

Содержание:

image not found or type unknown

Введение

Технологии (HTML, CSS, JavaScript), которые мы использовали, изменялись в зависимости от требований проекта, а принципы БЭМ должны были быть универсальны.

Мы сформулировали основные правила, по которым будут жить и развиваться наши проекты, и которые никак не будут зависеть от технологий и инструментов.

Чтобы ускорить разработку, необходимо было облегчить поддержку HTML и CSS отдельных компонентов страницы, сделать код менее связанным. Для этого мы разбили страницу на части. Так появилось новое понятие — блок. Блок мог состоять из различных элементов, которые не использовались вне самого блока. Состояния и поведение блока и элемента можно было задавать с помощью модификатора.

Это были три ключевых понятия, на которых основывалось большинство правил. Аббревиатура от трех слов **Б**лок, **Э**лемент и **М**одификатор стала названием методологии — БЭМ.

БЭМ - МЕТОДОЛОГИЯ

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

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 — 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 — 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.

Цель создания БЭМ

БЭМ предлагает общую семантическую модель для всех технологий, используемых во фронтэнд разработке .

Для чего нужна БЭМ - методология.

БЭМ-платформа — это частный случай реализации общих принципов БЭМ-методологии. Так как все технологии создавались с учетом требований 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.

Необходимые требования:

Разработчик должен понимать свой код (даже вернувшись к нему через год) и код любого программиста в команде БЭМ-проекта.

Любой блок кода может быть использован повторно: необходимо создать общую базу знаний и не писать каждый раз всё с нуля, а использовать готовые наработки.

Работая в одной команде, разработчики, менеджеры, дизайнеры и верстальщики должны называть одни и те же вещи одинаково. То есть говорить на одном языке.

Команды могут обмениваться специалистами для реализации какой-то конкретной функциональности.

Порог входа при переходе на новый проект должен быть снижен за счет одинаковой структуры организации всех БЭМ-проектов и одинаковых правил именования всех сущностей.

Используя понятия «блок», «элемент» и «модификатор» можно описать древовидную структуру документа. Такое описание называется BEM tree и является семантическим представлением интерфейса, абстракцией над DOM tree. В основу того, что мы сейчас называем БЭМом, легла идея независимых блоков, которую Виталий Харисов сформулировал и презентовал в 2007 году на первой российской конференции по фронтенду. Это было настолько давно, что тогда даже слова «фронтенд» ещё не было, тогда это называлось клиент-сайд.

Идея была в том, чтобы ограничить возможности CSS для более предсказуемых результатов. Использовать минимум глобальных стилей и каждый отдельный элемент страницы делать блоком со своим уникальным классом и стилями, которые полностью его описывают. Селекторы по элементам и ID, хрупкие связки вложенности — всё это заменялось на простые селекторы по классам. Каждый класс в стилях — это блок. Благодаря этому блоки можно легко менять местами, вкладывать друг в друга и не бояться конфликтов или влияния.

Потом появились абсолютно независимые блоки (АНБ), где у элементов внутри есть свой префикс с именем родителя, а состояния блоков снова дублируют класс, но уже с постфиксом. Подход обрёл черты современного БЭМа, одна из которых — многословность классов.

Эта многословность гарантирует уникальность элементов и модификаторов в рамках одного проекта. За уникальностью имён блоков вы следите сами, но это довольно просто, если каждый блок описан в отдельном файле. Глядя на такой класс в HTML или CSS, можно легко сказать, что это, и к чему оно относится.

Каждая БЭМ-сущность должна иметь свой класс.

CSS-свойства для блоков, элементов и модификаторов описываются только через классы.

Для разделения слов в именах используется дефис (-).

Элемент отделяется от блока двумя подчеркиваниями (___).

Модификатор — одним (_).

Имена БЭМ-сущностей записываются с помощью цифр и латинских букв в нижнем регистре.

Блоки могут выполнять разные функции на странице. В зависимости от предназначения блока может меняться его реализация. Под реализацией в БЭМ понимают поведение, внешний вид, шаблоны, документацию к блоку, все виды тестов, картинки и так далее.

Для реализации блока используются различные технологии, например:

- поведение — JavaScript, CoffeeScript;
- внешний вид — CSS, Stylus, Sass;
- шаблоны — Jade, Handlebars, XSL, BEMHTML, VH;
- документация — Markdown, Wiki, XML.

Выбор технологий реализации не ограничен, разве только требованиями проекта. В новой организации файловой структуры каждая технология реализации представляет собой отдельный файл с соответствующим расширением. Все файлы реализации блока хранятся в директории этого блока.

Блок — отдельная директория в файловой системе. Имя блока и его директории совпадают. Реализация блока разделяется на отдельные файлы. Файлы, относящиеся к блоку, всегда находятся в его директории. Блок становится ключевым понятием БЭМ. Соответственно, изменяется и структура файловой системы.

Выделять общие части реализаций блоков на один уровень, а частные случаи (например, специфическую реализацию для отдельных сервисов) — на другой.

Разделять проект на платформы. Общую реализацию для всех платформ хранить на одном уровне, а платформу-специфичную — выносить на другой.

Избежать копирования кода и создания новых сущностей, если необходимо изменить уже существующую функциональность.

Если сравнить уровни со слоями, то базовый слой — это исходная реализация блока, а каждый последующий слой накладывается сверху и дополняет (наследует) или изменяет базовую реализацию.

Вывод

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. Более подробно о ней можно прочитать здесь.

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.

Список литературы

1. <https://habr.com/ru/company/yandex/blog/276035/>
2. <https://medium.com/@dedguran/%D0%B2%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B2-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BB%D0%BE%D0%B3%D0%B%D0%B1%D1%8D%D0%BC-e6b156e1f795>
3. <https://itnan.ru/post.php?c=1&p=276035>
4. <https://ru.wikipedia.org/wiki/%D0%91%D0%AD%D0%9C>
5. <https://webdesign.tutsplus.com/ru/articles/an-introduction-to-the-bem-methodology--cms-194031>