

Рассмотрение функций библиотеки jQuery с аналогичными стандартными функциями из обычного языка программирования JavaScript

Кочитов Михаил Евгеньевич

*Приамурский государственный университет им. Шолом-Алейхема
студент*

Аннотация

В данной статье рассматриваются некоторые функции из JavaScript библиотеки jQuery, выполняющие идентичную работу с функциями, предоставляемыми из обычного языка программирования JavaScript. Также будет приведен пример их работы на веб-странице браузера.

Ключевые слова: библиотека, JavaScript, jQuery, функции

Exploring jQuery Library Functions with Similar Standard Functions from the Common JavaScript Programming Language

Kochitov Mikhail Evgenevich

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article discusses some functions from the jQuery JavaScript library that perform identical work with functions provided from the usual JavaScript programming language. An example of their operation on the browser web page will also be given.

Keywords: library, JavaScript, jQuery, functions

Многие веб-сайты наделены большим функционалом, который дает пользователям работать с ними легко и удобно без особых трудностей. Для такого функционала на клиентской части сайта используется язык программирования JavaScript, но большое количество сайтов снабжены JavaScript библиотекой jQuery, позволяющей компактно формировать код и функции, которые значительно упрощают разработку функционала на сайте. Однако библиотека jQuery использует язык программирования JavaScript и в ней имеются огромный набор методов и функций, которые позаимствованы из методов и функций обычного языка программирования JavaScript. Поэтому в данной статье будут рассмотрены функции, выполняющие идентичную работу с библиотекой jQuery и без нее.

В статье Тарана В.Н., Горщара Р.С. и Осыка В.Е. рассматривается сравнение реализации простой галереи с помощью HTML и jQuery [1]. Рассматривая статью Филоненко М.В. и Тюнина Е.Б. можно увидеть

библиотеку jQuery как инструмент быстрой разработки JavaScript [2]. Бурмистрова Н.В. и Смольянинов В.А. в своей статье рассмотрели некоторые возможности библиотеки jQuery в создании лендинг-сайта [3]. В статье Черномордова С.В. рассматриваются различия JavaScript и jQuery [4]. Рассматривая статью Притчиной М.И. и Зотовой Е.А. можно заметить разработку приложения с использованием технологий Phonegap, JavaScript, HTML и jQuery Mobile [5].

Целью данной статьи является рассмотрение функции библиотеки jQuery с аналогичными стандартными функциями из обычного языка программирования JavaScript. Также будет создан собственный пример, где будет показана их работа на веб-странице браузера.

Далее создадим веб-страницу и напишем в ней начальный HTML код структуры самой страницы, состоящей из шапки, в которой загрузим jQuery библиотеку, пустого пока содержимого страницы и пустого пока блока JavaScript кода

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Некоторые функции из jQuery</title>
    <script src="https://code.jquery.com/jquery-3.4.1.min.js" type="text/javascript"></script>
  </head>
  <body>
    <script>
  </script>
</body>
</html>
```

Рисунок 1 – Начальный HTML код с загрузкой библиотеки jQuery

На рисунке 1 изображен HTML код структуры веб-страницы с загруженной JavaScript библиотекой jQuery и пустыми блоками содержимого страницы и JavaScript кода. Теперь создадим в теле страницы три элемента: блок, поле ввода и кнопку, после присвоим им уникальные идентификаторы

```
<body>
  <div id="block">
  </div>
  <br><br>
  <input type="text" id="text" value="">
  <br><br>
  <input type="button" id="button" value="">
  <script>
</script>
</body>
```

Рисунок 2 – HTML код подготовительных трех элементов страницы

На рисунке 2 показан HTML код, в котором добавлен элемент блока (div), поле ввода (input с типом text) и кнопка (input с типом button). Все эти элементы являются пустыми, так как они не содержат каких либо обозначений в виде слов. К трем элементам присвоены идентификаторы:

block, text и button. Далее необходимо написать JavaScript код, в котором, для начала попробуем в элемент блока добавить случайный текст

