

# Методология БЭМ

## Введение

Перед тем как мы приступим к изучению методологии БЭМ давайте разберёмся а какую задачу мы решаем. Ведь технологий и возможностей достаточно много, но какие-то бывают не очень популярные или очень ситуативные. Возможно вы уже что-то слышали про методологии, но самое главное что нам нужно понять, при создании классов для html элементов мы не используем никакую логику, просто называем элементы, как нам будет удобней, из-за этого возникают сложности, давайте с них и начнём.

## Минусы стандартного именования классов

Давайте рассмотрим простой пример, для этого создадим стандартную разметку, внутри которой будет находить блок header для которого заданы произвольные стили, внутри блока header будет находиться лого сайта, который отталкивается от верхней границы на 15px, данная часть для нас является одной из самых важных.

### HTML

```
<header class="header center">

  <a href="index.html" class="logo">Logo</a>

</header>
```

### CSS

```
* {

  margin: 0;

  padding: 0;

}

.center {

  padding-left: calc(50% - 570px);

  padding-right: calc(50% - 570px);

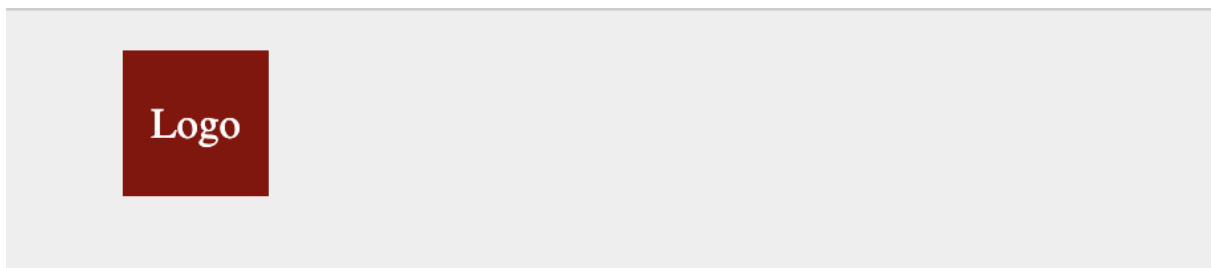
}

.header {

  height: 100px;

  background-color: #eee;
```

```
        display: flex;
    }
    .logo {
        background-color: darkred;
        width: 55px;
        height: 55px;
        text-decoration: none;
        display: flex;
        justify-content: center;
        align-items: center;
        color: white;
        margin-top: 15px;
    }
```



Казалось бы а в чем проблема? Давайте добавим какой-то блок content который будет для примера обладать произвольным цветом и высотой и блок footer внутри которого так же располагается лого.

