



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

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

📖 2.7. Основные понятия

🕒 ~ 7 минут

Блок

«Функционально независимый компонент страницы, который может быть повторно использован». © сайт bem.info.

Это такая часть страницы, которая может рассматриваться, разрабатываться и использоваться самостоятельно (как элемент дизайн-системы, как отдельная осмысленная единица интерфейса, как бизнес-сущность), которая имеет свой ограниченный код и не содержит никаких сведений о том, что находится за её границами.



Типичный макет с отдельными независимыми «блоками»

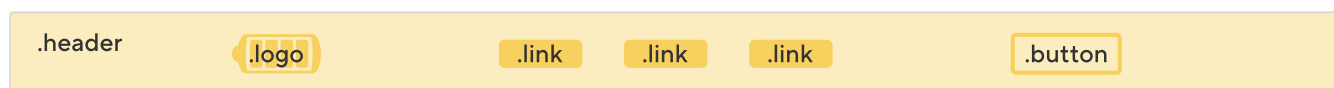
Шапка, которая есть на любой странице — наверняка блок. Если слайдер можно теоретически использовать не только на главной под шапкой — он будет блоком. Каталог — блок (*Новинки, С этим товаром покупают, Похожие товары, ...*). Карточка товара можно вытащить из каталога и использовать в другом месте. Кнопок много...

в разных местах — это блоки. Чем-то похоже на семантический тег `<article>`, но гораздо шире трактуется и не коррелирует с HTML напрямую.

То есть когда мы говорим о «компонентах», мы говорим об отдельных независимых блоках. Блок — это и есть компонент.

Элемент

Это часть, деталь блока, то, что вообще не может использоваться вне своего блока.



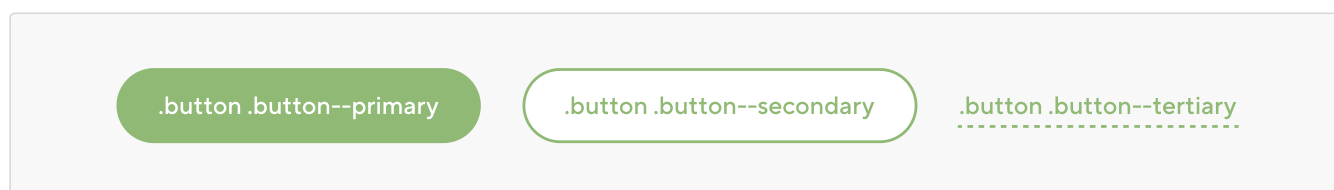
Типичный макет с отдельными независимыми блоками и их «элементами»

Вот например, пункт меню не может использоваться вне меню, хотя ссылка на одну из страниц, вероятно, может. Почему вероятно? Потому что в меню эта ссылка обычно используется как будто это кнопка, с увеличенной зоной клика и подчёркиванием по наведению. Если она примерно так же используется в других местах, пусть и с другим декором — это блок. А если на остальном сайте используются только строчные ссылки внутри текста — это по-прежнему элемент.

Ещё элемент — это сущность, которая определяет какую часть блока занимает HTML-элемент (место расположения, ширину/высоту, расстояние до соседей и так далее). Такой своеобразный солдат раскладки.

Модификатор

Это вариативность блока или элемента. Всегда только внешняя.



Несколько кнопок, есть обычные, а есть модифицированные

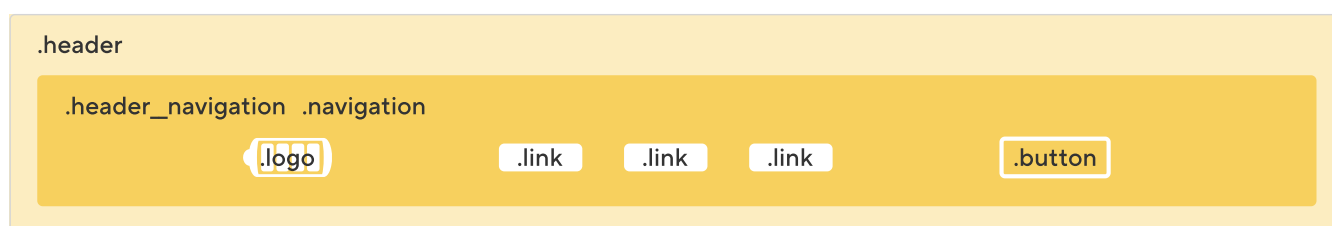
Например, у нас на сайте есть множество сущностей, которые означают пользовательское действие «кликни, и что-то произойдёт». Где-то это могут быть ссылки, где-то кнопки. Но у них общая функциональность, скорее всего они будут блоками, к которым мы допишем модификаторы, чтобы различать большие и малые, жёлтые и синие. Но внешняя составляющая для модификаторов — не главное, главное — это сама функциональность. Например, кнопка первичного действия и кнопка вторичного действия.

Задача верстальщика — определить, какие элементы интерфейса — независимые блоки, а какие — их элементы. Что будет базовым представлением блока, а что его модификацией.

Микс

Это приём, который позволяет использовать разные БЭМ-сущности применительно к одному HTML-элементу.

Один и тот же HTML-элемент может быть БЭМ-блоком и БЭМ-элементом одновременно. Мы же говорили, что это не связано с разметкой напрямую? Например, навигация по сайту наверняка будет блоком, но сеточные стили вроде выравнивания к ней нужно будет привязывать как к элементу шапки. Это называется миксом.



Навигация это одновременно и сам блок, и элемент страницы

За счёт миксов можно уменьшить количество HTML-элементов на странице, тем самым повысить её показатели по производительности.

