Contents

[JavaScript Method - Boolean 3](#_Toc527094779)

[Modify the function to return true if both arguments are true, and return false otherwise. 4](#_Toc527094780)

[Modify the funtion to return the average score 5](#_Toc527094781)

[JavaScript Method - Concatenation 6](#_Toc527094782)

[Modify the function to return true if the array contains a 0, and returns false otherwise. 7](#_Toc527094783)

[Modify the function to return the longest string in the given array. 8](#_Toc527094784)

[Add a method to the Person's prototype called "getInitials" that returns the first letter of their first and last name, both capitalized. 9](#_Toc527094785)

[Modify the function to return the average age of everyone in the people array, rounded to the nearst integer 10](#_Toc527094786)

[Modify the function to return first and last initials 11](#_Toc527094787)

[Add a method to the Person's prototype called "shoutName" that returns the person's name in all uppercase letters. 12](#_Toc527094788)

[Modify the function to insert the given number into the exact middle of the array and return the array. 13](#_Toc527094789)

[Modify the function to return true if the given decimal is even when it is rounded to the nearest integer, and false otherwise. 14](#_Toc527094790)

[Modify the function to push the given item onto the end of the array and return the array. 15](#_Toc527094791)

[Modify the function to return true if the given number is an integer, and false otherwise. 16](#_Toc527094792)

[Modify the function to return the number of digits in the given number. 17](#_Toc527094793)

[Modify the function to invoke the given callback and return its result. 18](#_Toc527094794)

[Modify the function to transform the given string into an array where each character in the array is in the same index as it was in the string, and return it. 19](#_Toc527094795)

[Modify the function to return true if the array contains a 0, and returns false otherwise. 20](#_Toc527094796)

[Modify the function to return the given string reversed. 21](#_Toc527094797)

[Modify the function to add all of the digits of the given number together and return the sum. 22](#_Toc527094798)

[Modify the function to return the given string with an exclamation mark added to the end of it. 23](#_Toc527094799)

[Modify the function to return the number of digits in the given number. 24](#_Toc527094800)

[Modify the function to return a new array of names of the people who are 16 or older. 25](#_Toc527094801)

[Modify the function to return a copy of the given array in reverse order. 26](#_Toc527094802)

[Modify the function to return the largest number in the given array. 27](#_Toc527094803)

[Modify the function to return the total summed age of all of the people. 28](#_Toc527094804)

[Modify the function to return the count of how many people are 16 or older. 29](#_Toc527094805)

[Add a method to the Person's prototype called "isLegalDriver" that returns true if the person is 16 or older. 30](#_Toc527094806)

[Modify the function to return the number of vowels (a e i o u, ignore y) in the given string. 31](#_Toc527094807)

# JavaScript Method - Boolean

function Person(name, age) {

this.name = name;

this.age = age;

this.driverAge = isLegalDriver;

}

function isLegalDriver() {

let ageVerify = Boolean(this.age >= 16);

return ageVerify;

}

/\* Do not modify code below this line \*/

const youngPerson = new Person('Jane', 15);

console.log(youngPerson.driverAge(), '<-- should be false');

const olderPerson = new Person('Joan', 16);

console.log(olderPerson.driverAge(), '<-- should be true');

# Modify the function to return true if both arguments are true, and return false otherwise.

function areBothTrue(bool1, bool2) {

if (bool1 && bool2) {

return true;

}

return false;

}

/\* Do not modify code below this line \*/

console.log(areBothTrue(true, false), '<-- should be false');

console.log(areBothTrue(true, true), '<-- should be true');

# Modify the funtion to return the average score

function getAverageTestScore(scores) {

let total = 0, i;

for (i = 0; i < scores.length; i += 1) {

total += scores[i];

}

return total / scores.length;

}

/\* Do not modify code below this line \*/

const avg = getAverageTestScore([80, 100]);

console.log(avg, '<-- should be 90');

# JavaScript Method - Concatenation

function addExclamation(s) {

let newString = this.newString;

let exclam = this.exlclam;

this.exclam = "!";

this.newString = s.concat(this.exclam);

return this.newString;

}

/\* Do not modify code below this line \*/

console.log(addExclamation('hello world'), '<-- should be "hello world!"');

# Modify the function to return true if the array contains a 0, and returns false otherwise.

function doesArrayContainZero(numbers) {

var n = numbers.includes(0);

return n;

}

/\* Do not modify code below this line \*/

console.log(doesArrayContainZero([1, 2, 3]), '<-- should be false');

console.log(doesArrayContainZero([1, 0, 2]), '<-- should be true');

# Modify the function to return the longest string in the given array.

function getLongestString(strings) {

var lg = 0;

var longest;

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

if (strings[i].length > lg) {

lg = strings[i].length;

longest = strings[i];

}

}

return longest;

}

/\* Do not modify code below this line \*/

const strings = ['long', 'longer', 'longest'];

console.log(getLongestString(strings), '<-- should be "longest"');

