Java Script Assignment

**1. Find the smallest number in an array**

Create a function that will display the smallest value in the array.

<html>

<head>

<title>smallest no</title>

</head>

<body>

<h1>The smallest number in the array:</h1>

<script>

document.writeln(Math.min(30, 45, 60, 7));

</script>

</body>

</html>

2. Sort strings by Alphabetical Order

Function that will return your string in Alphabetical order

<!DOCTYPE html>

<html>

<h3>Alphabatical Order </h3>

<body>

<p id="demo"> </p>

<script >

var str = "hello hai";

document.getElementById("demo").innerHTML = alphabeticalOrder(str);

function alphabeticalOrder(str) {

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

}

</script>

</body>

</html>

**3. Factorialize a number**

In mathematics, the factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. In simple terms, the Factorial of 7 is solved like this:

**7 \_ 6 \_ 5 \_ 4 \_ 3 \_ 2 \_ 1 = 5,040**

<html>

<head>

<title>factorial</title>

</head>

<body>

<script>

function factorializer(int) {

if (int <= 1) {

return 1;

} else {

return int \* factorializer(int - 1);

}

}

console.log(factorializer(7));

</script>

</body>

</html>

**4. Identify if a number is Odd or Even?**

A function that lets you know if a number is Even or Odd

<html>

<head>

<title>odd/even</title>

</head>

<body>

<script>

function oddOrEven(int) {

if (int %2!=0) {

return "odd";

} else {

return "Even";

}

}

console.log(oddOrEven(7));

</script>

  </body>

</html>

**5. Eliminate all odd numbers in an array.**

Remove all Odd number(s) in an array and return a new array that contains Even numbers only

<html>

<head>

<title>eliminate all odd no</title>

</head>

<body>

<script>

function evenOnly(arr){

result=arr.filter(arr=> arr % 2 ==0);

return result;

}

console.log(evenOnly([1, 2, 3, 4, 5, 6]));

</script>

</body>

</html>

**6. Return numbers only**

Create a function that will accept an array, check the data type of each element. The function will delete string elements and will return a the new array

<html>

<head>

<title>return numbers</title>

</head>

<body>

<script>

function numbersOnly(arr){

return arr.filter(arr=>typeof arr=="number");

}

console.log(numbersOnly(['text', 3, 7, 'github', 13, 'dev']));

</script>

    </body>

</html>

**7. Add up the numbers**

Return the sum of a number going back to it's root. In other words, the function will work like this:

**addUp(5);**

// 5 + 4 + 3 + 2 + 1 + 0 = **15**

<html>

<head>

<title></title>

</head>

<body>

<script>

function addUp(num) {

if (num <= 1) {

return num;

} else {

return num + addUp(num - 1);

}

}

console.log(addUp(8));

</script>

  </body>

</html>

**8. Return the Min, Max, Length and Average of an Array**

Create a function that will accept an array and do the following:

* Get the lowest element
* Get the highest element
* Get the length of array
* Get the Average of all element;
* Store these criteria in a new array

<html>

<head>

</head>

<body>

<script>

function minMaxLengthAverage(arr) {

const min = Math.min(...arr);

const max = Math.max(...arr);

const len = arr.length;

const ave = arr => arr.reduce((acc, curVal) => acc + curVal, 0) / len;

const average = ave(arr);

return [min, max, len, average];

}

console.log(minMaxLengthAverage([7, 13, 3, 77, 100]));

</script>

  </body>

</html>

**9. Sort Numbers in Ascending Order**

Array.sort() sorts the **strings** alphabetically. What if we want to sort **numbers** from lowest to highest? Will it produce a correct output?

<html>

<head>

</head>

<body>

<script>

function sortNumsAscending(arr) {

let sorter = (a, b) => {

return a - b;

};

if (arr == []) {

return [];

} else if (arr == null) {

return [];

} else {

return arr.sort(sorter);

}

}

console.log(sortNumsAscending([7, 13, 3, 77, 100]));

</script>

  </body>

</html>

**10. Convert Numbers in Roman Numerals**

Convert the given number to a Roman Numeral

<html>

<head>

</head>

<body>

<script>

function romanNumbers(num) {

let values = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1];

let romanNumerals = [

"M",

"CM",

"D",

"CD",

"C",

"XC",

"L",

"XL",

"X",

"IX",

"V",

"IV",

"I"

];

let roman = "";

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

while (values[i] <= num) {

roman += romanNumerals[i];

num -= values[i];

}

}

return roman;

}

console.log(romanNumbers(1989));

