

---

---

# ВЕМ и ветвление в GIT

---

---

**БЭМ или ВЕМ**

# CSS

Easy to learn, hard to master (maintain)

Легко научиться тяжело овладеть (поддерживать)

БЭМ облегчает поддержку проектов

# Краткая справка

БЭМ создан сотрудниками Yandex

Компонентный подход к веб-разработке.

В основе лежит принцип разделения интерфейса на независимые блоки.

# БЭМ/ВЕМ

## БЭМ

Блок

Элемент

Модификатор

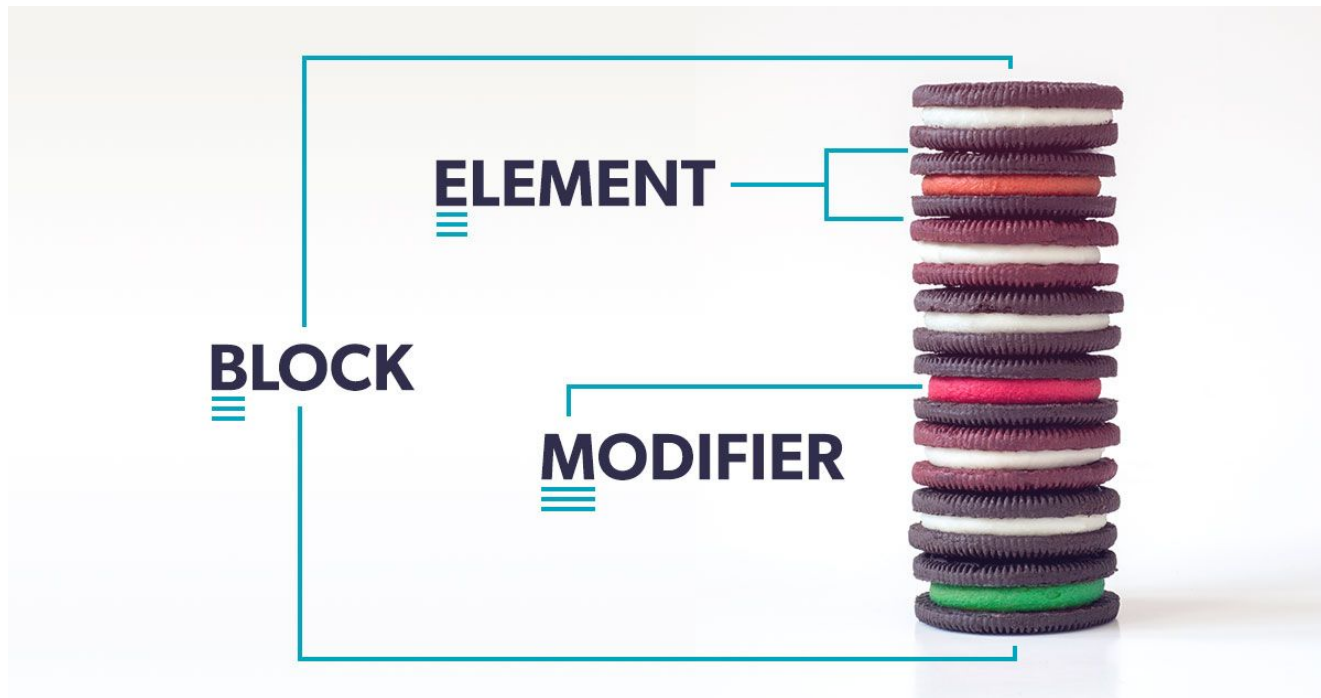
## ВЕМ

Block

Element

Modifier

# БЭМ/BEM



# Блок

Функционально независимый элемент страницы, который может быть использован несколько раз.

- Блок отвечает на вопрос “Что это?”.
- Имена для блоков максимально отражают содержимое.
- Не задаются отступы или позиционирование, чтобы он был независим и можно было легко переносить.
- Блоки можно вкладывать друг в друга, допустима любая вложенность.