# Add a method to the Person's prototype called "getInitials" that returns the first letter of their first and last name, both capitalized.

function Person(firstName, lastName) {  
  this.firstName = firstName;  
  this.lastName = lastName;  
  }  
  Person.prototype.getInitials = function() {  
    var fni = this.firstName.charAt(0);  
    fni = fni.toUpperCase();  
    var lni = this.lastName.charAt(0);  
    lni = lni.toUpperCase();  
    var fnilni = fni + lni;  
    return fnilni;  
  }

/\* Do not modify code below this line \*/

const johnDoe = new Person('john', 'doe');  
console.log(johnDoe.getInitials(), '<-- should be "JD"');

# Modify the function to return the average age of everyone in the people array, rounded to the nearst integer

function getRoundedAverageAge(people) {

  let cnt = people.length;

  let x = 0;

  for (i = 0; i < cnt; i++) {

    x += people[i].age;

  }

  return Math.ceil(x/cnt);

}

/\* Do not modify code below this line \*/

const examplePeopleArray = [

  { name: 'John', age: 19 },

  { name: 'Jack', age: 21 },

  { name: 'Jane', age: 22 }

];

console.log(getRoundedAverageAge(examplePeopleArray), '<-- should be 21');

# Modify the function to return first and last initials

function Person(firstName, lastName) {  
  this.firstName = firstName;  
  this.lastName = lastName;  
  }  
  Person.prototype.getInitials = function() {  
    var fni = this.firstName.charAt(0);  
    fni = fni.toUpperCase();  
    var lni = this.lastName.charAt(0);  
    lni = lni.toUpperCase();  
    var fnilni = fni + lni;  
    return fnilni;  
  }

/\* Do not modify code below this line \*/

const johnDoe = new Person('john', 'doe');  
console.log(johnDoe.getInitials(), '<-- should be "JD"');

# Add a method to the Person's prototype called "shoutName" that returns the person's name in all uppercase letters.

function Person(name) {

Person.prototype.shoutName = function() {

return name.toUpperCase();

}

}

/\* Do not modify code below this line \*/

const john = new Person('John');

console.log(john.shoutName(), '<-- should be "JOHN"');

# Modify the function to insert the given number into the exact middle of the array and return the array.

function insertIntoMiddle(array, item) {

var middle = array.length / 2;

array.splice(middle,0,item);

return array;

}

/\* Do not modify code below this line \*/

const items = insertIntoMiddle([1, 3], 2);

console.log(insertIntoMiddle([1, 3], 2), '<-- should be [1, 2, 3]');

console.log(insertIntoMiddle([1, 3, 7, 9], 5), '<-- should be [1, 3, 5, 7, 9]');

# Modify the function to return true if the given decimal is even when it is rounded to the nearest integer, and false otherwise.

function isRoundedNumberEven(decimal) {

let y = Math.round(decimal);

x = (y % 2) == 0;

return x;

}

/\* Do not modify code below this line \*/

console.log(isRoundedNumberEven(2.2), '<-- should be true');

console.log(isRoundedNumberEven(2.8), '<-- should be false');

# Modify the function to push the given item onto the end of the array and return the array.

function addItemToArray(array, item) {

array.push(item);

return array;

}

/\* Do not modify code below this line \*/

const items = addItemToArray([1, 2, 3], 4);

console.log(items, '<-- should equal [1, 2, 3, 4]');

function insertIntoMiddle(array, item) {

var middle = array.length / 2;

array.splice(middle,0,item);

return array;

}

/\* Do not modify code below this line \*/

const items = insertIntoMiddle([1, 3], 2);

console.log(insertIntoMiddle([1, 3], 2), '<-- should be [1, 2, 3]');

console.log(insertIntoMiddle([1, 3, 7, 9], 5), '<-- should be [1, 3, 5, 7, 9]');

# Modify the function to return true if the given number is an integer, and false otherwise.

function isInteger(num) {

return x = Number.isInteger(num);

}

/\* Do not modify code below this line \*/

console.log(isInteger(1), '<-- should be true');

console.log(isInteger(1.5), '<-- should be false');

# Modify the function to return the number of digits in the given number.

function getDigitCount(num) {

return int\_length = (''+num).length;

}

/\* Do not modify code below this line \*/

console.log(getDigitCount(246), '<-- should be 3');

# Modify the function to invoke the given callback and return its result.

function invokeCallback(cb) {

return cb('hello');

}

/\* Do not modify code below this line \*/

const exampleCallback = function() {

return 'hello';

};

const invokedResult = invokeCallback(exampleCallback);

console.log(invokedResult, '<-- should be "hello"');

# Modify the function to transform the given string into an array where each character in the array is in the same index as it was in the string, and return it.

function convertStringToArray(s) {

return s = s.split("");

}

/\* Do not modify code below this line \*/

const exampleString = 'hello';

const stringAsArray = convertStringToArray(exampleString);

console.log(stringAsArray, '<-- should be ["h", "e", "l", "l", "o"]');