```
<script>
  //jQuery функция
  $('#block').text("Привет Мир!");
  //Обычная функция
  document.getElementById('block').innerText = "Привет Мир!";
</script>
```

Рисунок 3 – Идентичные функции, добавляющие к блоку текст «Привет Мир!»

На рисунке 3 изображен JavaScript код, в котором написаны две функции, выполняющие идентичную работу – добавление текста «Привет Мир!» к элементу блока с идентификатором «block». Результат работы функций одной из них приводит к результату, показанному на рисунке ниже

Привет Мир!



Рисунок 4 – Элемент блока с добавленным текстом «Привет Мир!»

Далее рассмотрим следующие две идентичные функции, которые позволяют добавлять HTML теги в элемент блока. Для примера возьмем текст «Привет Мир!» с жирным шрифтом. JavaScript код этих двух функций показан на рисунке ниже

```
<script>
  //jQuery функция
  $('#block').html("<strong>Привет Мир!</strong>");
  //Обычная функция
  document.getElementById('block').innerHTML = "<strong>Привет Мир!</strong>";
</script>
```

Рисунок 5 – Идентичные функции, добавляющие жирный текст «Привет Мир!»

Как можно заметить на рисунке 5 для оформления текста в жирный шрифт используется тег , который должен быть по бокам текста.

Теперь глянем результат работы одной из этих двух функций на веб-странице

Привет Мир!



Рисунок 6 – Элемент блока с добавленным жирным текстом «Привет Мир!»

Глядя на рисунок 6, можно увидеть результат работы одной из двух функций. Они выполнили работу и в блоке текст «Привет Мир!» стал жирным. Теперь перейдем к следующим функциям, которые позволяют добавлять новый текст к элементу блока, не заменяя старый. Данный JavaScript код с этими функциями представлен на рисунке ниже

```
<script>
  //jQuery функция
  $('#block').html("<strong>Привет Мир!</strong><br>");
  $('#block').append("<i>Hello World!</i>");
  //Обычная функция
  document.getElementById('block').innerHTML = "<strong>Привет Мир!</strong><br>";
  document.getElementById('block').innerHTML += "<i>Hello World!</i>";
</script>
```

Рисунок 7 – Идентичные две функции, добавляющие к тексту «Привет Мир!» текст «Hello World!»

На рисунке 7 изображены две функции, первая присваивает к элементу блока жирный текст «Привет Мир!», а вторая добавляет (приписывает) без замены новый курсивный текст «Hello World!». Тег <i> используется для оформления текста в курсивный шрифт, а тег - в жирный шрифт. Теперь глянем результат работы этих функций на веб-странице

Привет Мир!
Hello World!



Рисунок 8 – Элемент блока с добавленными текстами «Привет Мир!» и «Hello World!»

На рисунке 8 представлен результат работы двух функций, они сделали свою работу и в элементе блока в две строки отобразились жирный «Привет Мир!» и курсивный «Hello World!» текст. Далее перейдем к элементам поля ввода и кнопки и попробуем присвоить им тоже значение «Привет Мир!» и «Hello World!». Для этого применим следующие функции, изображенные на рисунке ниже

```
<script>
  //jQuery функция
  $('#text').val("Привет Мир!");
  $('#button').val("Hello World!");
  //Обычная функция
  document.getElementById('text').value = "Привет Мир!";
  document.getElementById('button').value = "Hello World!";
</script>
```

Рисунок 9 – Идентичные функции, присваивающие элементу поля ввода и кнопке определенные значения

Как видно на рисунке 9 представлены функции присваивающие значение «Привет Мир!» к полю ввода и значение «Hello World!» к самой кнопке. Далее глянем результат работы этих функций на веб-странице

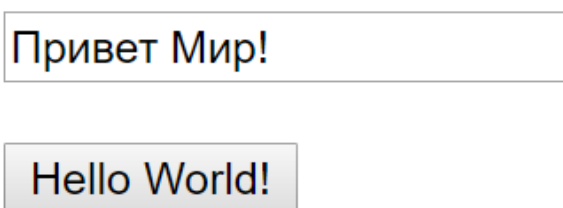


Рисунок 10 – Элементы поля ввода и кнопки с присвоенными значениями

На рисунке 10 показаны элементы поля ввода и кнопки с присвоенными значениями «Привет Мир!» и «Hello World!» соответственно. Теперь осталось рассмотреть последние идентичные функции, позволяющие задавать стили к элементам страницы. Для примера создадим в элементе блока текст со стилями зеленого, большого, жирного, курсивного и подчеркнутого шрифта. Для начала напишем эти функции, задающие нужные стили к элементу блока

