Reverse a String

Reverse the provided string.

You may need to turn the string into an array before you can reverse it.

Your result must be a string.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Global String Object**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String)
* [**String.prototype.split()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split)
* [**Array.prototype.reverse()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reverse)
* [**Array.prototype.join()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/join)

Code:

function reverseString(str) {

var firstSplit = str.split('');

var secondReverse = firstSplit.reverse();

var thirdJoin = secondReverse.join('');

return thirdJoin;

}

reverseString("hello");

Factorialize a Number

Return the factorial of the provided integer.

If the integer is represented with the letter n, a factorial is the product of all positive integers less than or equal to n.

Factorials are often represented with the shorthand notation n!

For example: 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Arithmetic Operators**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Arithmetic_Operators)

Code:

function factorialize(num) {

var factorial = 1;

var randomNumber = 1;

while(randomNumber <= num) {

factorial = factorial \* randomNumber;

randomNumber++;

}

return factorial;

}

factorialize(5);

Check for Palindromes

Return true if the given string is a palindrome. Otherwise, return false.

A *palindrome* is a word or sentence that's spelled the same way both forward and backward, ignoring punctuation, case, and spacing.

**Note**  
You'll need to remove **all non-alphanumeric characters** (punctuation, spaces and symbols) and turn everything lower case in order to check for palindromes.

We'll pass strings with varying formats, such as "racecar", "RaceCar", and "race CAR" among others.

We'll also pass strings with special symbols, such as "2A3\*3a2", "2A3 3a2", and "2\_A3\*3#A2".

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.replace()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/replace)
* [**String.prototype.toLowerCase()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/toLowerCase)

Code:

function palindrome(str) {

// Good luck!

var anyString = /[\W\_]/g;

var convertedString = str.toLowerCase().replace(anyString,'');

var palindromeString = convertedString.split('').reverse().join('');

return palindromeString === convertedString;

}

palindrome("never odd or even");

Find the Longest Word in a String

Return the length of the longest word in the provided sentence.

Your response should be a number.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.split()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split)
* [**String.length**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/length)

Code:

function findLongestWord(str) {

var splitString = str.split(' ');

var lengthOfI = [];

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

for (var j = 0; j < splitString[i].length; j++) {

var jLength = splitString[i].length;

lengthOfI.push(jLength);

}

}

var sortArray = lengthOfI.sort(function(a, b) {

return b - a;

});

return sortArray[0];

}

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

#### Title Case a Sentence

Return the provided string with the first letter of each word capitalized. Make sure the rest of the word is in lower case.

For the purpose of this exercise, you should also capitalize connecting words like "the" and "of".

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.split()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split)

Code:

function titleCase(str) {

var strLow = str.toLowerCase();

var strSplit = strLow.split(' ');

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

strSplit[i] = strSplit[i].charAt(0).toUpperCase() + strSplit[i].slice(1);

}

return strSplit.join(' ');

}

titleCase("I'm a little tea pot");

#### Return Largest Numbers in Arrays

Return an array consisting of the largest number from each provided sub-array. For simplicity, the provided array will contain exactly 4 sub-arrays.

Remember, you can iterate through an array with a simple for loop, and access each member with array syntax arr[i].

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Comparison Operators**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Comparison_Operators)

Code:

function largestOfFour(arr) {

// You can do this!

var largestArray = [0, 0, 0, 0];

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

for (var j = 0; j < arr[i].length; j++) {

if (arr[i][j] > largestArray[i]) {

largestArray[i] = arr[i][j];

}

}

}

return largestArray;

}

largestOfFour([[4, 5, 1, 3], [13, 27, 18, 26], [32, 35, 37, 39], [1000, 1001, 857, 1]]);

#### Confirm the Ending

Check if a string (first argument, str) ends with the given target string (second argument, target).

This challenge can be solved with the .endsWith()method, which was introduced in ES2015. But for the purpose of this challenge, we would like you to use one of the JavaScript substring methods instead.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.substr()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/substr)
* [**String.prototype.substring()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/substring)

Code:

function confirmEnding(str, target) {

// "Never give up and good luck will find you."

// -- Falcor

return str.substr(-target.length) === target;

}

confirmEnding("Bastian", "n");

#### Repeat a string repeat a string

Repeat a given string (first argument) num times (second argument). Return an empty string if num is not a positive number.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Global String Object**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String)

Code:

function repeatStringNumTimes(str, num) {

// repeat after me

if (num > 0) {

return str.repeat(num);

} else {

return "";

}

}

repeatStringNumTimes("abc", -2);

#### Truncate a string

Truncate a string (first argument) if it is longer than the given maximum string length (second argument). Return the truncated string with a ... ending.

Note that inserting the three dots to the end will add to the string length.