# Блок



```
<div class="имя-блока"></div>
```



# Блок



```
<header class="header">  
  <div class="logo"></div>  
</header>  
<div class="widget"></div>
```

# Элемент

Неотъемлемая составная часть блока, которая не может быть использована в отрыве от него.

- Также отвечает на вопрос “Что это?”.
- Элементы можно вкладывать друг в друга, допустима любая вложенность.
- Элемент всегда является часть блока. Элемент не обязательный компонент блока.

# Элемент



```
<div class="имя-блока">  
  <div class="имя-блока__имя-элемента"></div>  
</div>
```

# Элемент



```
<div class="sidebar">  
  <ul class="sidebar__list">  
    <li class="sidebar__item"></li>  
  </ul>  
</div>
```

# Блок или Элемент

Если используется повторно или на других страницах и не зависит ни от каких других компонентов, то блок.

Если фрагмент не может или не будет использоваться самостоятельно без родительского блока.

# Модификатор

Используется для определения или уточнения внешнего вида, состояния или поведения блока или элемента, когда нужно выделить объект из множества таких же.

- Отвечает на вопрос “Какой?” “Как выглядит?”, “Как себя ведет?”.
- Типы модификаторов:
  - Булевый: disabled, если его нет, то считается что true.
  - Ключ-значение: Например, «меню с темой оформления islands»
- Модификатор не может использоваться в отрыве от модифицируемого блока или элемента.

# Модификатор булевый



```
<div class="имя-блока имя-блока_имя-модификатора">  
  <div class="имя-блока__имя-элемента имя-блока__имя-элемента_имя-модификатора">  
    </div>  
  </div>
```

# Модификатор булевый



```
<form class="form form_focused">  
  <button class="form__button form__button_disabled"></button>  
</form>
```



# Модификатор ключ-значение



```
<div class="имя-блока имя-блока_имя-модификатора_значение-модификатора">  
<div class="имя-блока__имя-элемента имя-блока__имя-элемента_имя-модификатора_значение-модификатора">  
</div>  
</div>
```

# Модификатор ключ-значение



```
<form class="form form_theme_islands">  
  <button class="form__button form__button_size_m"></button>  
</form>
```

# Микс

Миксы позволяют:

- совмещать поведение и стили нескольких сущностей без дублирования кода;
- создавать семантически новые компоненты интерфейса на основе имеющихся.

