Три способа найти самое длинное слово в строке JavaScript

* [Сайтостроение](https://www.internet-technologies.ru/)
* [Статьи](https://www.internet-technologies.ru/articles/)
* [Веб-программирование](https://www.internet-technologies.ru/articles/webprogramming/)
* [Javascript](https://www.internet-technologies.ru/articles/webprogramming/javascript/)

3 года назад5K

С помощью разрабатываемого алгоритма мы узнаем в **JavaScript** длину строки и каждого слова. Затем мы сравним подсчеты, и определим, в каком слове содержится больше всего символов:

Это можно реализовать несколькими способами. Прежде всего, при помощи цикла **FOR**, методов **sort()** и **reduce()**.

**Содержание**

* [**Алгоритмическая задача**](https://www.internet-technologies.ru/articles/tri-sposoba-nayti-samoe-dlinnoe-slovo-v-stroke-javascript.html#header-1)
  + [Предложенные тест-кейсы](https://www.internet-technologies.ru/articles/tri-sposoba-nayti-samoe-dlinnoe-slovo-v-stroke-javascript.html#header-2)
    - [1. Находим самое длинное слово при помощи цикла FOR](https://www.internet-technologies.ru/articles/tri-sposoba-nayti-samoe-dlinnoe-slovo-v-stroke-javascript.html#header-3)
    - [2. Находим самое длинное слово с помощью метода sort()](https://www.internet-technologies.ru/articles/tri-sposoba-nayti-samoe-dlinnoe-slovo-v-stroke-javascript.html#header-4)
    - [3. Находим самое длинное слово с помощью метода reduce()](https://www.internet-technologies.ru/articles/tri-sposoba-nayti-samoe-dlinnoe-slovo-v-stroke-javascript.html#header-5)

**Алгоритмическая задача**

Вернуть длину самого длинного слова в предложении. Ответ должен быть выражен в числах:

function findLongestWord(str) {

return str.length;

}

findLongestWord("The quick brown fox jumped over the lazy dog");

**Предложенные тест-кейсы**

* **findLongestWord(“The quick brown fox jumped over the lazy dog”)** должна вернуть число;
* **findLongestWord(“The quick brown fox jumped over the lazy dog”)** должна вернуть число 6;
* **findLongestWord(“May the force be with you”)** должна вернуть число 5;
* **findLongestWord(“Google do a barrel roll”)** должна вернуть число 6;
* **findLongestWord(“What is the average airspeed velocity of an unladen swallow”)** должна вернуть число 8;
* **findLongestWord(“What if we try a super-long word such as otorhinolaryngology”)** должна вернуть число 19.

1. Находим самое длинное слово при помощи цикла FOR

Чтобы узнать длину строки **JavaScript**, мы воспользуемся методом **String.prototype.split()**.

Метод **split()** разбивает объект **String** на массив строк, разделяя строки на подстроки.

Между скобками в методе **split()** нужно будет добавить пробел,

var strSplit = “The quick brown fox jumped over the lazy dog”.split(‘ ‘);

что позволит получить массив из раздельных слов:

var strSplit = [“The”, “quick”, “brown”, “fox”, “jumped”, “over”, “the”, “lazy”, “dog”];

если не добавить пробел в скобках, то получится следующий результат:

var strSplit =

[“T”, “h”, “e”, “ “, “q”, “u”, “i”, “c”, “k”, “ “, “b”, “r”, “o”, “w”, “n”, “ “, “f”, “o”, “x”, “ “, “j”, “u”, “m”, “p”, “e”, “d”, “ “, “o”, “v”, “e”, “r”, “ “, “t”, “h”, “e”, “ “, “l”, “a”, “z”, “y”, “ “, “d”, “o”, “g”];

function findLongestWord(str) {

// Этап 1. Разбиваем строку на массив строк

var strSplit = str.split(' ');

// var strSplit = "The quick brown fox jumped over the lazy dog".split(' ');

// var strSplit = ["The", "quick", "brown", "fox", "jumped", "over", "the", "lazy", "dog"];

// Этап 2. Инициализируем переменную, которая будет содержать в себе размер самого длинного слова

var longestWord = 0;

// Этап 3. Создаем цикл FOR

for(var i = 0; i < strSplit.length; i++){

if(strSplit[i].length > longestWord){ // If strSplit[i].length больше, чем сравнимое слово...

longestWord = strSplit[i].length; // ...затем longestWord принимает новое значение

}

}

/\* Здесь strSplit.length = 9

Для каждой итерации: i = ? i < strSplit.length? i++ if(strSplit[i].length > longestWord)? longestWord = strSplit[i].length

1 итерация: 0 yes 1 if("The".length > 0)? => if(3 > 0)? longestWord = 3

2 итерация: 1 yes 2 if("quick".length > 3)? => if(5 > 3)? longestWord = 5

3 итерация: 2 yes 3 if("brown".length > 5)? => if(5 > 5)? longestWord = 5

4 итерация: 3 yes 4 if("fox".length > 5)? => if(3 > 5)? longestWord = 5

5 итерация: 4 yes 5 if("jumped".length > 5)? => if(6 > 5)? longestWord = 6

6 итерация: 5 yes 6 if("over".length > 6)? => if(4 > 6)? longestWord = 6

7 итерация: 6 yes 7 if("the".length > 6)? => if(3 > 6)? longestWord = 6

8 итерация: 7 yes 8 if("lazy".length > 6)? => if(4 > 6)? longestWord = 6

9 итерация: 8 yes 9 if("dog".length > 6)? => if(3 > 6)? longestWord = 6

10 итерация: 9 no

Конец цикла FOR \*/

//Этап 4. Возвращаем самое длинное слово

return longestWord; // 6

}

findLongestWord("The quick brown fox jumped over the lazy dog");

