**Introduction to Basic Algorithm Scripting**

In solving problems, break them down into smaller chunks. Then solve each chunk one by one.

Tip: To debug problems, try using console.log() to log variable values to the console.

**1. Convert Celsius to Fahrenheit**

function convertToF(celsius) {

let fahrenheit;

fahrenheit = (celsius \* 9/5) + 32;

return fahrenheit;

}

convertToF(30);

2. Reverse a String (Always ask in an interview)

function reverseString(str) {

return str.split("").reverse().join("");

}

reverseString("hello");

Another solution:

|  |
| --- |
| function reverseString(str) { |
|  | var newString = ""; |
|  | for (var i = str.length - 1; i >= 0; i--) { |
|  | newString += str[i]; |
|  | } |
|  | return newString; |
|  | } |
|  | reverseString('hello'); |

|  |
| --- |
| function reverseString(str) { |
|  | if (str === "") |
|  | return ""; |
|  | else |
|  | return reverseString(str.substr(1)) + str.charAt(0); |
|  | } |
|  | reverseString("hello"); |

3. Factorialize a Number

function factorialize(num) {

if(num === 0) {

return 1;

}

return num \* factorialize(num -1);

}

factorialize(5);

another soln:

function factorialize(num) {

for (a = 1;num >= 1; num--) {

a = num \* a;

}

return a;

}

factorialize(5);

**4. Find the Longest Word in a String**

function findLongestWordLength(str) {

var splitStr = str.split(' ');

var longest = 0;

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

if(splitStr[i].length > longest) {

longest = splitStr[i].length;

}

}

return longest;

}

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

Another soln

|  |
| --- |
| 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"); |

|  |
| --- |
| 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"); |

**5. Return Largest Numbers in Arrays**

function largestOfFour(arr) {

// You can do this!

var largestNumbers = [];

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

var largest = arr[i][0];

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

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

largest = arr[i][j];

}

}

largestNumbers[i] = largest;

}

return largestNumbers;

}

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

**6. Confirm the Ending**

function confirmEnding(str, target) {

if (str.substr(-target.length) === target) {

return true;

} else {

return false;

}

}

confirmEnding("Bastian", "n");

Another Solution:

Ternary Operator: (string.substr(-target.length) === target) ? true : false;

function confirmEnding(string, target) {

|  |  |
| --- | --- |
|  | return string.endsWith(target); // 'Bastian'.endsWith('n') |
|  | } |
|  | confirmEnding('Bastian', 'n'); |

**7. Repeat a String Repeat a String**

function repeatStringNumTimes(str, num) {

var repeatStrings = "";

while (num > 0) {

repeatStrings += str;

num--;

}

return repeatStrings;

}

repeatStringNumTimes("abc", 3);

Another Solution

Condition & Recursion

|  |
| --- |
| function repeatStringNumTimes(string, times) {  if(times < 0)  return "";  if(times === 1)  return string;  else  return string + repeatStringNumTimes(string, times - 1);  }  repeatStringNumTimes("abc", 3); |

**Repeat Method:**

function repeatStringNumTimes(string, times) {

if (times > 0)

return string.repeat(times);

else

return "";

}

repeatStringNumTimes("abc", 3);

**Ternary Operator**

function repeatStringNumTimes(string, times) {

return times > 0 ? string.repeat(times) : "";

}

repeatStringNumTimes("abc", 3);

**8. Truncate a String**

function truncateString(str, num) {

var truncatedStr = "";

if(str.length > num) {

truncatedStr = str.slice(0,num) + "...";

} else {

truncatedStr = str;

}

return truncatedStr;

}

truncateString("A-tisket a-tasket A green and yellow basket", 8);

**9) Finders Keepers**

function findElement(arr, func) {

let num = 0;

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

if(func(arr[i])){

num = arr[i];

return num;

}

}

return undefined;

}

findElement([1, 2, 3, 4], num => num % 2 === 0);

Another Solution:

function findElement(arr, func) {

let num = 0;

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

if(func(arr[i])){

num = arr[i];

return num;

}

}

return undefined;

}

findElement([1, 2, 3, 4], num => num % 2 === 0);

**10. Boo Who**

function booWho(bool) {

// What is the new fad diet for ghost developers? The Boolean.

return typeof bool === 'boolean';

}

booWho(null);

**11. Title Case a Sentence**

function titleCase(str) {

str = str.toLowerCase().split(' ');

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

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

}

return str.join(' ');

}

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

Another Solution: map() method

function titleCase(str) {

return str.toLowerCase().split(' ').map(function(word) {

return (word.charAt(0).toUpperCase() + word.slice(1));

}).join(' ');

}

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

map() and replace()

function titleCase(str) {

return str.toLowerCase().split(' ').map(function(word) {

return word.replace(word[0], word[0].toUpperCase());

}).join(' ');

}

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

**12. Slice and Splice**

function frankenSplice(arr1, arr2, n) {

let newArr = arr2.slice();

newArr.splice(n, 0, ...arr1);

return newArr;

}

frankenSplice([1, 2, 3], [4, 5, 6], 1); [4, 1, 2, 3, 5]

**13. Falsy Bouncer**

function bouncer(arr) {

return arr.filter(Boolean);

}

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

**14. Where do I Belong**

function getIndexToIns(arr, num) {

var newArr = arr.sort(function(a,b) {return a-b});

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

if(newArr[i] >= num){

return i;

}

}

return newArr.length;

}

getIndexToIns([40, 60], 50);

Another solution:

function getIndexToIns(arr, num) {

arr.push(num);

arr.sort(function(a, b){return a-b});

return arr.indexOf(num);

}

**15. Mutations**

function mutation(arr) {

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

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

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

if(firstArr.indexOf(arrToFind[i]) === -1) {

return false;

}

}

return true;

}

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

**16. Chunky Monkey**

function chunkArrayInGroups(arr, size) {

var newArr = [];

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

var chunk = arr.slice(i, i+size);

newArr.push(chunk);

}

return newArr;

}

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