

## Сортировка элементов списка веб-страницы с помощью javascript

*Радионов Сергей Владимирович*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной работе реализована сортировка элементов списка товаров веб-страницы интернет-магазина с помощью JavaScript. Также наглядно показан процесс и способ реализации сортировки, который можно применить для любого списка элементов веб-страницы.

**Ключевые слова:** сайт, сортировка, JavaScript, интернет-магазин, html.

## Sorting the elements of the list of web pages using javascript

*Radionov Sergey Vladimirovich*

*Sholom Aleichem Pryamursky State University*

*student*

### **Abstract**

This article implemented the sorting of items on the list of products on the web page of an online store using JavaScript. Also clearly shows the process and method of implementing sorting, which can be applied to any list of elements of a web page.

**Keywords:** website, sorting, JavaScript, online store, html.

Когда-то давным давно, когда JavaScript использовался только для анимации и слайдеров, все манипуляции с данными производили на стороне сервера. Например, для того, чтобы отсортировать товары человек нажимал соответствующую кнопку сортировки по необходимому критерию. Далее страница перезагружалась и мы видели товары, попадающие в рамки фильтра. Принцип работы сортировки в этом случае заключался в следующем: При нажатии на кнопку на сервер уходил POST или GET запрос, содержащий критерий сортировки. Серверный скрипт, обрабатывающий этот запрос, анализировал его и отправлял запрос в базу данных. Получив ответ от базы данных, скрипт сортировал полученные элементы и отправлял пользователя на специальную страницу или перезагружал существующую, на которой и выводился результат сортировки. У такого подхода есть большой минус - для каждой фильтрации или сортировки нужно было перезагружать страницу. А при медленном интернете каждая новая перезагрузка вызывает у пользователя только раздражение. С использованием javascript сортировка осуществляется мгновенно, что нравится пользователям сайтов.

Р. Жуйков, Е. Шарыгин в своей статье предлагают общий подход к предварительной компиляции программ на динамических языках, а также применение этого подхода для улучшения двух виртуальных машин - JavaScriptCore и V8. При реализации улучшенной виртуальной машины JavaScriptCore с использованием предварительной компиляции была учтена специфика использования JavaScript-программ в составе локально хранящихся приложений для платформы ARM. Для виртуальной машины V8 для платформы x86-64 в рамках исследования предварительная компиляция была реализована с помощью кэширования в отдельный файл одного из оптимизированных внутренних представлений [1]. В статье Е.В. Шукалова даны основные рекомендации по проектированию каталога b-2-b-компании. Их выполнение позволит клиенту и веб-студии говорить на одном языке и экономить время [2]. Д.В. Марцулевич разработал шкалу для оценки удобства использования (юзабилити) интернет-магазинов [3]. Рассмотрев статью О.Б. Кремер, Р.И. Быков можно увидеть, что в ней представлены этапы разработки динамического сайта, включающие кроссплатформенную технологию PHP и систему управления базами данных MySQL. Сайт предназначен для использования в учебном процессе по циклу дисциплин, в которых рассматриваются различные алгоритмы для решения конкретной задачи [4]. В работе В.Г. Варданян описываются разработанные методы оптимизации динамических многоуровневых компиляторов с учетом информации о профиле выполнения программы. Метод был реализован в динамическом компиляторе языка JavaScript V8, разработанном компанией Google. Использование профиля выполнения программы позволяет оптимизировать программу для конкретных входных данных. Это особенно актуально в связи с использованием JavaScript в операционных системах. Сценарий использования оптимизации на основе профиля программы в операционных системах следующий: на этапе тестирования программного обеспечения можно организовать сбор информации о профиле программы и использовать его для оптимизации приложений под конкретные случаи выполнения. Одним из новых применений использования информации о профиле программы может быть обеспечение немедленного переключения выполнения часто исполняющихся участков кода на уровень оптимизирующего компилятора. Другое применение - удаление обратных переходов на неоптимизирующие уровни выполнения [5]. Статья А.В. Кувайцева посвящена проблеме выбора среды разработки для веб-приложений на языке JavaScript. Проведенное исследование позволяет уменьшить затраты времени на выбор подходящей среды программирования, так как в статье рассмотрены современные решения от самых простых, вроде текстового редактора с подсветкой синтаксиса, до полноценных IDE [6]. Статья Е.В. Сомовой и Е.В. Киселёва была написана для определения роли специалистов сферы веб-разработок в современном рынке труда; в ней определяется важность использования скриптового языка программирования JavaScript при создании веб-сайта, описываются самые популярные JavaScript платформы [7]. Не менее значимы иностранные исследования [8-9].