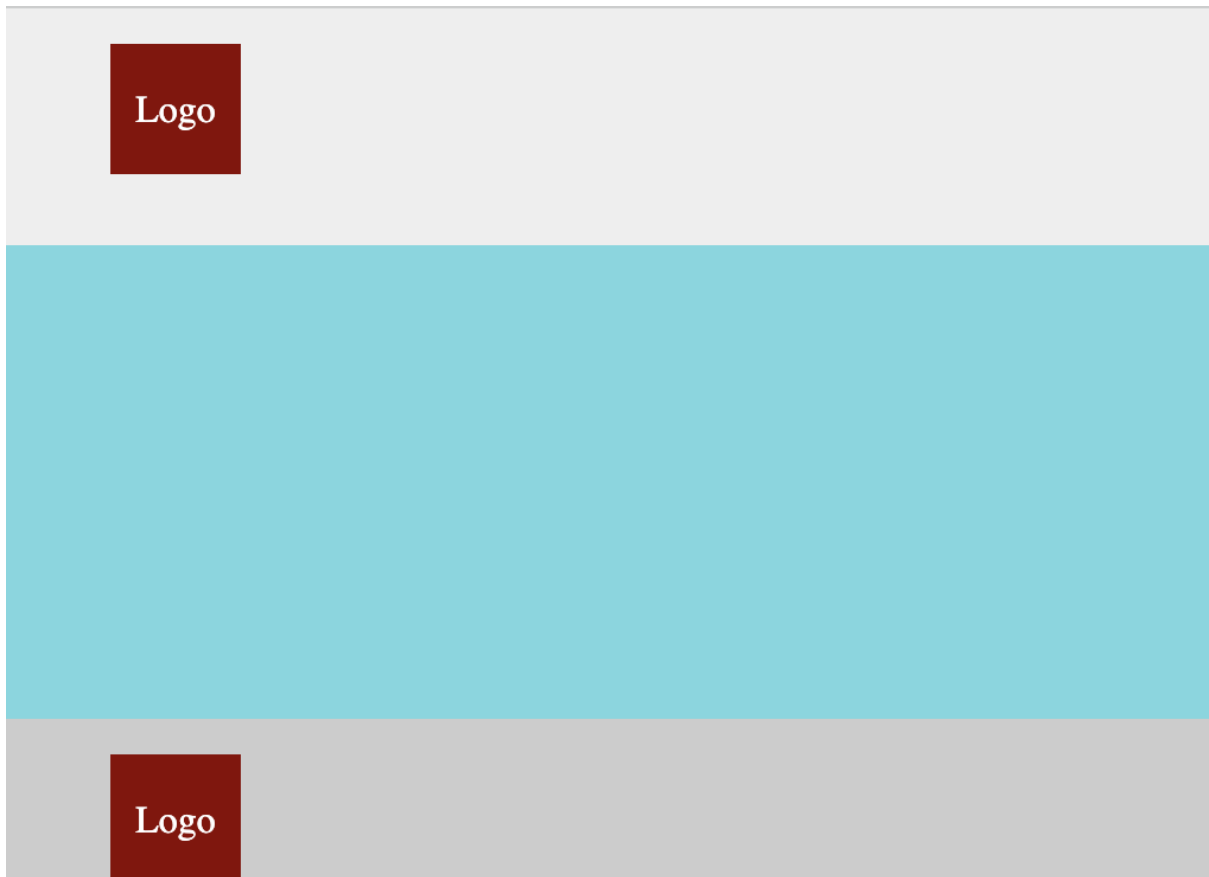
## HTML

```
<div class="content"></div>
<footer class="footer center">
    <a href="index.html" class="logo">Logo</a>
</footer>
```

## CSS

```
.content {
    background-color: rgb(116, 215, 224);
    height: 200px;
}
```

```
.footer {  
    background-color: #ccc;  
    display: flex;  
}
```



А вот и ответ на наш вопрос, мы добавили лого, но для него уже есть какое-то позиционирование, а теперь представим что footer будет меньше или нужно чтобы лого был справа, придётся переделывать позиционирование которое было задано в шапке сайта, в нашем примере это `margin-top: 15px;` но ведь такого позиционирования может быть огромное количество, плюс таких заданий будет так же очень много.

Именно эту задачу решает методология, нужно отметить что БЭМ это просто одна из самых популярных методологий, бывают и другие, так что если вы вдруг решили выбрать какую-то другую методологию, то после изучения БЭМ вы с легкостью сможете выбрать любую понравившуюся, тут главное понять принцип работы

## Плюсы использования методологии

Принцип работы с методологией БЭМ заключается в разделении интерфейса на независимые блоки. Это позволяет создать чёткую структуру и уменьшить вероятность появления ошибок и накладок. Рассмотрим плюсы такого подхода:

**Независимость блоков:** каждый блок может быть переиспользован, что упрощает разработку и поддержку кода.

```
<header class="header center">

<a href="index.html" class="header__logo">Logo</a>

</header>
```

CSS

```
.header__logo {

...

margin-top: 15px;

}
```

Теперь, когда мы добавляем логотип в footer, он может иметь совершенно другой класс:

```
<footer class="footer center">

<a href="index.html" class="footer__logo">Logo</a>

</footer>
```

В этом случае позиционирование `.header__logo` не влияет на `.footer__logo`.

