



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) / [2. Методологии вёрстки](#) /

2.4. Введение

 ~ 4 минуты

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 нескольку 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 та 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 стилевой 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

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 стиливых файлов.

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

Прочитали главу?

Нажмите кнопку «Готово», чтобы сохранить прогресс.

Готово

⚠ Если вы обнаружили ошибку или неработающую ссылку, выделите ее и нажмите Ctrl + Enter

Поиск по материалам

Git

[Все материалы](#)

В самом начале



- ☐ Пройдите опрос
- ☐ Укажите персональные данные
- ☐ Изучите регламент
- ☐ Прочитайте FAQ
- ☐ Добавьте свой Гитхаб
- ☐ Выберите наставника
- ☐ Создайте проект

Мой наставник



[Выбрать наставника](#)

Работа с наставником

У вас осталось **10** из 10 консультаций.

[История](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

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

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696



Участник

© ООО «Интерактивные обучающие технологии», 2013–2023