Оптимизация страниц

Если невозможно использовать микс (например, из-за дизайна не получается применить сеточные и другие стили к одному и тому же HTML-элементу), то создаются отдельные обёртки.

БЭМ-дерево

Абстрактная структура страницы в блоках, элементах и модификаторах. Показывает вложенность блоков друг в друга, подчинение элементов блокам, наличие модификаторов.

Вид и формат, в котором мы выразим БЭМ-дерево, не важен: важнее всего показать именно вложенность блоков и элементов друг в друга, увидеть всю композицию, понять, как именно относятся друг к другу все блоки и элементы внутри страницы.

```
page /* это страница, блок */
  page__header page-header /* это микс, элемент страницы - шапка, и отдель.....
```

```

блок шапки */
page__nav container /* микс, навигация как элемент страницы и сеточный
контейнер */
site-navigation /* навигация - блок */
site-navigation__item /* элемент навигации */
site-navigation__link /* ещё элемент навигации */
site-navigation__item
site-navigation__link site-navigation__link--active /* элемент с
модификатором */

```

Для построения БЭМ-дерева можно использовать [генератор БЭМ-дерева от Йоксель](#), он подсвечивает грубые ошибки и позволяет увидеть весь результат работы в очень прозрачной форме. Нюансы и детали он не учитывает, например, когда на блоке висит и класс элемента этого же блока (`<section class="blog blog__title">`). Это придётся проверить самостоятельно.

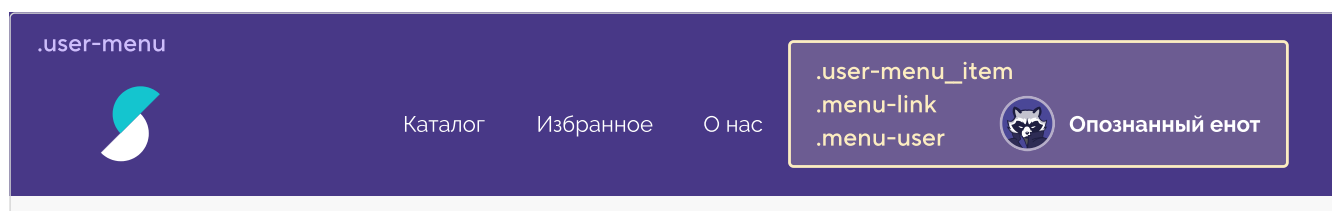
БЭМ-сущность (блок, элемент или модификатор)

Определяется названием класса. Лучше, если у каждой такой сущности будет своя зона ответственности: одна сущность отвечает за внешний вид, другая — за положение на странице. Разные разработчики предпочитают определять принадлежность к бизнес-сущности или элементу интерфейса. Не страшно, если один элемент страницы будет одновременно и блоком и элементом и модификатором.

```

<a class="user-menu__item menu-link menu-user" href="#">Опознанный енот</a>

```



Ссылка на профиль пользователя и одновременно пункт меню в пользовательском меню

Нотация

Соглашение по именованию. Их достаточно много, и очень важно придерживаться именно того соглашения, которое принято для этого проекта.

В некоторых проектах используют префиксы (к примеру, `b-`). Классическая яндексовская нотация БЭМ: `имя-блока__имя-элемента_модификатор`, при этом обычно имена — односоставные (`site-navigation__link_active`), а у многих блоков и элементов не будет модификаторов, но нотация показывает самый развёрнутый из возможных вариантов.

Мы будем использовать в обучающих материалах популярную нотацию, где в названии класса элементы отделяются от блока двойным «подчёркиванием» `__`, а модификаторы — двойным дефисом `--`.

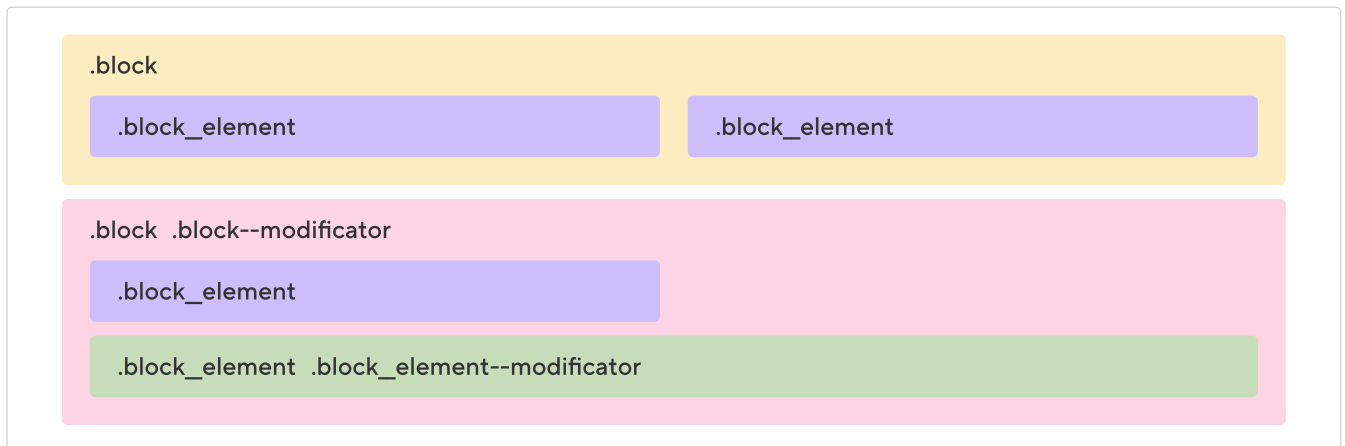


Схема наименования классов (нотация) в проекте

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

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

