GUVI : Zen Code-Sprints :— JavaScript Functions — Warmup 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)

Correct code :

var num = 10;

function addFive(num) {

  return num + 5;

}

var result = addFive(-5);

console.log(result); // Output is 0

1. **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)

Correct code :

let num = 5;

function getOpposite(num) {

    return isNaN(num) || num % 1 !== 0 ? -1 : num - (num\*2);

}

let result = getOpposite(5);

console.log(result); // Output is -5

1. **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)

correct code :

let min = 5;

function toSeconds(min) {

    return min \* 60;

}

let secs = toSeconds(5);

console.log(secs); // Output is 300

1. **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)

Correct code :

var mystr = "5";

function toInteger(mystr) {

  return +mystr;

}

var myint = toInteger("6");

console.log(myint); // Output is 6

1. **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)

Correct code :

let myint = 0;

function nextNumber(myint) {

    return ++myint;

}

let myNextint = nextNumber(-3);

console.log(myNextint); // Output is -2

1. **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)

Correct code :

let arr = [1, 2, 3];

function getFirstElement(arr) {

 return arr[0];

}

let data = getFirstElement([-500, 0, 50]);

console.log(data); // Output is -500

1. **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)

Correct Code :

let arr = [1, 2, 3];

function hourToSeconds(hour) {

    return hour\*(60\*60);

}

let data = hourToSeconds(24);

console.log(data); // Output is 86400

1. **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

function findPerimeter(num1,num2) {

}

var peri = findPerimeter(6,7)

Correct code :

function findPerimeter(num1, num2) {

  return 2 \* (num1 + num2);

}

let peri = findPerimeter(6, 7);

console.log(peri); // Output is 26

1. **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)

Correct code :

function lessThan100(num1, num2) {

  return num1 + num2 < 100;

}

let res = lessThan100(22, 15); // returns true

// let res = lessThan100(90, 15); // returns false

console.log(res); // Output is true

1. **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)

Correct code :

function remainder(num1, num2) {

    return num1 % num2;

}

let res = remainder(1, 3);

console.log(res); // Output is 1

1. **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)

Correct code :

function CountAnimals(tur, horse, pigs) {

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

}

let legs = CountAnimals(2, 3, 5);

console.log(legs); // Output is 36

1. **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)

Correct code :

function frames(num1, num2) {

  return num1 \* (num2 \* 60);

}

let fps = frames(1, 2);

console.log(fps); // Output is 120

1. **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)

Correct code :

function divisibleByFive(num1) {

    return num1 % 5 === 0;

}

let divisible = divisibleByFive(5);

console.log(divisible); // Output is true

1. **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)

Correct code :

function isEven(num) {

  return isNaN(num) ? -1 : num % 2 === 0;

}

let even = isEven(5);

console.log(even); // Output is false

1. **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  
}

Correct code :

function areBothOdd(num1, num2) {

  return num1 % 2 !== 0 && num2 % 2 !== 0;

}

console.log(areBothOdd(1, 3)); // Output is true

console.log(areBothOdd(1, 4)); // Output is false

1. **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  
}

Correct code :

function getFullName(firstName, lastName) {

  return firstName + " " + lastName;

}

console.log("GUVI", "GEEK"); // Output is “GUVI GEEK”

1. **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  
}

Correct code :

function getLengthOfWord(word1) {

  return typeof word1 === 'string' ? word1.length : -1;

}

console.log(getLengthOfWord("GUVI")); // Output is 4

console.log(getLengthOfWord("")); // Output is 0

console.log(getLengthOfWord()); // Output is -1

console.log(getLengthOfWord(9)); // Output is -1

1. **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  
}

Correct code :

function isSameLength(word1, word2) {

    return word1.length === word2.length;

}

console.log(isSameLength("GUVI", "GEEK")); // Output is true

1. **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)  
{  
   
}

Correct code :

console.log(getDistance(100, 100, 400, 300));

// Output is 360.56

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

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

}

1. **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  
}

Correct code :

function getNthElement(array, n) {

  return array[n];

}

console.log(getNthElement([1, 3, 5], 1)); // Output is 3

1. **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  
}

Correct code :

function getLastElement(array) {

    return array[0] ? array[array.length - 1] : -1;

}

console.log(getLastElement([1, 2, 3, 4])); // Output is 4

1. **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  
}

Correct code :

var obj = {

  mykey: "value",

};

function getProperty(obj, key) {

    return obj[key];

