

Массивы JavaScript

Неустроев А. В.

Неустроев Айаал Васильевич / Neustroev Aiaal Vasilevich - студент,
кафедра информационных технологий,
Институт математики и информатики
Северо-Восточный федеральный университет, г. Якутск

Аннотация: в статье предлагается описание интерпретируемого языка JavaScript. Создание и использование массивов в JavaScript. Измерение времени поиска в массиве.

Ключевые слова: JavaScript, массивы, HTML.

JavaScript – это интерпретируемый язык или язык скриптов. Используется для добавления интерактивности в HTML страницы. JavaScript используют еще для создания cookies, для определения браузера пользователя, чтобы загружать страницу специально для этого браузера [1].

Для того, чтобы начать писать на JavaScript языке, надо знать другие языки. Например, язык разметки HTML и каскадную таблицу стилей CSS.

Хотел бы остановиться на элементе JavaScript массив. Массив - это объект, который используют для хранения группы значений в одной переменной. Каждое пронумерованное значение называется элементом массива, а число, с которым связывается элемент, называется его индексом. В JavaScript массив имеет разные типы, можно в массиве создать массив. Создать массивы в JavaScript можно тремя способами [3].

1 способ: Обычный

```
var myFriends=new Array();  
myFriends [0]="John";  
myFriends [1]="Chris";  
myFriends [2]="Kerry";
```

2 способ: Сокращенный

```
var myFriends=new Array("John","Chris","Kerry");
```

3 способ: литеральный массив

```
var myFriends=["John","Chris","Kerry"];
```

Вывод элемента из массива document.write(myFriends[0]); → John.

Измерил время поиска в массиве без перебора в массиве. Выбрал три вида поиска данных в массиве [2].

1. Склейка массива в строку и поиск по нему

```
if (myarray.join().search("Строка поиска") != -1) { ... }
```

2. Этот пример похож на первый, но

```
if (("#" + myarray.join("#,#") + "#").search("#Строка поиска#") != -1) { ... }
```

3. Использование объекта вместо массива, «значениями массива» являются ключи хеша

```
if ("Строка поиска" in myarray) { ... }
```

Измерил время поиска. Для этого создал тестовую страницу, на которой создается 100000 строк, и ищу один из них. Все данные, которые получил, есть в таблице 1.

Таблица 1. Измерение времени поиска в массиве

Браузер	1 вид	2 вид	3 вид
Internet Explorer	341 мс.	217 мс.	0 мс.
Opera	243 мс.	154 мс.	0 мс.
Chrome	117 мс.	87 мс.	0 мс.
Firefox	183 мс.	119 мс.	0 мс.
Safari	124 мс.	96 мс.	0 мс.

Итогом проведенной работы стало, что 3 вид использование объекта вместо массива работает очень быстро, всегда находил сразу. Второй вид индексация массива находит быстро, чем первый вид. Самый медленный в моей работе стал первый вид. Склейка массива в строку ищет медленно, потому что я думаю из-за того, что присваивает массив в строку, только потом ищет, из-за этого медленный. Еще один аспект надо приложить к выбору браузера. Здесь результат еще зависит от браузера. На какой технологии сделан браузер, chromium или WebKit. JavaScript это на мой взгляд язык, который должен знать каждый уважающий программист. Если использовать язык хорошо, тогда он очень сильный в плане решения поставленных задач.

Литература

1. JavaScript. Подробное руководство / Под ред. Дэвид Флэнаган – СПб, Символ-Плюс, 2008. – 20-23 с.
2. JavaScript. Шаблоны / Под ред. Стоян Стефанов - СПб, Символ-Плюс 2011. – 120-123 с.
3. Выразительный Javascript / Под ред. Marijn Haverbeke - No Starch Press, 2015. – 37-41 с.