**Javascript Functions: Warup Problems**

1) **Problem**:

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

addFive(5);  
addFive(0);  
addFive(-5);

Output:

10  
5  
0

var num = 10;function addFive(num) {   
  
}var result = addFive(num)

Solution:

var num = 10;

function addFive(*num*) {

return *num*+5;

}var result = addFive(num)

console.log("Add Five:" + result);

2) **Problem**:

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

Input:

getOpposite(5);  
getOpposite(0);  
getOpposite(-5);  
getOpposite(“5a”);  
getOpposite(5.5);

Output:

-5  
0  
5  
-1  
-1

var num = 5;function getOpposite(num) {}var result = getOpposite(num)

Solution:

let getOppposite = (*value*) => {

    if (*value* === 0)

        return 0

    if (Number.isInteger(*value*)) {

        return (-*value*)

    } else {

        return -1

    }

}

console.log("Opposite of 5 is: " + getOppposite(5))

console.log("Opposite of 0 is: " + getOppposite(0))

console.log("Opposite of -5 is: " + getOppposite(-5))

console.log("Opposite of 5a is: " + getOppposite('5a'))

console.log("Opposite of 5.5 is: " + getOppposite(5.5))

3) **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

var min = 5;function toSeconds(min) {}var secs = toSeconds(min)

Solution:

var min = 5;

function toSeconds(*min*) {

    return *min*\*60

}

var secs = toSeconds(min)

console.log(secs);

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

Examples  
toInteger(“6”) ➞ 6

toInteger(“1000”) ➞ 1000

toInteger(“12”) ➞ 12

var mystr = "5";function toInteger(mystr) {}var myint = toInteger(mystr)

Solution:

var mystr = "5";

function toInteger(*mystr*) {

    return parseInt(*mystr*);

}var myint = toInteger(mystr)

console.log(myint);

5) **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

var myint = 0;function nextNumber(myint) {}var myNextint = nextNumber(myint)

Solution:

var myint = 0;function nextNumber(*myint*) {

    return *myint*+1;

}var myNextint = nextNumber(myint);

console.log(myNextint);

6) **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) {}var data = getFirstElement(arr)

Solution:

var arr = [1, 2, 3];function getFirstElement(*arr*) {

    return *arr*[0]

}var data = getFirstElement(arr)

console.log(data);

7) **Problem**

Convert Hours into Seconds

Write a function that converts hours into seconds.

Examples  
hourToSeconds(2) ➞ 7200

hourToSeconds(10) ➞ 36000

hourToSeconds(24) ➞ 86400

var arr = [1, 2, 3];function hourToSeconds(arr) {}var data = hourToSeconds(arr)

Solution:

var hour=2;

function hourToSeconds(*hour*) {

return *hour*\*60\*60

}

var data = hourToSeconds(hour)

console.log(data)

8) 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

function findPerimeter(num1,num2) {}var peri = findPerimeter(6,7)

Solution:

function findPerimeter(*num1*,*num2*) {

    return 2\*(*num1*+*num2*);

}var peri = findPerimeter(6,7);

console.log(peri);

9) **Problem**

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

Examples  
lessThan100(22, 15) ➞ true  
// 22 + 15 = 37

lessThan100(83, 34) ➞ false  
// 83 + 34 = 117

function lessThan100(num1,num2) {}var res = lessThan100(22,15)

Solution:

function lessThan100(*num1*,*num2*) {

    if(*num1*+*num2* <100)

    {

        return true;

    }

    else

    return false;

}

var res = lessThan100(22,15)

console.log(res)

10) **Problem**

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

function remainder(num1,num2) {}var res = remainder(1,3)

Solution:

function remainder(*num1*,*num2*) {

    return *num1*%*num2*

}var res = remainder(-9,45)

console.log(res)

11) **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) {}var legs = CountAnimals(2,3,5)

Solution:

function CountAnimals(*tur*,*horse*,*pigs*) {

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

}

var legs = CountAnimals(2,3,5)

console.log(legs)

12) **Problem**

Frames Per Second  
Create a function that returns the number of frames shown in a given number of minutes for a certain FPS.