```
<script>
//jQuery функция
$('#block').html("Привет Мир!");
$('#block').css({'color':'green'});
$('#block').css({'font-size':'36px'});
$('#block').css({'font-weight':'bold'});
$('#block').css({'font-style':'italic'});
$('#block').css({'text-decoration':'underline'});
//Обычная функция
document.getElementById('block').innerHTML = "Привет Мир!";
document.getElementById('block').style.color = "green";
document.getElementById('block').style.fontSize = "36px";
document.getElementById('block').style.fontWeight = "bold";
document.getElementById('block').style.fontStyle = "italic";
document.getElementById('block').style.textDecoration = "underline";
</script>
```

Рисунок 11 – Идентичный набор функций, задающий определенные стили к тексту «Привет Мир!» в элементе блока

На рисунке 11 продемонстрирован JavaScript код, который дает возможность функциям из библиотеки jQuery и обычными функциями выполнять идентичную работу и задавать такие же определенные стили текста «Привет Мир!» к элементу блока. Далее осталось увидеть результат измененного текста на веб-странице

Привет Мир!

Рисунок 12 – Элемент блока со стилизованным текстом «Привет Мир!»

На рисунке 12 показан последний результат измененного текста «Привет Мир!» с заданными стилями: зеленый цвет, размер шрифта 36 пикселей, жирный, курсивный и подчеркнутый шрифт.

Таким образом, были рассмотрены функции библиотеки jQuery с аналогичными стандартными функциями из обычного языка программирования JavaScript. Также были приведены примеры этих функций, которые отличаются по синтаксису, но одинаковы по выполняющей работе. В заключение можно сказать, что веб-сайты можно разрабатывать на языке программирования JavaScript без использования библиотеки jQuery, но многие разработчики прибегают к использованию данной библиотеки, так как она значительно упрощает написание JavaScript кода на клиентской части, тем самым ускоряя процесс разработки веб-сайтов.

Библиографический список

1. Таран В.Н., Горщар Р.С., Осыка В.Е. Сравнение реализации простой галереи с помощью HTML и jQuery // В сборнике: Наука, образование и государство в XXI веке сборник статей Международной научно-практической конференции. Под общей редакцией Г.Ю. Гуляева. 2016. С. 33-40.
2. Филоненко М.В., Тюнин Е.Б. Библиотека jQuery как инструмент быстрой разработки JavaScript // В сборнике: Информационное общество: современное состояние и перспективы развития Сборник материалов XI международного студенческого форума. 2018. С. 354-355.
3. Бурмистрова Н.В., Смольянинов В.А. Некоторые возможности библиотеки jQuery в создании Лендинг-сайта // В сборнике: Научные труды Калужского государственного университета имени К.Э. Циолковского Сер. "Естественные науки" Калужский государственный университет им. К.Э. Циолковского. 2018. С. 309-315.
4. Черномордов С.В. Различия JavaScript и jQuery // В сборнике: Технологии XXI века: проблемы и перспективы развития сборник статей Международной научно-практической конференции : в 2 ч.. 2017. С. 142-144.
5. Притчина М.И., Зотова Е.А. Разработка приложения с использованием технологий Phonegap, JavaScript, HTML и jQuery Mobile // Вестник МГУП имени Ивана Федорова. 2013. № 2. С. 50-54.
6. jQuery – Write less, do more // URL: <https://jquery.com/> (дата обращения 15.01.2020)
7. Современный учебник JavaScript // URL: <https://learn.javascript.ru/> (дата обращения 15.01.2020)
8. JavaScript и jQuery – одинаковые функции // URL: <https://ploshadka.net/javascript-jquery/> (дата обращения 15.01.2020)