

## Разметка по БЭМ

Выполнен на 0%

Введение

Углублённая теория

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 /

# Введение

Вы, наверное, ещё не знаете до конца, зачем же вам нужен БЭМ и что это такое.

БЭМ — это система «Блок — Элемент — Модификатор». Это один из подходов к компонентной вёрстке, и одна из самых распространённых в мире методологий вёрстки ([вот тут есть статистика использования разных методологий](#)). Если вы планируете верстать в настоящем рабочем процессе, то вы наверняка столкнётесь с БЭМ или другими методологиями, возможно, основанными на БЭМ.

## Что такое компонентная вёрстка?

Это такой подход к вёрстке, когда отдельные компоненты сайта создаются изолированно и могут использоваться в любом месте, вне зависимости от окружения и содержания.

Вот, например, у вас есть на сайте статистика. На главной странице она повествует о компании в целом, а на странице основного продукта — о его особенностях, и выглядят они чуть-чуть по-разному, но в общем — это одно и то же. И смысл компонентного подхода в том, чтобы эта статистика была совершенно независима от окружения (что вокруг неё? Нет разницы!) и могла использоваться и там, и тут без дополнительной разработки.

Основной принцип — как можно больше переиспользовать (*re-use*) код. Для этого стоит не верстать по несколько раз одно и то же.

## Почему так важно верстать компонентно?

Потому что это позволяет сделать структуру кода прозрачной. Например, у вас много страниц, и они все сложные, не просто «страница блога», а рейтинги, статистика, новости, баннеры, слайдеры, сайдбары, всплывающие окна, куча навигации. В такой ситуации разработка ресурса, скорее всего, становится делом целой команды.

Кто-то пришёл в проект с самого начала, кто-то присоединился позже. Что делать, если нужно исправить или дополнить какой-то участок? Какими методами искать? До компонентного подхода искали «на память». Кто-то помнил, где лежит этот кусок кода или тратил время на то, чтобы перелопатить весь стиливой файл или постараться найти все места, где используется *искомое*.

Но большие проекты лучше не привязывать к памяти отдельного разработчика, лучше, чтобы была какая-то системность, которую можно быстро объяснить новому сотруднику. Ещё лучше — если этот новый сотрудник заранее предполагает, что за система лежит в основе ресурса, тогда достаточно объяснить нюансы.

Впрочем, не очень большие проекты тоже лучше писать так, чтобы через год не было мучительно больно вспоминать, где и что написано, и почему так.

## Какие есть методологии?

« С самого начала у меня была какая-то тактика и я её придерживался. — Александр Несяев

Тут нужно сказать, что все, кого мы об этом спрашивали, говорили, что «нужно иметь **какую-то** методологию и придерживаться её». Так что посмотрим, каких методологий придерживаются чаще всего.

Это: [BEM](#), [Atomic CSS \(Utility-first CSS\)](#), [OOCSS](#), [SMACSS](#), [ITCSS](#).

Методологии вёрстки в действии

Если вам интересно посмотреть методологии вёрстки в действии, а не просто узнать их основные принципы, советуем заглянуть на этот [ресурс](#).

## Почему именно БЭМ?

Да, почему не одна из других систем?

Во-первых, БЭМ всё ещё одна из самых популярных систем в мире.

Во-вторых, он всё-таки требует и довольно детальных объяснений, как и что делать, и рассказа о его философии, и тренировки.

В-третьих, мы сами его используем.

В-четвёртых, это именно система, достаточно разветвлённая и в полном своём формате описывающая не только имена классов, но и файловую систему, и метод написания CSS и JS, и подключение скриптов и стилевых файлов.

В общем, это такая концепция, освоение которой поможет в последствие войти в почти любую методику без больших потерь, ведь вы научитесь работать с компонентным подходом и придерживаться одного подхода внутри всего проекта.

В этом навыке мы говорим не о полном *БЭМ-стеке*, а только о соглашении в наименовании классов. Первым, кто начал использовать в своих проектах схему наименования, основанную на принципах БЭМ, был [Гарри Робертс](#). Это привело к появлению новой методологии *BEMIT (BEM+ITCSS)*. Наименования CSS-классов по БЭМ используют многие компании, в том числе и *HTML Academy*.

## Ознакомились со статьёй?

Сохранить прогресс

Углублённая теория



### 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