# Микс



```
<div class="parent">  
  <form class="form parent_contact-form">  
  </form>  
</div>
```

# Ссылки

Официальная документация - <https://ru.bem.info/>

Шпаргалка - <https://9elements.com/bem-cheat-sheet/>

Зачем нужен БЭМ? - <https://htmlacademy.ru/blog/boost/frontend/short-5>

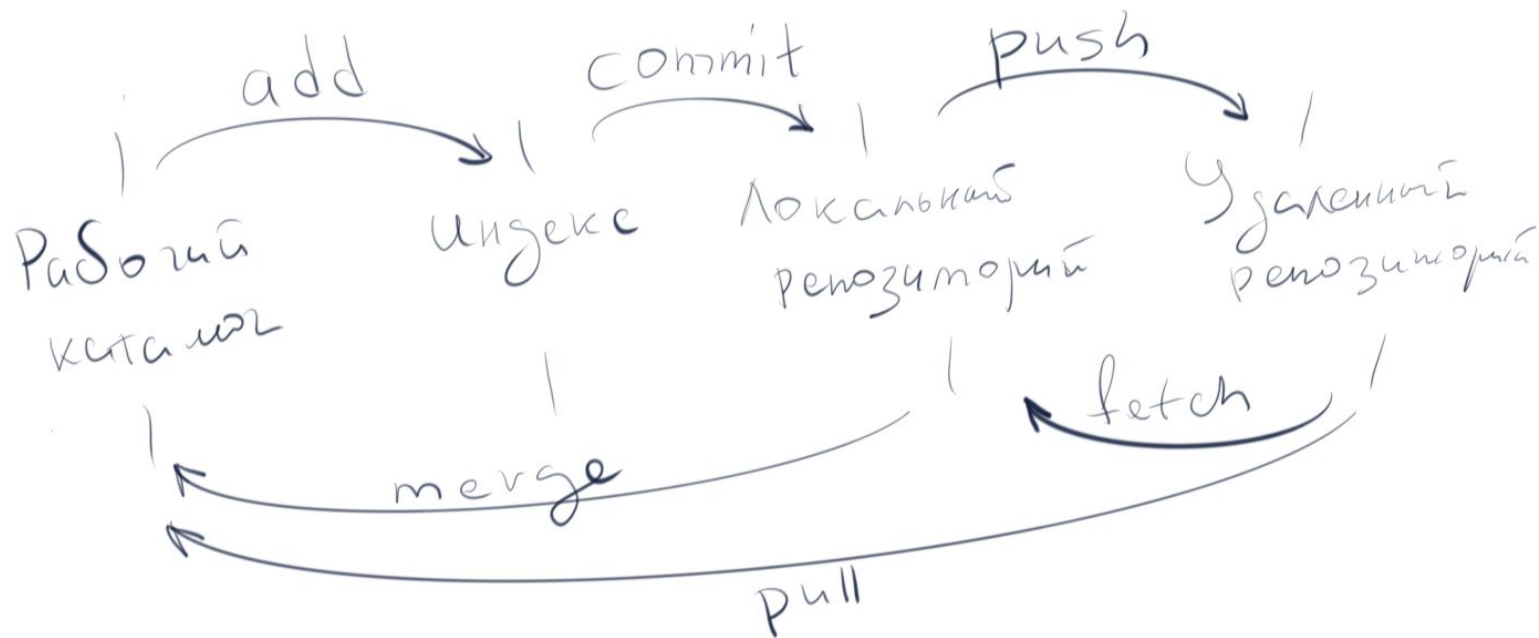
Примеры имен классов - <https://tpverstak.ru/common-css-class-names/>

# Методологии

- Object-Oriented CSS (OOCSS)
  - Характеризуется разделением контейнеров и содержимого с помощью «объектов» CSS.
- Scalable and Modular Architecture for CSS (SMACSS)
  - Руководство по стилю написания CSS с пятью категориями правил.
- SUIT CSS (SUITCSS)
  - Структурированные имена классов и значимые дефисы.
- Atomic CSS (Atomic)
  - Разбиение стилей на атомарные или неделимые части.