## Четкая структура

Благодаря строгим правилам именования, которые предоставляет БЭМ, разметка сайта становится легко читаемой даже для новичков в команде.

Пример:

Представьте, что у вас есть карточка товара:

Без БЭМ:

```
<div class="card">



<h2 class="title">Название товара</h2>

<p class="description">Описание товара</p>

</div>
```

С использованием БЭМ

```
<div class="product-card">



<h2 class="product-card__title">Название товара</h2>

<p      class="product-card__description">Описание
товара</p>

</div>
```

Во втором примере сразу становится понятно, что все элементы принадлежат карточке товара.

## Меньше дублирования кода

С помощью БЭМ вы можете задавать общие стили для блоков и элементов, а затем модифицировать их при необходимости с помощью модификаторов.

Пример:

```
<button      class="button      button--primary">Главная
кнопка</button>

<button                                class="button
button--secondary">Второстепенная кнопка</button>
```

```
.button {
padding: 10px 20px;
border: none;
border-radius: 5px;
}

.button--primary {
background-color: blue;
color: white;
}

.button--secondary {
background-color: gray;
color: black;
```

```
}
```

В этом примере базовые стили кнопки определены в `.button`, а модификаторы позволяют быстро изменять внешний вид кнопки.

## Легкость масштабирования

Так как каждый блок является независимым, ввод новых элементов или блоков не влияет на существующую структуру.

Пример:

Если бы мы захотели добавить к нашей карточке товара кнопку покупки, это было бы так просто, как:

```
<div class="product-card">
...
<button
class="product-card__buy-button">Купить</button>
</div>
```

## Избегание стилевых конфликтов

Поскольку каждый блок и элемент имеют уникальные классы, это исключает возможность случайного пересечения стилей.

Пример:

Представьте, что у вас в проекте есть два разных блока, каждый из которых имеет элемент с классом `.title`. Если не использовать БЭМ, стили одного `.title` могут переопределить стили другого. Но в БЭМ каждый элемент будет иметь

свой уникальный класс, например `.product-card__title` и `.news-article__title`, что исключает подобные конфликты.

Итак, давайте подведем итог тому, что мы узнали о методологии БЭМ и ее преимуществах при разработке веб-интерфейсов.

БЭМ — это методология разработки, которая предлагает четкую структуру для именования классов в CSS и соответствующую HTML-разметку. Она была разработана в Яндексе и с тех пор стала популярной не только в российском, но и в мировом веб-сообществе.

**Четкая структура:** БЭМ привносит в проекты порядок. Даже в больших и сложных проектах, где участвуют десятки разработчиков, благодаря БЭМ, структура остается понятной и читаемой. Каждый элемент разметки имеет свое предназначение и место, что упрощает работу над проектом на любом этапе его жизненного цикла.

**Меньше дублирования кода:** Один из главных недугов больших CSS-файлов — это дублирование. БЭМ позволяет определить базовые стили для блоков и элементов, и затем расширять их или модифицировать с помощью модификаторов. Это позволяет избежать ненужного дублирования и делает код более управляемым.

**Легкость масштабирования:** В современных веб-проектах необходимость в масштабировании появляется рано или поздно. Благодаря независимости блоков в БЭМ, внесение изменений и добавление новых компонентов не вызывает проблем с существующим кодом.

**Избегание стилевых конфликтов:** Наверное, каждый разработчик сталкивался с проблемой, когда стили одного компонента влияли на другой, не имея с этим никакого отношения. БЭМ минимизирует такие риски, обеспечивая каждому блоку и элементу уникальные идентификаторы.

В завершение стоит сказать, что БЭМ — это не просто набор правил именования. Это целая концепция разработки, которая делает код более структурированным, устойчивым к изменениям и легким для понимания.



Используя БЭМ, команды разработчиков могут быстрее и эффективнее работать над проектами, избегая многих распространенных проблем и ошибок. Это инвестиция в качество кода, которая окупается на протяжении всего жизненного цикла проекта.