However, if the given maximum string length num is less than or equal to 3, then the addition of the three dots does not add to the string length in determining the truncated string.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.slice()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/slice)

Code:

function truncateString(str, num) {

// Clear out that junk in your trunk

if (str.length > num && num > 3) {

return str.slice(0, num - 3) + "...";

} else if (str.length > num && num <= 3) {

return str.slice(0, num) + "...";

} else {

return str;

}

}

truncateString("A-tisket a-tasket A green and yellow basket", 11);

#### Chunky Monkey

Write a function that splits an array (first argument) into groups the length of size (second argument) and returns them as a two-dimensional array.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Array.prototype.push()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push)
* [**Array.prototype.slice()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/slice)

Code:

function chunkArrayInGroups(arr, size) {

// Break it up.

var newArr = [];

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

newArr.push(arr.slice(i, i + size));

}

return newArr;

}

chunkArrayInGroups(["a", "b", "c", "d"], 2);

#### Slasher Flick

Return the remaining elements of an array after chopping off n elements from the head.

The head means the beginning of the array, or the zeroth index.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Array.prototype.slice()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/slice)
* [**Array.prototype.splice()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice)

Code:

function slasher(arr, howMany) {

// it doesn't always pay to be first

arr.splice(0, howMany);

return arr;

}

slasher([1, 2, 3], 2);

#### Mutations

Return true if the string in the first element of the array contains all of the letters of the string in the second element of the array.

For example, ["hello", "Hello"], should return true because all of the letters in the second string are present in the first, ignoring case.

The arguments ["hello", "hey"] should return false because the string "hello" does not contain a "y".

Lastly, ["Alien", "line"], should return true because all of the letters in "line" are present in "Alien".

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**String.prototype.indexOf()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/indexOf)

Code: function mutation(arr) {

var matata = arr[1].toLowerCase();

var hakuna = arr[0].toLowerCase();

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

if (hakuna.indexOf(matata[i]) < 0)

return false;

}

return true;

}

mutation(["hello", "Hello"]);

#### Falsy Bouncer

Remove all falsy values from an array.

Falsy values in JavaScript are false, null, 0, "", undefined, and NaN.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Boolean Objects**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Boolean)
* [**Array.prototype.filter()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter)

Code:

function bouncer(arr) {

// Don't show a false ID to this bouncer.

var sam = arr.filter(Boolean,"");

return sam;

}

bouncer([7, "ate", "", false, 9]);

#### Seek and Destroy

You will be provided with an initial array (the first argument in the destroyer function), followed by one or more arguments. Remove all elements from the initial array that are of the same value as these arguments.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Arguments object**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/arguments)
* [**Array.prototype.filter()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter)

Code:

function destroyer(arr) {

// Remove all the values

var args = arr.slice.call(arguments);

args.splice(0, 1);

return arr.filter(function(element) {

return args.indexOf(element) === -1;

});

}

destroyer([1, 2, 3, 1, 2, 3], 2, 3);

#### Where do I belong

Return the lowest index at which a value (second argument) should be inserted into an array (first argument) once it has been sorted. The returned value should be a number.

For example, getIndexToIns([1,2,3,4], 1.5) should return 1 because it is greater than 1 (index 0), but less than 2 (index 1).

Likewise, getIndexToIns([20,3,5], 19) should return 2 because once the array has been sorted it will look like [3,5,20] and 19 is less than 20 (index 2) and greater than 5 (index 1).

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Write your own code.

Here are some helpful links:

* [**Array.prototype.sort()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)

Code:

function getIndexToIns(arr, num) {

// Find my place in this sorted array.

var sam = arr.sort(function(a, b) {

return a - b;

});

var i = 0;

while (i < arr.length) {

if (arr[i] >= num) {

return parseInt(i);

}

i++;

}

return arr.length;

}

getIndexToIns([40, 60], 50);

#### Caesars Cipher

One of the simplest and most widely known ciphers is a Caesar cipher, also known as a shift cipher. In a shift cipher the meanings of the letters are shifted by some set amount.

A common modern use is the [**ROT13**](https://en.wikipedia.org/wiki/ROT13) cipher, where the values of the letters are shifted by 13 places. Thus 'A' ↔ 'N', 'B' ↔ 'O' and so on.

Write a function which takes a [**ROT13**](https://en.wikipedia.org/wiki/ROT13) encoded string as input and returns a decoded string.

All letters will be uppercase. Do not transform any non-alphabetic character (i.e. spaces, punctuation), but do pass them on.

Remember to use [**Read-Search-Ask**](https://github.com/FreeCodeCamp/freecodecamp/wiki/FreeCodeCamp-Get-Help) if you get stuck. Try to pair program. Write your own code.

Here are some helpful links:

* [**String.prototype.charCodeAt()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/charCodeAt)
* [**String.fromCharCode()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/fromCharCode)

Code:

function rot13(str) { // LBH QVQ VG!

return str.split('').map.call(str, function(char) {

i = char.charCodeAt(0);

if (i < 65 || i > 90) {

return String.fromCharCode(i);

} else if (i < 78) {

return String.fromCharCode(i + 13);

}

return String.fromCharCode(i - 13);

}).join('');

}

// Change the inputs below to test

rot13("SERR PBQR PNZC");