**Problem**:

Write a function called “addFive”.  
Given a number, “addFive” returns 5 added to that number.  
  
Input:

addFive(5);

var num = 5;

function addFive(num) {

return num+5

}

var result = addFive(num)

addFive(-5);



var num = -5;

function addFive(num) {

return num+(5)

}

var result = addFive(num)

**Problem**:

Write a function called “getOpposite”.  
Given a number, return its opposite

Input:

getOpposite(5);  
getOpposite(0);   
getOpposite(-5);  
getOpposite(“5a”);

Output:

-5  
0  
5  
-1

answer🡪

var num;

function getOpposite(num)

{

If(num>0)

{

return num \* (-1)

}

If(num===0)

{

Return 0;

}

If(num<0)

{

return num \* (-1)

}

If(typeof(num)===typeof(“str”))

{

return “-1”

}

}

var result = getOpposite(num)

**Problem**:

Fill in your code that takes an number minutes and converts it to seconds.

Examples  
toSeconds(5) ➞ 300

toSeconds(3) ➞ 180

toSeconds(2) ➞ 120

**answer🡪**

var min;

function toSeconds(min)

{

Return num\*60

}

var secs = toSeconds(min)

**Problem**  
Create a function that takes a string and returns it as an integer.

Examples  
toInteger(“6”) ➞ 6

toInteger(“1000”) ➞ 1000

toInteger(“12”) ➞ 12

**answer🡪**

var mystr;

function toInteger(mystr)

{

Return parseInt(mystr)

}

var myint = toInteger(mystr)

**Problem**

Create a function that takes a number as an argument, increments the number by +1 and returns the result.

Examples  
nextNumber(0) ➞ 1

nextNumber(9) ➞ 10

nextNumber(-3) ➞ -2

**answer🡪**

var myint ;

function nextNumber(myint)

{

Return myint+1;

}

var myNextint = nextNumber(myint)

**Problem**

Create a function that takes an array and returns the first element.

Examples  
getFirstElement([1, 2, 3]) ➞ 1

getFirstElement([80, 5, 100]) ➞ 80

getFirstElement([-500, 0, 50]) ➞ -500

var arr = [1,2,3];

function getFirstElement(arr)

{

Return arr[0]

}var data = getFirstElement(arr)

**Problem**

Convert Hours into Seconds

Write a function that converts hours into seconds.

Examples  
hourToSeconds(2) ➞ 7200

hourToSeconds(10) ➞ 36000

hourToSeconds(24) ➞ 86400

var num;

function hourToSeconds(num)

{

Return num\*60\*60

}

var data = hourToSeconds(num)

**Problem**

Find the Perimeter of a Rectangle  
Create a function that takes height and width and finds the perimeter of a rectangle.

Examples  
findPerimeter(6, 7) ➞ 26

findPerimeter(20, 10) ➞ 60

findPerimeter(2, 9) ➞ 22

**answer🡪**

function findPerimeter(num1,num2)

{

Return 2\*(num1+num2)

}

var peri = findPerimeter(6,7)

**Problem**

Less Than 100?  
Given two numbers, return true if the sum of both numbers is less than 100. Otherwise return false.

**Answer🡪**

function lessThan100(num1,num2)

{

If((num1+num2)<100)

{

Return “true”

}

Else

{

Return “false”

}

}

var res = lessThan100(22,15)

There is a single operator in JavaScript, capable of providing the remainder of a division operation. Two numbers are passed as parameters. The first parameter divided by the second parameter will have a remainder, possibly zero. Return that value.

Examples  
remainder(1, 3) ➞ 1

remainder(3, 4) ➞ 3

remainder(-9, 45) ➞ -9

remainder(5, 5) ➞ 0

**Answer🡪**

function remainder(num1,num2)

{

var k=num1%num2

return k

}

var res = remainder(1,3)

**Problem**

Old macdonald had a farm:

MacDonald is asking you to tell him how many legs can be counted among all his animals. The farmer breeds three species:

turkey = 2 legs  
horse = 4 legs  
pigs = 4 legs

The farmer has counted his animals and he gives you a subtotal for each species. You have to implement a function that returns the total number of legs of all the animals.

Examples  
CountAnimals(2, 3, 5) ➞ 36

CountAnimals(1, 2, 3) ➞ 22

CountAnimals(5, 2, 8) ➞ 50

function CountAnimals(tur,horse,pigs)

{

Return (tur\*2 + horse\*4 + pigs\*4)

}

var legs = CountAnimals(2,3,5)

**Problem**:

Check if an Integer is Divisible By Five  
Create a function that returns true if an integer is evenly divisible by 5, and false otherwise.

Examples  
divisibleByFive(5) ➞ true

divisibleByFive(-55) ➞ true

divisibleByFive(37) ➞ false

function divisibleByFive(num1)

{

If(num1%5===0)

{

Return “true”

}

Else

{

Return “false”

}

}

var divisible = divisibleByFive(5)

**Problem**:

Write a function called “isEven”.  
Given a number, “isEven” returns whether it is even.  
  
Input:  
isEven(12);  
isEven(0);  
isEven(11);

function isEven(num){

if(num%2===0)

return “true”

else

return “false”

}var even = isEven(5)

**Problem**:  
Write a function called “areBothOdd”.  
Given 2 numbers, “areBothOdd” returns whether or not both of the given numbers are odd.  
  
Input:  
areBothOdd(1, 3);  
areBothOdd(1, 4);  
areBothOdd(2, 3);  
areBothOdd(0, 0)

function areBothOdd(num1, num2){  
if(num1%2!===0 && num2%2!==0)

{

Return “true”

}

Else

Return “false”

}

**Problem**:  
Write a function called “getFullName”.  
Given a first and a last name, “getFullName” returns a single string with the given first and last names separated by a single space.  
  
