

Разметка по БЭМ

Выполнен на 0%

Введение

Углублённая теория

Предыстория

Частые проблемы из-за неправильной CSS-архитектуры проекта

Основные понятия

Свод правил БЭМ-а

БЭМ для CMS

Названия классов по БЭМ

БЭМ — это про компоненты

Темизация

Плюсы и минусы БЭМ

Методика

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, средний уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, сложный уровень

Кейс 8, сложный уровень

Кейс 9, сложный уровень

Главная / Моё обучение / Разметка по БЭМ / Углублённая теория /

Плюсы и минусы БЭМ

Плюсов в БЭМ больше, с них и начнём.

Плюсы

— Простота — низкий порог вхождения в технологию, даже новичок может использовать БЭМ в своём проекте.

— Гибкость — БЭМ-сущности формируются под нужды и специфику проекта. Нет чётких правил, что одни сущности отвечают только за сетки, а другие — только за темы (можно сравнить со *SMACSS*).

— Независимость — компоненты создаются так, чтобы их можно было легко переносить из проекта в проект без каких-либо доработок.

— Командная разработка интерфейса — монолит макета делится на независимые блоки, которые разрабатываются отдельными участниками команды, возможна параллельная работа, кто-то собирает эти блоки вместе.

— Лучшая производительность — из-за того, что БЭМ не использует вложенные селекторы, браузеры быстрее вычисляют специфичность селекторов и быстрее обрабатывают правило. Правда некоторые источники утверждают, что современные движки браузеров достаточно эффективны, и практически во всех сложных случаях (за исключением разве что самых экстремальных) выигрыш в скорости пренебрежительно мал.

— Понятность (человекочитаемость) — вопрос больших проектов, структура проекта прозрачна, всё что относится к блоку/компоненту собрано внутри одной папки. Простые селекторы помогают быстрее понять ваш код другим разработчикам и вам по прошествии некоторого промежутка времени.

— Меньше конфликтов при определении стилей для элемента — имена классов уникальные и не повторяются. Перебивать вес селекторов не нужно. Как правило, достаточно сделать модификатор (возможно, придётся использовать флаг `!important` только для внешних библиотек, чтобы перебить правило, которое мы не имеем права переписывать).

— Расширяемость — официальную документацию по БЭМу каждая команда может расширить своими мыслями и опытом набитых шишек. Хотя это уже не совсем БЭМ, а какой-то расширенный вариант БЭМ для вашей команды/компании.

А ещё БЭМ помогает новичкам сформировать компонентное мышление. Это облегчает расширяемость проекта, ведь все компоненты независимы, любой из них легко изменить, и эти изменения не затронут другие компоненты. А значит, меньше шансов что-либо сломать.

Минусы

— Увеличение объёма файлов со стилями (каждый DOM-узел с классом, плюс длинные названия классов и несколько классов для одного узла, возможно, дублирование стилей). Проблему можно решить за счёт минификации итоговых файлов со стилями и добавлением в сборку только нужных файлов.

— «Страшный» HTML-код (в разметке появляется куча классов). В исходный код страницы с разметкой пользователи заглядывают редко. Так что если вы раздумываете, использовать или не использовать БЭМ, то этот пункт вряд ли будет решающим.

— Сложности при реализации. В стилях каждый компонент — это отдельный файл (в CSS есть директива `@import`), а вот писать компонентно на «ванильном» HTML — это проблема. Пока что, к сожалению, нет подходящих инструментов для сборки. В таком процессе разработки тяжело сделать декомпозицию.

Ознакомились со статьёй?

Сохранить прогресс

Темизация

Методика



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Профессии

Фронтенд-разработчик

React-разработчик

Фулстек-разработчик

Бэкенд-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

PHP. Профессиональная веб-разработка

PHP и Yii. Архитектура сложных веб-сервисов

Node.js. Разработка серверов приложений и API

Анимация для фронтендеров

Вёрстка email-рассылок

Vue.js для опытных разработчиков

Регулярные выражения для фронтендеров

Шаблонизаторы HTML

Алгоритмы и структуры данных

Анатомия CSS-каскада

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Остальное

Написать нам

Мероприятия

Форум