

Ace your JavaScript Interview. [Get my ebook](#). 100 solved Javascript, 20 solved React, & 2 frontend system design questions (**1160+ copies sold**). Get a [Free preview](#).



Advertisements

Reverse a string using recursion

Posted on [April 21, 2019](#) | by [Prashant Yadav](#)

Posted in [Algorithms](#), [String](#) | Tagged [Easy](#)

An algorithm to reverse a [string](#) using recursion.

We are going to implement two different algorithms to reverse a string using recursion in javascript. Everything will be written in [ES6](#).

Example

Input:
'prashant'

Output:
'tnahsarp'

Copy

A simple method

Implementation

- We will create function `reverseString` which will take string and the length of the string as a input.
- Then we will add a condition to check if the length of the string is 0 then return empty string `''`.
- Else we will add the last character of the string with second last character by calling the same function recursively.

```
function reverseString(str, n){  
  //If the length is 0  
  //then return an empty string  
  if(n == 0){  
    return '';  
  }  
  
  //Call the function recursively with one character less and so on.  
  return str[n-1] + reverseString(str, --n);  
}
```

Copy

Practically
prepare for
your
JavaScript
interview

[JavaScript
Revision](#)

[JavaScript-
Concept Based
Problems](#)

[Data Structures](#)

[Algorithms](#)

[Machine
Coding](#)

[Web
Fundamentals](#)

Advertisements



[Copy](#)**Input:**

```
console.log(reverseString('prashant', 8));
```

Output:

```
'tnahsarp'
```

Time complexity: $O(n)$.

Space complexity: $O(n)$.

Time and Space complexity

- We are call the same function recursively till the length of the string is greater than 0, so Time complexity is $O(n)$.
- As we are calling the function recursively it will stored in the call stack, so Space complexity is $O(n)$.

Using [String.substring\(\)](#) to reverse the string recursively.

`String.substring(start, end)` returns all the substring between the start and end index.

Implementation

- We are going to use the above approach only but instead of using string's length, we will pass the truncated substring itself.
- Then we will add the last character of the string with second last character and so on by calling the same function recursively.
- Keep calling the function till there are characters in the string. If there are no character then return empty string.

[Copy](#)

```
function reverseString(str){  
  //If the length is 0  
  //then return an empty string  
  if(str.length === 0){  
    return '';  
  }  
  
  //Call the function recursively with one character less and so on.  
  return str.substring(str.length, str.length-1) + reverseString(str.substring(0,  
str.length-1));  
}
```

[Copy](#)**Input:**

```
console.log(reverseString('prashant', 8));
```

Output:

```
'tnahsarp'
```

`str.substring(str.length, str.length-1)` will return the last character of the string like `'t'`.

Advertisements



`reverseString(str.substring(0, str.length-1))` will call the same function recursively without last character like `'prashan'`.

Keep repeating this till the string has characters in it.

```
't' + reverseString('prashan')
'tn' + reverseString('prasha')
'tna' + reverseString('prash')
'tnah' + reverseString('pras')
'tnahs' + reverseString('pra')
'tnahsa' + reverseString('pr')
'tnahsar' + reverseString('p')
'tnahsarp' + reverseString('')
'tnahsarp' + '';

return 'tnahsarp';
```

Copy

Time complexity: $O(n)$;

Space complexity: $O(n^2)$;

Time and Space complexity

- We are call the same function recursively for all the characters of the string, so Time complexity is $O(n)$.
- Recursive function will be stored in call stack and each `substring()` will return a copy of string with start & end index which will also take space, so Space complexity is $O(n^2)$.

[Prepare for your JavaScript Interview practically on each Interview rounds and grab that job.](#)

BEGIN LEARNING

Recommended Posts:

[Check if an array is palindrome in javascript](#)

[Learn how to reverse words in a string](#)

- [Check if binary tree has path sum](#)
- [knapsack problem in Javascript \(Bounded & Unbounded\)](#)
- [Print right view of a binary tree](#)
- [Check if two string are anagram of each other](#)
- [Find the intersection point of two linked list](#)
- [Convert Fahrenheit to Celsius in JavaScript](#)
- [Bubble sort algorithm in javascript](#)
- [Algorithm to merge two sorted array](#)

[Prev](#)

[Next](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Start typing...

Name*

Name

Email*

Email

POST COMMENT

Advertisements



[About Us](#)

[Contact Us](#)

[Privacy Policy](#)

[Advertise](#)



Handcrafted with ♥ somewhere in **Mumbai**

© 2023 [LearnersBucket](#) | [Prashant Yadav](#)