Input:

getFullName(“GUVI”, “GEEK”);  
getFullName(“GUVI”, );  
getFullName(, “GEEK”);  
getFullName(“”, “”);

**answer🡪**  
function getFullName(firstName, lastName)

{

return firstName+" "+lastName  
  
}

**Problem**:  
Write a function called “getLengthOfWord”.  
Given a word, “getLengthOfWord” returns the length of the given word.  
Input:

getLengthOfWord(“GUVI”);  
getLengthOfWord(“”);  
getLengthOfWord();  
getLengthOfWord(9);

Output:

4  
0  
-1  
-1

**Answer🡪**

function getLengthOfWord(word1){

return world1.length  
}

**Problem**:  
Write a function called “isSameLength”.  
Given two words, “isSameLength” returns whether the given words have the same length.  
Input:  
isSameLength(“GUVI”, “GEEK”);  
Output:  
true

**answer🡪**

function isSameLength(word1, word2){  
if(world1.length===world2.length)

{

Return “is same length”

}

Else

{

Return “not same length”

}

}

Write a function called “getNthElement”.  
Given an array and an integer, “getNthElement” returns the element at the given integer, within the given array. If the array has a length of 0, it should return ‘undefined’.  
Input:  
getNthElement([1, 3, 5], 1);  
Output:  
3

function getNthElement(array,n){

return array[n] }

**Problem**:

Write a function called “addProperty”.  
Given an object and a key, “addProperty” adds a new property on the given object with a value of true.  
  
var obj = {  
}  
Input:  
addProperty(obj, “mykey”);

Output:

{  
mykey: true  
}

Answer🡪

var obj = {

mykey: "value"

};

function addProperty(obj, key){

obj.mykey=key

}

addProperty(obj,"true")

console.log(obj)

**Problem**:

Return an array, where the first element is the count of positives numbers and the second element is sum of negative numbers.

answer—>

console.log(ar2); var arr = [-5, 10, -3, 12, -9, 5, 90, 0, 1];

var ar2=countPositivesSumNegatives(arr)

function countPositivesSumNegatives(arr)

{

var count=0

var sum=0

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

{

if(arr[i]>0)

{

count++

}

if(arr[i]<0)

{

sum+=arr[i]

}

}

return[count,sum]

}

console.log(ar2);

**Problem**:

Return an array, where the first element is the count of positives numbers and the second element is sum of negative numbers.

answer—>

console.log(ar2); var arr = [-5, 10, -3, 12, -9, 5, 90, 0, 1];

var ar2=countPositivesSumNegatives(arr)

function countPositivesSumNegatives(arr)

{

var count=0

var sum=0

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

{

if(arr[i]>0)

{

count++

}

if(arr[i]<0)

{

sum+=arr[i]

}

}

return[count,sum]

}

console.log(ar2);

**Problem**:

Create a function that receives an array of numbers and returns an array containing only the positive numbers

function getPositives(ar)

{

var t=[]

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

{

if(ar[i]>0)

{

t.push(ar[i])

}

}

return t

}

var ar = [-5, 10, -3, 12, -9, 5, 90, 0, 1];

var ar2 = getPositives(ar);

console.log(ar2);

**Problem**:

Write a function `powersOfTwo` which will return list of all powers of 2 from 0 to n (where n is an exponent).

n = 0 -> 2⁰ -> [1]

n = 1 -> 2⁰, 2¹ -> [1,2]

n = 2 -> 2⁰, 2¹, 2² -> [1,2,4]

Input:  
powersOfTwo(0)  
powersOfTwo(1)  
powersOfTwo(2)  
Output:  
1  
1,2  
1,2,4

Answer🡪

function powersOfTwo(n){

var a=[]

for(var i=0;i<=n;i++)

{

a.push(Math.pow(2,i))

}

return a.join(" ")

}

console.log(powersOfTwo(4))

**Problem**:

Find the maximum number in an array of numbers

Answer🡪

function findMax(ar)  
{  
var t=ar.sort((a,b)=>a-b)

return t[t.length-1]  
}

var ar = [-5, 10, -3, 12, -9, 5, 90, 0, 1];  
var max = findMax(ar);

**Problem**:

Print the first 100 prime numbers

printPrimes(100);

function printPrimes(nPrimes)

{

var n = 0;

var i = 2;

while(n < nPrimes)

{

if (isPrime(i))

{

console.log(n, " → ", i);

n++;

}

i++;

}

}

// Returns true if a number is prime

function isPrime(n)

{

var count=0

for(var i=1;i<100;i++)

{

if(n%i===0)

{

count++

}

}

if(count===2)

{

return n

}

}

**Problem**:

Reverse a string

var s = reverseString("JavaScript");

console.log(s);

function reverseString(s)

{

var t=s.split("")

var final=[]

for(var i=t.length-1;i>=0;i--)

{

final.push(t[i])

}

return final.join("")

}

**Problem**:

Create a function that will merge two arrays and return the result as a new array

var ar1 = [1, 2, 3];  
var ar2 = [4, 5, 6];

var ar = mergeArrays(ar1, ar2);  
console.log(ar);

function mergeArrays(ar1, ar2)  
{  
 var result =[]

for(var i=0;i<ar1.length;i++)

{

Result.push(ar1[i])

}

For(var j=0;j<ar2.length;j++)

{

Result.push(ar2[j])

}

return result;  
}