Без комментариев использование **JavaScript** длина строки **length** выглядит так:

function findLongestWord(str) {

var strSplit = str.split(' ');

var longestWord = 0;

for(var i = 0; i < strSplit.length; i++){

if(strSplit[i].length > longestWord){

longestWord = strSplit[i].length;

}

}

return longestWord;

}

findLongestWord("The quick brown fox jumped over the lazy dog");

2. Находим самое длинное слово с помощью метода sort()

Мы воспользуемся методом **Array.prototype.sort()**, который позволяет сортировать массив по определенному критерию, а затем возвращает длину первого элемента в данном массиве.

Метод **sort()** сортирует элементы массива и возвращает массив.

Если мы просто отсортируем массив:

var sortArray = [“The”, “quick”, “brown”, “fox”, “jumped”, “over”, “the”, “lazy”, “dog”].sort();

то получим следующий результат:

var sortArray = [“The”, “brown”, “dog”, “fox”, “jumped”, “lazy”, “over”, “quick”, “the”];

Перед тем, как узнать длину строки в JavaScript, мы сортируем элементы по определенному критерию,

[].sort(function(firstElement, secondElement) {

return secondElement.length — firstElement.length;

})

в котором длина второго элемента сравнивается с длиной первого:

function findLongestWord(str) {

// Этап 1. Разбиваем строку на массив строк

var strSplit = str.split(' ');

// var strSplit = "The quick brown fox jumped over the lazy dog".split(' ');

// var strSplit = ["The", "quick", "brown", "fox", "jumped", "over", "the", "lazy", "dog"];

// Этап 2. Сортируем элементы в массиве

var longestWord = strSplit.sort(function(a, b) {

return b.length - a.length;

});

/\* Процесс сортировки

a b b.length a.length var longestWord

"The" "quick" 5 3 ["quick", "The"]

"quick" "brown" 5 5 ["quick", "brown", "The"]

"brown" "fox" 3 5 ["quick", "brown", "The", "fox"]

"fox" "jumped" 6 3 ["jumped", quick", "brown", "The", "fox"]

"jumped" "over" 4 6 ["jumped", quick", "brown", "over", "The", "fox"]

"over" "the" 3 4 ["jumped", quick", "brown", "over", "The", "fox", "the"]

"the" "lazy" 4 3 ["jumped", quick", "brown", "over", "lazy", "The", "fox", "the"]

"lazy" "dog" 3 4 ["jumped", quick", "brown", "over", "lazy", "The", "fox", "the", "dog"]

\*/

// Этап 3. Возвращаем длину первого элемента в массиве

return longestWord[0].length; // var longestWord = ["jumped", "quick", "brown", "over", "lazy", "The", "fox", "the", "dog"];

// longestWord[0]="jumped" => jumped".length => 6

}

findLongestWord("The quick brown fox jumped over the lazy dog");

Код, с помощью которого мы узнаем длину строки **JavaScript**, без комментариев выглядит так:

function findLongestWord(str) {

var longestWord = str.split(' ').sort(function(a, b) { return b.length - a.length; });

return longestWord[0].length;

}

findLongestWord("The quick brown fox jumped over the lazy dog");

3. Находим самое длинное слово с помощью метода reduce()

**reduce()** применяет функцию к каждому значению в массиве (*слева направо*), чтобы сократить его до единственного значения. Метод исполняет **callback-функцию** лишь один раз для каждого элемента в массиве.

Вы можете представить исходное значение в качестве второго аргумента для метода **reduce**. Здесь мы добавляем пустую строку **“”**.

[].reduce(function(previousValue, currentValue) {...}, “”);

function findLongestWord(str) {

// Этап 1. Разбиваем строку на массив строк

var strSplit = str.split(' ');

// var strSplit = "The quick brown fox jumped over the lazy dog".split(' ');

// var strSplit = ["The", "quick", "brown", "fox", "jumped", "over", "the", "lazy", "dog"];

// Этап 2. Применяем метод reduce

var longestWord = strSplit.reduce(function(longest, currentWord) {

if(currentWord.length > longest.length)

return currentWord;

Else

return longest;

}, "");

/\* Процесс уменьшения

currentWord longest currentWord.length longest.length if(currentWord.length > longest.length)? var longestWord

"The" "" 3 0 yes "The"

"quick" "The" 5 3 yes "quick"

"brown" "quick" 5 5 no "quick"

"fox" "quick" 3 5 no "quick"

"jumped" "quick" 6 5 yes "jumped"

"over" "jumped" 4 6 no "jumped"

"the" "jumped" 3 6 no "jumped"

"lazy" "jumped" 4 6 no "jumped"

"dog" "jumped" 3 6 no "jumped"

\*/

// Этап 3. Возвращаем длину longestWord

return longestWord.length; // var longestWord = "jumped"

// longestWord.length => "jumped".length => 6

}

findLongestWord("The quick brown fox jumped over the lazy dog");

Без комментариев:

function findLongestWord(str) {

var longestWord = str.split(' ').reduce(function(longest, currentWord) {

return currentWord.length > longest.length ? currentWord : longest;

}, "");

return longestWord.length;

}

findLongestWord("The quick brown fox jumped over the lazy dog");