}

console.log(getProperty(obj, "mykey")); // Output is value

console.log(getProperty(obj, "DummyKey")); // undefined

1. **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}

Correct code :

var obj = {};

function addProperty(obj, key) {

    obj[key] = true;

}

addProperty(obj, "newkey");

console.log(obj); // Output is {newkey: true}

1. **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  
}

Correct code :

let obj = {

  fName: "Michael",

  lName: "Xavier",

};

function removeProperty(obj, key) {

    delete obj[key];

}

removeProperty(obj, "lName");

console.log(obj); // Output is {fName: "Michael"}

1. **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);

correct code :

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

let ar2 = function countPositivesSumNegatives(arr) {

  let ans = [0, 0];

  arr.map((num) => {

    num < 0 ? (ans[1] = ans[1] + num) : (ans[0] = ans[0] + 1);

  });

  return ans;

};

console.log(ar2(arr)); // Output is [6, -17]

1. **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);

correct code :

function getPositives(ar) {

  let positives = arr.filter((num) => num >= 0);

  return positives;

}

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

var ar2 = getPositives(arr);

console.log(ar2); // Output is [10, 12, 5, 90, 0, 1]

1. **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;  
}

Correct code :

function powersOfTwo(n) {

  let res = [];

  for (let i = 0; i <= n; i++) {

    res.push(Math.pow(2, i));

  }

  return res;

}

console.log(powersOfTwo(0)); // Output is [1]

console.log(powersOfTwo(1)); // Output is [1, 2]

console.log(powersOfTwo(2)); // Output is [1, 2, 4]

1. **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);

Correct code :

function findMax(ar) {

  return Math.max(...ar);

}

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

var max = findMax(ar);

console.log("Max: ", max); // Output is Max: 90

1. **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  
}

Correct code :

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++;

  }

}

// code continues in next page

// Returns true if a number is prime

function isPrime(n) {

  let factors = [];

  for (let i = 1; i <= n; i++) {

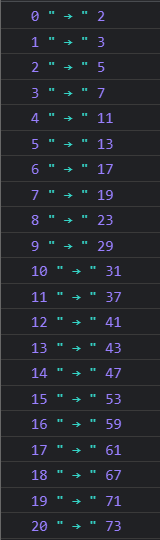
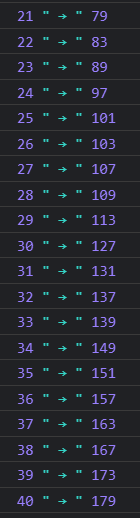
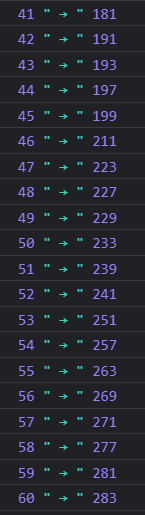
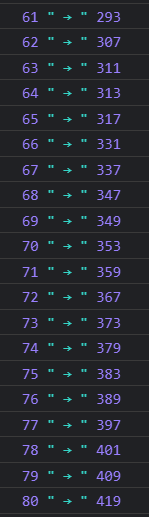
    if (n % i === 0) factors.push(i);

  }

  return factors.length === 2;

}

Output :

1. **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  
}

Correct code :

console.log(getPrimes(10, 100));

function getPrimes(nPrimes, startAt) {

  let primeNumbers = [];

  let i = startAt;

  while (primeNumbers.length < nPrimes) {

    if (isPrime(i)) primeNumbers.push(i);

    i++;

  }

  return primeNumbers;

}

// Returns true if a number is prime

function isPrime(n) {

  let factors = [];

  for (let i = 1; i <= n; i++) {

    if (n % i === 0) factors.push(i);

  }

  return factors.length === 2;

}

Output : **[101, 103, 107, 109, 113, 127, 131, 137, 139, 149]**

1. **Problem**:

Reverse a string

var s = reverseString("JavaScript");

console.log(s);

function reverseString(s)  
{  
 // your code here   
}

Correct code :

var s = reverseString("JavaScript");

console.log(s);

function reverseString(s) {

    // using built-in functions

    // return s.split("").reverse().join("");

    let reversed = "";

    s.split("").map((letter) => {

        reversed = letter + reversed;

    })

    return reversed;

}

// Output is tpircSavaJ

1. **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;  
}

Correct code :

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;

}

// Output is [1, 2, 3, 4, 5, 6]

1. **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  
}

Correct code :

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

function sumCSV(s) {

  let sum = 0;

  s.split(", ").forEach((num) => {

    sum += +num;

  });

  return sum;

}

// Output is 57.3