файловую структуру блока, где для элементов и модификаторов выделяются каталоги.

button/

control/

button__control.css

button.css

button.js

button.tpl

header/

_christmas/

header_christmas.css

header.css

header.tpl

tabbed-pane/

tabbed-pane.css

tabbed-pane.js

tabbed-pane.tpl

Заключение.

БЭМ-методология — это набор правил и рекомендаций по организации работы над проектом.

В какой-то момент мы отделили методологию от ее практической реализации — платформы.
БЭМ-платформа — это частный случай реализации общих принципов БЭМ-методологии. Так как все технологии создавались с учетом требований наших проектов и развивались постепенно, БЭМ-платформа наиболее полно охватывает все возможности, которые предоставляет БЭМ-методология.

Все части БЭМ-платформы интегрированы для совместной работы, но могут быть использованы и по отдельности. Каждая часть решает конкретную задачу и её можно настраивать под свой процесс и заменять на другие.

БЭМ-методология Методология разработана в компании Яндекс и широко используется в продуктах различных компаний. Она нашла применение в составе специально разработанного HTML5-фреймворка при редизайне и рефакторинге почтового сервиса mail.ru. БЭМ также использована в продукте Google, как Material

Design Lite, HTML5-фреймворке, наподобие Twitter Bootstrap, поддерживающим Material design.

Список литературы

1. Михеева, Е. В. Информационные технологии в профессиональной деятельности / Е. В. Михеева. – М.: Издательский центр «Академия», 2008. – 384 с
2. Кричалов, А. А. Компьютерный дизайн. Учебное пособие / А. А. кричалов. – Мн.: СТУ МГМУ, 2008 г. – 154 с.
3. https://ru.wikipedia.org
4. https://biblioclub.ru