

# Использование jQuery

Пример: <https://kodaktor.ru/jquery>

jQuery, как правило, включается в веб-страницу как один внешний JavaScript-файл:

```
<head>
  <script src="//ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
  </script>
</head>
```

Вся работа с jQuery ведётся с помощью функции `$`. Если на сайте применяются другие JavaScript библиотеки, где `$` может использоваться для своих нужд, то можно использовать её синоним — `jQuery`. Второй способ считается более правильным, а чтобы код не получался слишком громоздким, можно писать его следующим образом:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
  $(( )=>{
    // код, который выполняется после загрузки страницы
  });
</script>
```

Работу с jQuery можно разделить на 2 типа:

- Получение jQuery-объекта с помощью функции `$( )`. Например, передав в неё CSS-селектор, можно получить jQuery-объект всех элементов HTML, попадающих под критерий и далее работать с ними с помощью различных методов jQuery-объекта. В случае, если метод не должен возвращать какого-либо значения, он возвращает ссылку на jQuery объект, что позволяет вести цепочку вызовов методов согласно концепции [текущего интерфейса](#).
- Вызов глобальных методов у объекта `$`, например, удобных итераторов по массиву.

Типичный пример манипуляции сразу несколькими узлами DOM заключается в вызове `$` функции со строкой селектора CSS, что возвращает объект jQuery, содержащий некоторое количество элементов HTML-страницы. Эти элементы затем обрабатываются методами jQuery. Например,

```
$( 'div.test' ).add( 'p.quote' ).addClass( 'blue' ).slideDown( 'slow' );
```

находит все элементы div с классом test, а также все элементы p с классом quote, и затем добавляет им всем класс blue и визуально плавно спускает вниз. Здесь методы add, addClass и slideDown возвращают ссылку на исходный объект `$( 'div.test' )`, поэтому возможно вести такую цепочку.

Методы, начинающиеся с `$.`, удобно применять для обработки глобальных объектов. Например:

```
$.each([1,2,3], function() {
```

```
document.write(this + 1);  
});
```

добавит на страницу 234.

`$.ajax` и соответствующие функции позволяют использовать методы [AJAX](#). Например:

```
$.ajax({  
  type: 'POST',  
  url: 'some.php',  
  data: {name: 'John', location: 'Boston'},  
  success: function(msg){  
    alert( 'Data Saved: ' + msg );  
  }  
});
```

В этом примере идет обращение к скрипту `some.php` с параметрами `name=John&location=Boston`, и полученный результат выдается в сообщении посредством `alert()`.

Пример добавления к элементу обработчика события `click` с помощью jQuery:

```
$('#a').click(function() {  
  alert('Hello world!');  
});
```

В данном случае при нажатии на элемент `<a>` происходит вызов `alert('Hello world!')`.