Для того чтобы отсортировать элементы с помощью JavaScript, необходимо чтобы каждый элемент списка имел значение параметра, по которому будет осуществляться сортировка, возьмем параметры «price», который будет отвечать за значение цены и параметр «name», в котором будет содержаться название элемента.

Для удобного нахождения всех элементов списка в JavaScript, нужно чтобы все элементы списка включали один класс, либо имели одинаковый id, который будет использоваться только для элементов списка, в котором будет осуществляться сортировка. Будем использовать поиск по классу, поэтому добавим всем элементам списка класс, например, «list-item». Необходимо поместить все элементы списка в блок с уникальным id, например «list».

На рисунке 1 показано как будет выглядеть участок кода html, отвечающий за формирование списка.

```
<div id="list">
  <a href="url/" class="list-item" data-price="200" data-name="Рога">Рога</a>
  <a href="url/" class="list-item" data-price="400" data-name="Копыта">Копыта</a>
  <a href="url/" class="list-item" data-price="250" data-name="Рога длинные">Рога длинные</a>
</div>
```

Рисунок 1. Список элементов html

Внутри тела ссылки обязательно должен находиться текст, там может быть размещена сложная структура элементов. Далее необходимо добавить кнопки, при нажатии на которые будет осуществляться сортировка. В каждой кнопке необходимо добавить соответствующий id сортировки. В нашем случае для первой кнопки это будет «price», для второй «name». Необходимо еще добавить общий класс для кнопок, пусть он будет называться «sort-btn». Обрабатывать нажатие будем с помощью js.

```
<button id="price" class="sort-btn">цена</button>
<button id="name" class="sort-btn">название</button>
```

Рисунок 2. Кнопки сортировки html

Перейдем к программированию сортировки на JavaScript. Создадим файл sort.js. Добавим в него функцию, которая будет срабатывать при завершении загрузки страницы. Будем программировать с помощью общедоступного плагина jQuery [10], который значительно сократит количество программного кода.

```
$(document).ready(function () {  
    var state = 'up';  
    var sort = 'price';  
    $('.sort-btn').click(function () {  
        if($(this).attr('id') === sort){  
            if(state === 'up') {  
                state = 'down';  
            } else {  
                state = 'up';  
            }  
        } else {  
            sort = $(this).attr('id');  
            state = 'up';  
        }  
        Sort(sort, state);  
    });  
});
```

Рисунок 3. Функция при загрузке страницы html

Переменная state отвечает за порядок сортировки, по возрастанию – up, по убыванию – down. По умолчанию установим возрастание. Переменная sort отвечает за параметр по которому осуществляется сортировка. По умолчанию сортируем по цене. Далее назначаем на кнопки сортировки, которые находим по классу sort-btn, функцию. Если нажата кнопка, с тем же id, что и переменная sort, то меняем порядок сортировки, иначе меняем значение sort и устанавливаем порядок сортировки по возрастанию. Далее вызываем функцию сортировки (рис. 4).

```
function Sort(sort, state) {  
    if (sort === 'price') {  
        sortPrice(state);  
    } else if (sort === 'name') {  
        sortName(state);  
    }  
}
```

Рисунок 4. Функция сортировки

На вход функции подается параметр сортировки и порядок. В зависимости от входных параметра вызывается соответствующая функция. Для параметра price функции показаны на рисунке 5. Данной функцией можно сортировать любой числовой параметр. В функции получаем список элементов в переменную products, далее сортируем с помощью метода sort плагина jQuery, указываем по какому параметру нужно сортировать и в зависимости от переменной state выбираем порядок сортировки. Далее в html в блоке с id «list» меняем исходный список на отсортированный.

```
function sortPrice(state) {  
    var products = $('<div>.list-item</div>');  
    products.sort(function (a,b) {  
        if (state === "up")  
            return $(a).data("price")-$(b).data("price");  
        else  
            return $(b).data("price")-$(a).data("price");  
    });  
    $('#div#list').html(products);  
}
```

