

# Как сделать табы?

Чтобы сделать табы на сайте, мы напишем собственный JavaScript-код, который будет показывать/скрывать нужные нам блоки по клику.

## HTML

Начнём с HTML-разметки. В ней есть важные моменты, которые стоит разобрать.

```
<!-- кнопки табов -->
<ul class="tabs-nav">
  <li class="tabs-nav__item"><button class="tabs-nav__btn
tabs-nab__btn--active" data-path="one">Первый таб</button></li>
  <li class="tabs-nav__item"><button class="tabs-nav__btn"
data-path="two">Второй таб</button></li>
  <li class="tabs-nav__item"><button class="tabs-nav__btn"
data-path="three">Третий таб</button></li>
</ul>
<div class="tabs-content">
  <!-- элементы табов -->
  <div class="tabs-item tabs-item--active" data-target="one">
    Далеко-далеко за словесными горами в странe1.
  </div>
  <div class="tabs-item" data-target="two">
    Далеко-далеко за словесными горами в стране2.
  </div>
  <div class="tabs-item" data-target="three">
    Далеко-далеко за словесными горами в стране3.
  </div>
</div>
```

Итак, важные моменты.

1. У кнопок имеются так называемые дата-атрибуты (data-path). Дата-атрибуты — это атрибуты, которые мы можем придумать самостоятельно. Вместо path можно написать всё что угодно, но по смыслу нам подходит именно это, т. к. path в переводе — «путь». Внутри атрибутов три разных значения — one, two, three, для простоты. Здесь также могут быть любые значения.
2. У блоков с классом tabs-item также есть дата-атрибут data-target, и значения такие же — one, two, three. Это сделано намеренно, чтобы мы могли связать

кнопки и элементы табов. Дата-атрибуты — частая практика для подобных вещей, когда нужно связать разные элементы между собой.

3. Также у первой из кнопок и у первого из элементов таба есть дополнительный класс с префиксом `--active`. Этот класс нужен для того, чтобы особым образом стилизовать эти элементы.

## CSS

Напишем простейший код, который сделает наши табы рабочими по умолчанию, даже без JS, — выделит активную кнопку другим цветом и скроет все элементы таба, кроме первого.

```
.tabs-nab__btn--active {  
  color: tan;  
}  
  
.tabs-item {  
  display: none;  
}  
  
.tabs-item--active {  
  display: block;  
}
```

Как видите, для активной кнопки задали другой цвет, скрыли все элементы таба и вернули к показу только один — активный.

## JS

Теперь напишем JS-код, который будет переключать табы. Тут всё довольно логично, но давайте разберём построчно, что же происходит.

```
document.querySelectorAll('.tabs-nav__btn').forEach(function(tabsBtn) {  
  tabsBtn.addEventListener('click', function(e) {  
    const path = e.currentTarget.dataset.path;  
  
    document.querySelectorAll('.tabs-nav__btn').forEach(function(btn) {  
btn.classList.remove('tabs-nav__btn--active')));  
    e.currentTarget.classList.add('tabs-nav__btn--active');
```

```
document.querySelectorAll('.tabs-item').forEach(function(tabsBtn) {
  tabsBtn.classList.remove('tabs-item--active')});

document.querySelector(`[data-target="${path}"]`).classList.add('tabs-item--active');
});
});
```

#### 1. document.querySelectorAll('.tabs-nav\_\_btn').forEach(function(tabsBtn){

С помощью метода `querySelectorAll` мы находим все элементы с классом `tabs-nav__btn` на странице — так сказать, «пробегаем» по ним циклом. Внутри `function` есть слово `tabsBtn` — это переменная, которую мы создали сами, дали ей название. Она нужна для удобства далее.

Поскольку мы используем цикл `forEach`, JavaScript будет смотреть на каждую кнопку в отдельности и на каждом шаге цикла помещать её в переменную `tabsBtn` — таким образом, мы сможем обратиться к каждой из кнопок, что нам и нужно.

#### 2. tabsBtn.addEventListener('click', function(e){

А здесь мы как раз и обращаемся к каждой кнопке, навешивая обработчик события `click`, то есть говоря, что по клику мы сейчас будем делать какое-то действие. Очень важна здесь буква `e` внутри `function`, т. к. она нам понадобится далее.

#### 3. const path = e.currentTarget.dataset.path;

Здесь мы создаём переменную `path` и помещаем в неё некое значение. Как раз здесь нам понадобится та самая буква `e`. Если по-простому — это идентификатор события, которое произошло. Произошёл клик, и в этом идентификаторе хранится информация о клике — куда кликнули, какие координаты и т. д. Запись `e.currentTarget` выделяет конкретный нажатый элемент — то есть нашу кнопку.

А запись `dataset.path` позволяет нам получить значения атрибута `data-path`, то есть, по сути, всё, что делает эта строка, — получает в переменную значение `data-атрибута` нажатого элемента. Думаю, вы уже понимаете логику — мы кликаем по кнопке, получаем её атрибут, и затем остаётся только найти элемент с точно таким же атрибутом.

#### 4. А теперь перенесёмся в последнюю строку, которая и отвечает за поиск элемента таба:

```
document.querySelector(`[data-target="${path}"]`).classList.add('tabs-item--active');
```

Здесь мы обращаемся к элементу с `data-target` и даём ему класс активности. Но обращаемся хитрым образом. Запись `[data-target="{path}"]` подставляет нашу переменную `path` в `data-target`, и именно за счёт этого находится нужный элемент. То есть мы кликнули на кнопку, нашли у неё значение атрибута `data-path`, подставили его при поиске `data-target` — и тем самым нашли нужный элемент таба.

Здесь есть важный момент: если ранее везде используются обычные одинарные кавычки, то именно в этой строке — обратные. Это нужно для удобства написания кода. Здесь используются так называемые [шаблонные строки](#).

- Итак, мы реализовали основное — по клику на кнопку находится верный элемент и ему добавляется класс. Но есть нюанс — не скрывается предыдущий элемент. Вернёмся чуть-чуть назад в коде и посмотрим, как это реализовано:

```
document.querySelectorAll('.tabs-item').forEach(function(tabsBtn){
  tabsBtn.classList.remove('tabs-item--active')});
```

Довольно простая строчка, в ней мы ищем все элементы таба, которые существуют, и у всех удаляем класс активности. То есть по клику на кнопку мы сперва удаляем у всех элементов класс, а потом уже ищем нужный и добавляем ему класс активности. Тем самым всегда будет активен только один элемент, что нам и нужно.

- И последнее действие, которое осталось, — менять класс активности и у кнопок тоже. Логика здесь на 100% такая же, как и в случае с элементами таба, — удалить классы у всех и добавить только нажатому.

```
document.querySelectorAll('.tabs-nav__btn').forEach(function(btn){
  btn.classList.remove('tabs-nav__btn--active')});
e.currentTarget.classList.add('tabs-nav__btn--active');
```

Собственно, здесь это и происходит. Сперва прошлись по всем и удалили класс, а затем снова использовали `e.currentTarget` для поиска текущего нажатого элемента и дали ему класс.

Теперь табы работают корректно.