# Ветвление в GIT

# Работа GIT





# Список веток



# Создание новой ветки



```
git branch имя-новой-ветки
```

# Переключение на ветку



```
git checkout имя-ветки
```

# Создать новую ветку и сразу переключиться



```
git checkout -b имя-новой-ветки
```

# Слияние веток



```
git merge имя-ветки
```

# Слияние веток



```
git rebase имя-ветки
```

# Отмена слияния



```
git merge --abort
```

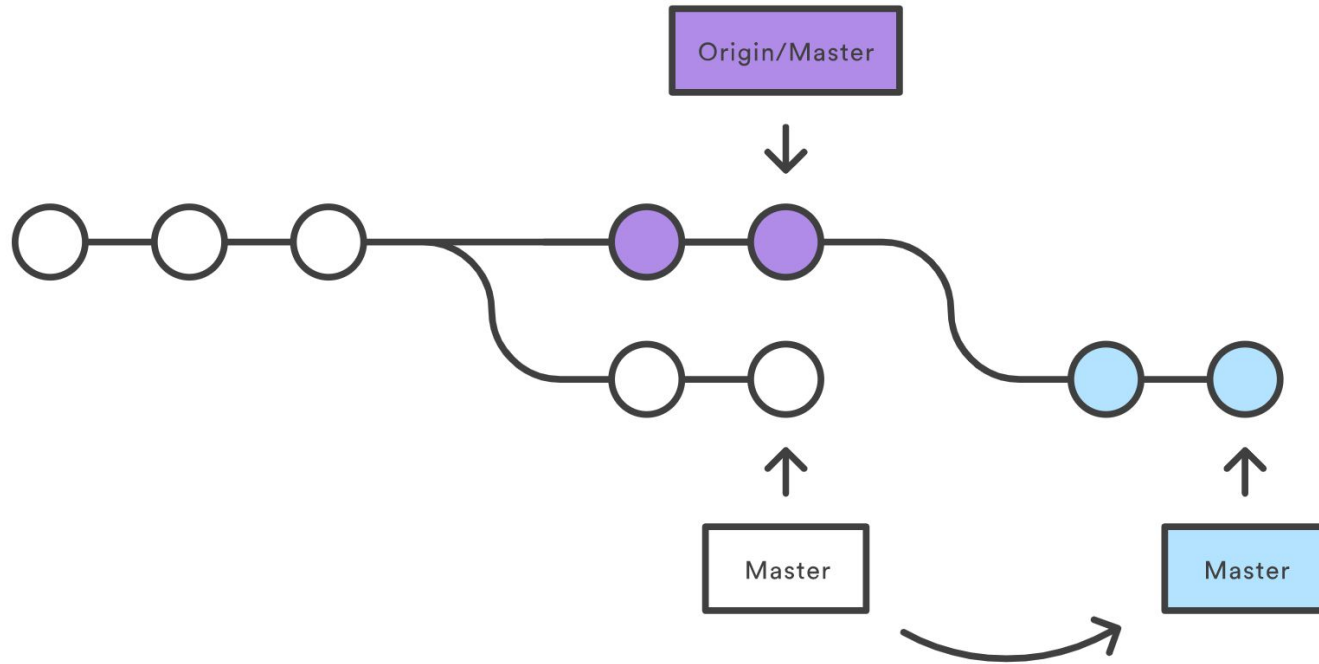
# Откат



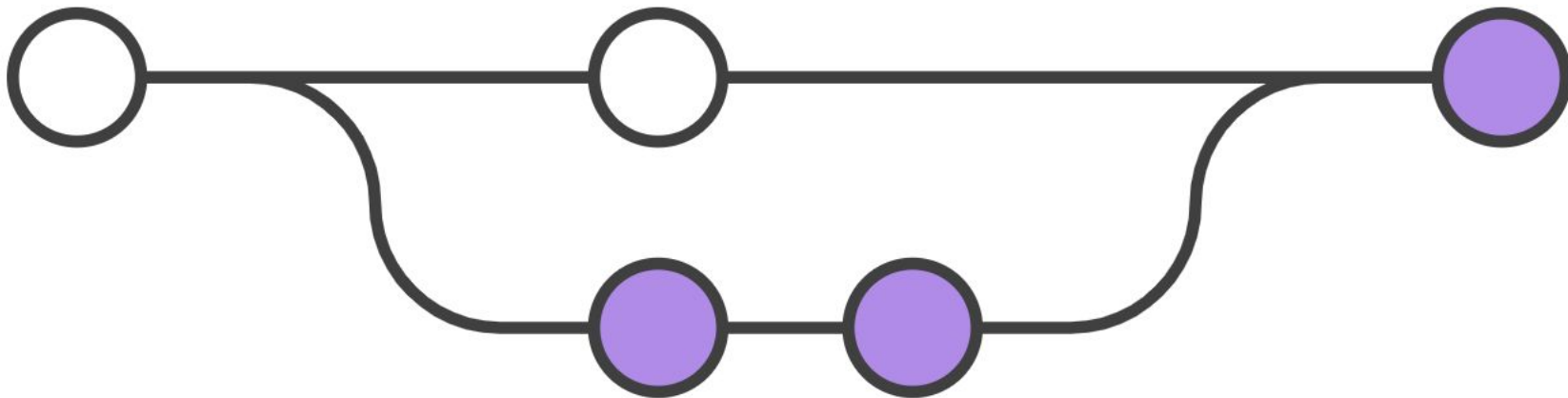
```
git reset --hard  
git reset --soft  
git reset --mixed
```