Рисунок 5. Функция сортировки по цене

```
function sortName(state) {  
    var products = $('<div>.list-item</div>');  
    products.sort(function (a,b) {  
        var name1 = $(a).data("name").toUpperCase();  
        var name2 = $(b).data("name").toUpperCase();  
        if (state === "up")  
            return name1.localeCompare(name2);  
        else  
            return name2.localeCompare(name1);  
    });  
    $('#div#list').html(products);  
}
```

Рисунок 6. Функция сортировки по названию

Функция сортировки по названию (рис.6) выглядит схожим образом, регистр букв в сортировке не учитывается.

Далее необходимо добавить плагин jQuery и скрипт sort.js в блок body страницы html (рис. 7)

```
<script type="text/javascript" src="https://code.jquery.com/jquery-3.2.1.min.js"></script>  
<script type="text/javascript" src="sort.js"></script>
```

Рисунок 7. Добавление скрипта в html

Таким образом в статье был продемонстрирован способ сортировки с помощью javascript. На рисунке 8 показан результат сортировки цены по возрастанию, а на рисунке 9 отображена сортировка по названию.

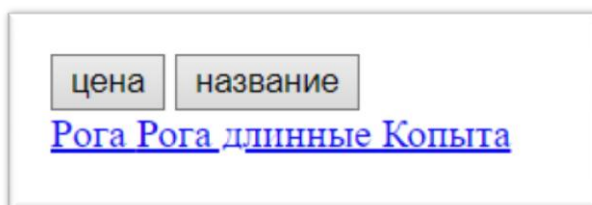


Рисунок 8. Html - сортировка элементов по возрастанию цены

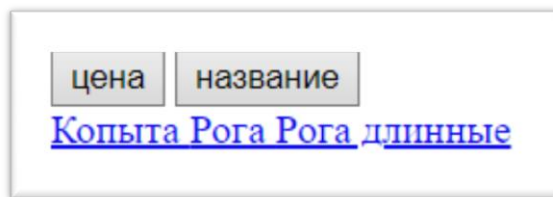


Рисунок 9. Html – сортировка элементов по возрастанию названия

### Библиографический список

1. Жуйков Р., Шарыгин Е. Методы предварительной оптимизации программ на языке javascript // Труды Института системного программирования РАН. 2015. Т. 27. № 6. С. 67-86.
2. Шукалова Е.В. Как построить эффективный каталог на сайте b-2-b-компании // Индустриальный и b2b маркетинг. 2008. № 3. С. 236-243.
3. Марцулевич Д.В. Разработка шкалы для оценки удобства использования (юзабилити) интернет-магазинов // Конкурентоспособность в глобальном мире: экономика, наука, технологии. 2017. № 7-2 (54). С. 97-104.
4. Кремер О.Б., Быков Р.И. Разработка динамического сайта для использования в учебном процессе по циклу дисциплин // В сборнике: Инновации в информационных технологиях, машиностроении и автотранспорте, сборник материалов Международной научно-практической конференции. Кузбасский государственный технический университет им Т.Ф. Горбачева. 2017. С. 25-28.
5. Варданын В.Г. Методы оптимизации программ на языке javascript, основанные на статистике выполнения программы // Труды Института системного программирования РАН. 2016. Т. 28. № 1. С. 5-20.
6. Кувайцев А.В. Анализ сред разработки веб-приложений на языке javascript // Современные научные исследования и инновации. 2016. № 9 (65). С. 121-124.
7. Сомова Е.В., Киселёв Е.В. Javascript как незаменимая составляющая функционирования любого сайта // В сборнике: Современные социально-экономические процессы: проблемы, закономерности, перспективы сборник статей III Международной научно-практической конференции: в 2 частях. 2017. С. 31-33.
8. Tilkov S., Vinoski S. Node. js: Using JavaScript to build high-performance network programs //IEEE Internet Computing. 2010. Т. 14. №. 6. С. 80-83.
9. Saxena P. et al. A symbolic execution framework for javascript //Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010. С. 513-528.
10. jQuery URL: <https://jquery.com/> (дата обращения: 11.01.2019).