# Modify the function to return true if the array contains a 0, and returns false otherwise.

function doesArrayContainZero(numbers) {

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

if (numbers[i] === 0) {

return true;

}

}

return false;

}

/\* Do not modify code below this line \*/

console.log(doesArrayContainZero([1, 2, 3]), '<-- should be false');

console.log(doesArrayContainZero([1, 0, 2]), '<-- should be true');

# Modify the function to return the given string reversed.

function reverseString(s)  
{  
    var splitString = s.split(""); // var splitString = "hello".split("");  
    var reverseArray = splitString.reverse(); // var reverseArray = ["h",   "e", "l", "l", "o"].reverse();  
    var joinArray = reverseArray.join(""); // var joinArray = ["o", "l", "l", "e", "h"].join("");  
    return joinArray; // "olleh"  
}

/\* Do not modify code below this line \*/

console.log(reverseString('hello'), '<-- should be "olleh"');

# Modify the function to add all of the digits of the given number together and return the sum.

function getSumOfDigits(num) {

let x = ("" + num).split("");

let result = 0;

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

n = Number(x[i]);

result += n;

console.log(result);

}

return result;

}

/\* Do not modify code below this line \*/

console.log(getSumOfDigits(42), '<-- should be 6');

console.log(getSumOfDigits(103), '<-- should be 4');

# Modify the function to return the given string with an exclamation mark added to the end of it.

function addExclamation(s) {

s = s + '!';

return s;

}

/\* Do not modify code below this line \*/

console.log(addExclamation('hello world'), '<-- should be "hello world!"');

# Modify the function to return the number of digits in the given number.

function getDigitCount(num) {

x = num.toString().length;

return x;

}

/\* Do not modify code below this line \*/

console.log(getDigitCount(246), '<-- should be 3');

# Modify the function to return a new array of names of the people who are 16 or older.

function getNamesOfLegalDrivers(people) {

let cnt = people.length;

let array = [];

for (i = 0;i<cnt;i++) {

if (people[i].age >= 16) {

array.push(people[i].name);

}

}

return array;

}

/\* Do not modify code below this line \*/

const examplePeopleArray = [

{ name: 'John', age: 14 },

{ name: 'Joey', age: 16 },

{ name: 'Jane', age: 18 }

];

console.log(getNamesOfLegalDrivers(examplePeopleArray), '<-- should be ["Joey", "Jane"]');

# Modify the function to return a copy of the given array in reverse order.

function copyAndReverseArray(array) {

return array = array.slice().reverse();

}

/\* Do not modify code below this line \*/

const original = [1, 2, 9, 8];

const reversed = copyAndReverseArray(original);

console.log(original, '<-- should still be [1, 2, 9, 8]');

console.log(reversed, '<-- should be [8, 9, 2, 1]');

# Modify the function to return the largest number in the given array.

function getLargestNumber(numbers) {

return Math.max(...numbers);

}

/\* Do not modify code below this line \*/

const largestNumber = getLargestNumber([1, 9, 5]);

console.log(largestNumber, '<-- should be 9');

# Modify the function to return the total summed age of all of the people.

function getSummedAge(people) {

let cnt = people.length;

let x = 0;

for (i = 0; i < cnt; i++) {

x += people[i].age;

}

return x;

}

/\* Do not modify code below this line \*/

const examplePeopleArray = [

{ name: 'John', age: 10 },

{ name: 'Jack', age: 20 },

{ name: 'Jane', age: 25 }

];

console.log(getSummedAge(examplePeopleArray), '<-- should be 55');

# Modify the function to return the count of how many people are 16 or older.

function countLegalDrivers(people) {

let cnt = people.length;

let array = [];

let total = 0;

for (i = 0;i<cnt;i++) {

if (people[i].age >= 16) {

total += 1;

}

}

return total;

}

/\* Do not modify code below this line \*/

const examplePeopleArray = [

{ name: 'John', age: 15 },

{ name: 'Jane', age: 16 },

{ name: 'Jack', age: 17 }

];

console.log(countLegalDrivers(examplePeopleArray), '<-- should be 2');

# Add a method to the Person's prototype called "isLegalDriver" that returns true if the person is 16 or older.

function Person(name, age) {

this.name = name;

this.age = age;

}

Person.prototype.isLegalDriver = function() {

x = this.age >= 16;

return x;

}

/\* Do not modify code below this line \*/

const youngPerson = new Person('Jane', 15);

console.log(youngPerson.isLegalDriver(), '<-- should be false');

const olderPerson = new Person('Joan', 16);

console.log(olderPerson.isLegalDriver(), '<-- should be true');

# Modify the function to return the number of vowels (a e i o u, ignore y) in the given string.

function getVowelCount(s) {

cnt = s.length;

vowelcnt = 0;

for (i = 0; i < cnt; i++) {

if (s.charAt(i).match(/[aeiouAEIOU]/)) {

vowelcnt ++;

}

}

return vowelcnt;

}

/\* Do not modify code below this line \*/

console.log(getVowelCount('hello world'), '<-- should be 3');