</script>

  </body>

</html>

**11.Absolute Sum**

Return the absolute sum of all the array elements

<html>

<head>

</head>

<body>

<script>

function getAbsSum(arr) {

const reducer = (acc, currVal) => {

return acc + currVal;

};

return Math.abs(arr.reduce(reducer));

}

console.log(getAbsSum([-1, -3, -5, -4, -10, 0]));

</script>

  </body>

</html>

**12.Lopping a triangle**

Form a triangle using hash tags

**Example:**

> #

> ##

> ###

> ####

> #####

<html>

<h3>hash pattern</h3>

<body>

<script >

var num=10;

for(var i=1; i <= num; i++)

{

for(var j=1; j<=i; j++)

{

document.write('#');

}

document.write('<br />');

}

</script>

</body>

</html>

**13.Count the number of words**

Return how many words was given

<html>

<head>

</head>

<body>

<script>

function countWords(str) {

return str.split(" ").length;

}

console.log(countWords('hello gm have a great day'));

</script>

  </body>

</html>

**14.Multiply by length**

Multiply all elements in an array by it's length

<html>

<head>

</head>

<body>

<script>

function MultiplyByLength(arr) {

let len = arr.length;

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

arr[i] = arr[i] \* len;

}

return arr;

}

console.log(MultiplyByLength([4,1,1]));

</script>

  </body>

</html>

**15. Repeating letters**

Create a function that will repeat each string character two times

<html>

<head>

</head>

<body>

<script>

function doubleChar(str) {

let x = str.split("");

return x.map(x => x.repeat(2)).join("");

}

console.log(doubleChar('renu'));

</script>

  </body>

</html>

**Array Functions Assignment**

16. Create a function named reversePlusOne. This function should:

* + Take one argument, an array of at least two numbers.
  + This function should return:
    - the array *reversed* with a 1 added at the front

For example:

reversePlusOne([1,2]); // returns [1,2,1]

reversePlusOne([5,4,3,2]); // returns [1,2,3,4,5]

* Create a function named plusesEverywhere. This function should:
  + Take one argument, an array of at least two numbers.
  + This function should return:
    - a String made of all the values in the array separated by +

For example:

plusesEverywhere([1,2,3]); // returns "1+2+3"

plusesEverywhere([18,24]); // returns "18+24"

* Create a function named arrayQuantityPlusOne. This function should:
  + Take one argument, an array of numbers.
  + This function should return:
    - one greater than the number of items in the array

For example:

arrayQuantityPlusOne([0,0,1,0,2,1]); // returns 7

arrayQuantityPlusOne([42]); // returns 2

<!DOCTYPE html>>

<head>

</head>

<body>

<script>

//--------------16a---------------

var array=[4,3,2];

function reverseplusone(array){

array.reverse();

array.unshift(1);

console.log(array);

}

reverseplusone(array);

//---------------16b------------

let array11=[1,2,3];

function pluseverywhere(array){

return array.join("+");

}

console.log(pluseverywhere(array11));

//----------------------16c--------

let number22=[1,2,3];

function quantityplusone(array){

console.log(array.length+1);

}

quantityplusone(number22);

</script>

</body>

</html>

17. Complete the createCourse function. This function should:

* + take three arguments that will define course properties
    - courseTitle (string)
    - courseDuration (string)
    - courseStudents (array)
  + return an object that has each property assigned its proper value

For example:

createCourse('Bloc Front-End Engineering', '4 weeks', ['Joe', 'Tim', 'Rob'])

// should return {title: 'Bloc Front-End Engineering', duration: '4 weeks', students: ['Joe', 'Tim', 'Rob']}

* Complete the addProperty function. This function should:
  + Take three arguments:
    - object: an object to add a properties to
    - newProp: a property that we want to add to the object
    - newValue: a value that we want the new property to have
  + If object doesn't already have a property named newProp, then add newProp with value of newValue to object
  + If object already has newProp, return the object argument.

For example:

addProperty({}, 'firstName', 'Jim') // should return { firstName: 'Jim' }

addProperty({firstName: 'Rob'}, 'firstName', 'Jim') // should return {firstName: 'Rob'}

* Complete the formLetter function. This function should:
  + take one argument, a letter, which has three properties recipient, sender, and msg
  + combine the three properties into a single string with an additional greeting and closing
  + insert additional new lines between the greeting, message, and signature.

For example:

formLetter({ recipient: "James", sender: "Richard", msg: "Things are well." })

// should return "Hello James,\n\nThings are well.\n\nSincerely,\nRichard"

* Complete the canIGet function. This function should:
  + Take two arguments:
    - item: represents what the user wants to buy
    - money: represents how many dollars a user has
  + return true if a user can afford a given item according to the price chart below, and false otherwise:
    - 'MacBook Air' - $999
    - MacBook Pro' - $1299
    - 'Mac Pro' - $2499
    - 'Apple Sticker' - $1
  + Return false if the item is not in the above list of Apple products

Do this with 0 'if' conditions! (Hint: Place the above price table in an object).