Examples  
frames(1, 1) ➞ 60

frames(10, 1) ➞ 600

frames(10, 25) ➞ 15000

function frames(num1,num2) {}var fps = frames(1,2)

Solution:

function frames(*num1*,*num2*) {

    return (*num1*\*10)\*(*num2*\*6)

}

var fps = frames(1,1)

console.log(fps)

13) **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) {}var divisible = divisibleByFive(5)

Solution:

function divisibleByFive(*num1*) {

    if(*num1*%5 ===0)

    {

        return true;

    }

    else return false;

}var divisible = divisibleByFive(5)

console.log(divisible);

14) **Problem**:

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

Output:

true  
true  
false  
-1

function isEven(num){  
 // your code here  
}var even = isEven(5)

Solution:

function isEven(*num*){

   if(Number.isInteger(*num*))

   { if(*num* %2 ===0)

    {

        return true

    }

    else

     return false

   }

   else return -1

   }var even = isEven('1a')

   console.log(even)

15) **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);

Output:

true  
flase  
flase  
flase

function areBothOdd(num1, num2){  
 // your code here  
}

Solution:

function areBothOdd(*num1*, *num2*){

    if((*num1*+*num2*)%2===0)

    {

        return true

    }

    else return false

   }

var odd= areBothOdd(1,4);

console.log(odd);

16) **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(“”, “”);

Output:

“GUVI GEEK”  
“GUVI”  
“GEEK”  
“”

function getFullName(firstName, lastName){  
 // your code here  
}

Solution:

function getFullName(*firstName*, *lastName*){

return *firstName* + " " + *lastName*;

}

var out=getFullName("GUVI","GEEK");

console.log(out);

17) **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

function getLengthOfWord(word1){  
 // your code here  
}

Solution:

function getLengthOfWord(*word1*){

    var len=*word1*.length;

    if(length >= 0)

    {

    return len;

    }

    else

    return -1

*// your code here*

   }

   var out=getLengthOfWord("")

   console.log(out);

18) **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

function isSameLength(word1, word2){  
 // your code here  
}

Solution:

function isSameLength(*word1*, *word2*){

    if(*word1*.length === *word2*.length)

    {

        return true

    }

    else return false

      }

      var out=isSameLength("GUVI","GEEK");

      console.log(out);

19)

**Problem**:

Create a function to calculate the distance between two points defined by their x, y coordinates

console.log(getDistance(100, 100, 400, 300));function getDistance(x1, y1, x2, y2)  
{  
   
}

Solution:

function getDistance(*x1*, *y1*, *x2*, *y2*)

{

    return Math.sqrt(Math.pow((*x2*-*x1*),2)+Math.pow((*y2*-*y1*),2));

}

var out= getDistance(100,100,400,300)

console.log(out)

20) **Problem**:

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){  
 // your code here  
}

Solution:

function getNthElement(*array*,*n*){

    return *array*[*n*];

   }

   var out=getNthElement([1,3,5],1)

   console.log(out)

21) **Problem**:

Write a function called “getLastElement”.  
Given an array, “getLastElement” returns the last element of the given array. If the given array has a length of 0, it should return ‘-1’.  
Input:  
getLastElement([1, 2, 3, 4]);  
Output:  
4

function getLastElement(array){  
 // your code here  
}

Solution:

function getLastElement(*array*){

   return *array*[*array*.length-1];

   }

   console.log(getLastElement([1,2,3,4]))

22) **Problem**:

Write a function called “getProperty”.  
Given an object and a key, “getProperty” returns the value of the property at the given key. If there is no property at the given key, it should return undefined.  
  
var obj = {  
mykey: “value”  
};

Input:  
getProperty(obj,’mykey’);  
getProperty(obj,’dummykey’);  
Output:  
value  
NA

var obj = {  
 mykey: “value”  
};function getProperty(obj, key) {  
 // your code here  
}

Solution:

var obj = {

    mykey: "value"

   };function getProperty(*obj*, *key*) {

    return *obj*.mykey;

   }

   console.log(getProperty(obj,'mykey'))

23) **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  
}

var obj = {  
 mykey: “value”  
};function addProperty(obj, key){  
 // your code here}

Solution:

var myObj = {

    mykey: "value"

};

function addProperty(*obj*, *key*) {

*obj*[*key*]=true;

    console.log(*obj*);

    return *obj*.key;

  }

addProperty(myObj, 'mykey');

console.log(myObj.mykey);

24) **Problem**:

Write a function called “removeProperty”.  
Given an object and a key, “removeProperty” removes the given key from the given object.  
Input:  
removeProperty(obj, “name”);  
Output:  
undefined

function removeProperty(obj, key){  
 // your code here  
}

Solution:

var myObj = {

    mykey: "value",

    name: "abc"

};

function removeProperty(*obj*, *key*){

 delete *obj*[*key*];

return myObj;

   }

   removeProperty(myObj,"mykey");

   console.log(myObj);

25) **Problem**:

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

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

Solution:

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

let ar2=function countPositivesSumNegatives(*arr*) {

    let count = 0;

    let ncount =0;

    for(let i=0;i<*arr*.length;i++)

    {

    if(*arr*[i]>0)

    {

        count +=*arr*[i];

    }

    else

     ncount+=*arr*[i];

}

let sum = [count,ncount]

return sum

}

console.log(ar2(arr));

26) **Problem**:

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

function getPositives(ar)  
{  
 // your code here  
}var ar = [-5, 10, -3, 12, -9, 5, 90, 0, 1];  
var ar2 = getPositives(ar);console.log(ar2);

Solution:

function getPositives(*ar*)

{

    let count=0

    for(let i=0;i<*ar*.length;i++)

    {

    if(*ar*[i]>0)

    {

        count +=*ar*[i];

    }

}

return count;

}

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

var ar2 = getPositives(ar);

console.log(ar2);

27) **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

function powersOfTwo(n){  
 return res;  
}

Solution:

function powersOfTwo(*n*){

    let res =[];

    for(let i=0;i<=*n*;i++)

    {res.push( Math.pow(2,i));

    }

    return res;

}

console.log(powersOfTwo(2))

28) **Problem**:

Find the maximum number in an array of numbers

function findMax(ar)  
{  
// your code here  
}var ar = [-5, 10, -3, 12, -9, 5, 90, 0, 1];  
var max = findMax(ar);  
console.log(“Max: “, max);

Solution:

function findMax(*ar*)

{

let ar2=*ar*.sort(function(*a*,*b*){return *a*-*b*})

return ar2[ar2.length-1];

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

var max = findMax(ar);

console.log("Max: ", max);

29) **Problem**:

Print the first 100 prime numbers

printPrimes(100);// Function prints the first nPrimes numbers  
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)  
{  
 // your code here  
}

Solution:

printPrimes(100);

function printPrimes(*nPrimes*)

{

 var n = 0;

 var i = 2;

 while(n < *nPrimes*)

 {

 if (isPrime(i))

 {

 console.log(n," → ", i);

 n++;

 }

 i++;

 }

}

function isPrime(*n*)

{

    for(let i=2;i<*n*;i++)

            {

                if(*n*% i === 0)

                {

                return false;

                }

            }

            return true;

        }

30) **Problem**:

Create a function that will return in an array the first “nPrimes” prime numbers greater than a particular number “startAt”

console.log(getPrimes(10, 100));function getPrimes(nPrimes, startAt)  
{// your code here  
 isPrime(i)  
}// Returns true if a number is prime  
function isPrime(n)  
{  
 // your code here  
}

Solution:

console.log(getPrimes(10, 100));

function getPrimes(*nPrimes*,*startAt*)

{

 var n = 0;

 var i = *startAt*;

for(i=*startAt*;n<*nPrimes*;)

 {

 if (isPrime(i))

 {

 console.log(n," → ", i);

 n++;

 }

 i++;

 }

}

function isPrime(*n*)

{

    for(let i=2;i<*n*;i++)

            {

                if(*n*% i === 0)

                {

                return false;

                }

            }

            return true;

        }

31) **Problem**:

Reverse a string

var s = reverseString("JavaScript");  
console.log(s);function reverseString(s)  
{  
 // your code here   
}

Solution:

var s = reverseString("JavaScript");

console.log(s);

function reverseString(*s*)

{

   return *s*.split('').reverse().join('');

}

32) **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 = [];//this will add the first array to the result array  
for(let el of ar1)  
 {  
 result.push(el);  
 }  
   
 //Some piece of code goes here   
   
 return result;  
}

Solution:

var ar1 = [1, 2, 3];

var ar2 = [4, 5, 6];

var ar = mergeArrays(ar1, ar2);

console.log(ar);function mergeArrays(*ar1*, *ar2*)

{

 var result = [];*//this will add the first array to the result array*

for(let el of *ar1*)

 {

 result.push(el);

 }

 for(let el of *ar2*)

 {

 result.push(el);

 }

 return result;

}

33) **Problem**:

Calculate the sum of numbers received in a comma delimited string

console.log(sumCSV(“1.5, 2.3, 3.1, 4, 5.5, 6, 7, 8, 9, 10.9”));function sumCSV(s)  
{  
 // your code here  
}

Solution:

console.log(sumCSV("1.5, 2.3, 3.1, 4, 5.5, 6, 7, 8, 9, 10.9"));

function sumCSV(*s*)

{

   var array = *s*.split(',');

   var sum =0;

   for(let i=0;i<array.length;i++)

   {

       sum = sum+ + (array[i])

   }

   return sum;

}