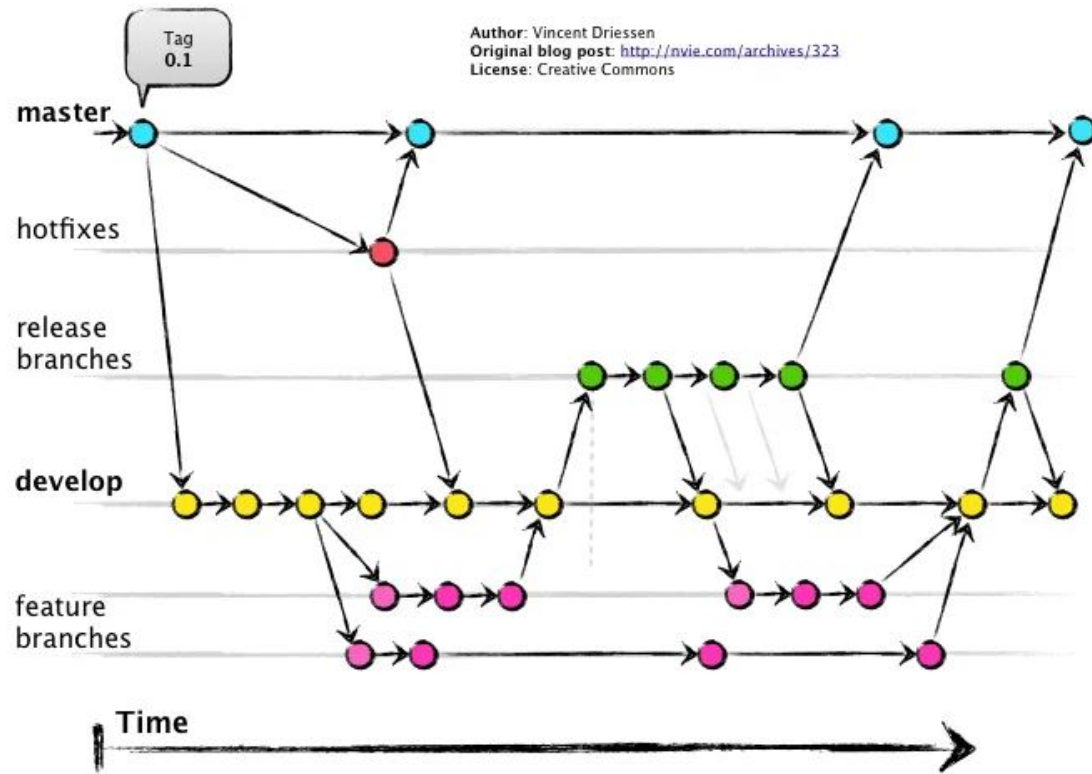
# Centralized Workflow



# Feature Branch Workflow



# GitFlow



# Ссылки

Работа с ветками - [https://learngitbranching.js.org/?locale=ru\\_RU](https://learngitbranching.js.org/?locale=ru_RU)

Лекция от Яндекса - <https://youtu.be/yDSs80lu3ak>

GitFlow - <https://nvie.com/posts/a-successful-git-branching-model/>

Про GitFlow на Хабре - <https://habr.com/ru/post/106912/>

GitHub Flow на Хабре - <https://habr.com/ru/post/346066/>

Набор готовых команд - <https://github.com/lmangazaliev/git-tips>

“Как исправить” - <https://habr.com/ru/company/flant/blog/419733/>