For example:

canIGet('MacBook Air', 100) // returns false

<html>

<head>

</head>

<body>

<script>

//------------17a--------------

function createCourse (courseTitle, courseDuration, courseStudents)

{

var newObj =

{

title: courseTitle,

duration: courseDuration,

students: courseStudents

};

return newObj;

};

document.write(createCourse('Bloc Front-End Engineering', '4 weeks', ['Joe', 'Tim', 'Rob']));

//-----------17b----------------

function addProperty (object, newProp, newValue)

{

if(!object.hasOwnProperty(newProp))

{

object[newProp] = newValue;

}

return object;

};

document.write(addProperty({}, 'firstName', 'Jim'));

//--------------------17c-------------

function formLetter(object){

return "Hello " + object.recipient + ",\n\n" + object.msg + "\n\n" +

"Sincerely,\n" + object.sender;

};

document.write( formLetter({ recipient: "James", sender: "Richard", msg:

"Things are well." }));

//-------------17d------------

function canIGet(item, money){

var itemList = {

'MacBook Air': 999,

'MacBook Pro': 1299,

'Mac Pro': 2499,

'Apple Sticker': 1

};

return money >= itemList[item];

};

document.write(canIGet('MacBook Air', 100) );

document.write(canIGet('MacBook Air', 1000) );

</script>

    </body>

</html>

18. Complete the formLetter function. This function should:

* + Take three strings as arguments: the first name of the recipient, the sender's signature name, and the message of the letter
  + combine the three into a single string with additional greetings and closings
  + insert additional new lines between the greeting, message, and signature

For example:

Ex.formLetter("James", "Richard", "Things are well.");

...should return:

"Hello James,\n\nThings are well.\n\nSincerely,\nRichard"

* Complete the sliceItAndCombineIt function. This function should:
  + take a string and four indices (numbers)
  + return a new string which is the concatenation of two substrings marked by the first and second index of each pair of indices. For example:

sliceItAndCombineIt("This is a Test", 0, 4, 5, 7) // returns "Thisis"

sliceItAndCombineIt("This is a Test", 0, 4, 1, 2) // returns "Thish".

* Complete the findFirstMatch function. This function should:
  + Take two strings as arguments. The first string is the one to search, the second is the one to search for.
  + Return the position (i.e. index) of the first match of string being searching for

For example:

findFirst("Roses are red", "re") // returns 7 (the position of the "re" in "are")

* Complete the findLastMatch function. This function should:
  + Take two strings as arguments. The first string is the one to search, the second is the one to search for
  + Return the position (a.k.a. the index) of the last match of string we're searching for For example:

findFirst("Roses are red", "re") returns 10 (the position of the "re" in "red")

* Complete the substringBetweenMatches function. This function should:
  + Take two strings as arguments. The first string is the one to search, the second is the one to search for
  + Return the substring between the first match and the last match
  + Not include the first match or the last match in the returned substring For example:

findFirst("Roses are red, apples are really red.", "red") // returns ", apples are really "

<html>

<head>

</head>

<body>

<script>

//----------------18a----------------------

function formLetter(firstName, signatureName, message)

{

var newObj =

{

name: firstName,

signature: signatureName,

message: message

};

return newObj;

};

console.log(formLetter("James", "Richard", "Things are well.");)

//----------------------------------------------------------------------18b-----------------------------

var sliceItAndCombineIt=function(myString, a,b,c,d){

return myString.substring(a,b)+myString.substring(c,d);

}

console.log(sliceItAndCombineIt("This is a Test", 0, 4, 5, 7) );

//-------------------------18c-----------------------------------------------------

var findFirstMatch =function(myString,str){

return myString.indexOf(str);

}

console.log(findFirstMatch("Roses are red", "re"));

//----------------------------------------------------------------------18d-----------------------------

var findFirstMatch =function(myString,str){

return myString.lastIndexOf(str);

}

console.log(findFirstMatch("Roses are red", "re"));

//---------------------------------------------------------------------18e-----------------------------

var substringBetweenMatches =function(myString,str){

var strlength=str.length;

var firstsub=myString.indexOf(str)+strlength;

var lastsub=myString.lastIndexOf(str);

return myString.substring(firstsub,lastsub);

}

console.log(substringBetweenMatches("Roses are red, apples are really red.", "red") );

</script>

    </body>

</html>