

Ejercicios de JavaScript.

- Escribe una función que reciba dos números como parámetros y devuelva su suma.

Solución:

```
function sumar(a, b) {  
  return a + b;  
}
```

```
console.log(sumar(3, 5)); // Output: 8
```

- Escribe una función que reciba un número como parámetro y devuelva true si es par, false si es impar.

Solución:

```
function esPar(num) {  
  return num % 2 === 0;  
}
```

```
console.log(esPar(4)); // Output: true
```

- Escribe una función que reciba un arreglo de números y devuelva la suma de todos los elementos.

Solución:

```
function sumaArreglo(arr) {  
  return arr.reduce((total, num) => total + num, 0);  
}
```

```
console.log(sumaArreglo([1, 2, 3, 4])); // Output: 10
```

- Escribe una función que reciba un arreglo de números y devuelva un nuevo arreglo con solo los números pares.

Solución:

```
function filtrarPares(arr) {  
  return arr.filter(num => num % 2 === 0);  
}
```

```
console.log(filtrarPares([1, 2, 3, 4])); // Output: [2, 4]
```

- Escribe una función que reciba un arreglo de strings y devuelva un nuevo arreglo con la longitud de cada string.

Solución:

```
function longitudStrings(arr) {  
  return arr.map(str => str.length);  
}
```

```
}
```

```
console.log(longitudStrings(["hola", "adios", "buenos días"])); //  
Output: [4, 5, 11]
```

- Escribe una función que reciba un string y devuelva el mismo string pero con todas las letras en mayúsculas.

Solución:

```
function convertirMayusculas(str) {  
  return str.toUpperCase();  
}
```

```
console.log(convertirMayusculas("hola mundo")); // Output: "HOLA  
MUNDO"
```

- Escribe una función que reciba un string y devuelva true si es un palíndromo, false si no lo es.

Solución:

```
function esPalindromo(str) {  
  const reversa = str.split('').reverse().join('');  
  return str === reversa;  
}
```

```
console.log(esPalindromo("anita lava la tina")); // Output: true
```

- Escribe una función que reciba un número y devuelva su factorial.

Solución:

```
function factorial(num) {  
  if (num === 0 || num === 1) {  
    return 1;  
  } else {  
    return num * factorial(num - 1);  
  }  
}
```

```
console.log(factorial(5)); // Output: 120
```

- Escribe una función que reciba un número y devuelva true si es primo, false si no lo es.

Solución:

```
function esPrimo(num) {  
  if (num <= 1) {  
    return false;  
  }  
  for (let i = 2; i <= Math.sqrt(num); i++) {  
    if (num % i === 0) {  
      return false;  
    }  
  }  
  return true;  
}
```

```
    }  
  }  
  return true;  
}  
  
console.log(esPrimo(7)); // Output: true
```

- Escribe una función que reciba un número y devuelva un arreglo con todos los números primos hasta ese número.

Solución:

```
• function primosHasta(num) {  
  const primos = [];  
  for (let i = 2; i <= num; i++) {  
    if (esPrimo(i)) {  
      primos.push(i);  
    }  
  }  
  return primos;  
}  
  
console.log(primosHasta(20)); // Output: [2, 3, 5, 7, 11, 13, 17, 19]
```

- Escribe una función que reciba un número y devuelva la suma de todos los números naturales hasta ese número.

Solución:

```
• function sumaNaturalesHasta(num) {  
  let suma = 0;  
  for (let i = 1; i <= num; i++) {  
    suma += i;  
  }  
  return suma;  
}  
  
console.log(sumaNaturalesHasta(5)); // Output: 15
```

- Escribe una función que reciba un arreglo de números y devuelva el número más grande.

Solución:

```
• function maximo(arr) {  
  return Math.max(...arr);  
}  
  
console.log(maximo([3, 5, 1, 9, 2])); // Output: 9
```

- Escribe una función que reciba un arreglo de números y devuelva el número más pequeño.

Solución:

- ```
function minimo(arr) {
 return Math.min(...arr);
}
```

```
console.log(minimo([3, 5, 1, 9, 2])); // Output: 1
```

- Escribe una función que reciba un arreglo de números y devuelva la suma de los números pares.

#### **Solución:**

- ```
function sumaPares(arr) {  
    return arr.reduce((total, num) => num % 2 === 0 ? total + num :  
    total, 0);  
}
```

```
console.log(sumaPares([1, 2, 3, 4, 5, 6])); // Output: 12
```

- Escribe una función que reciba un arreglo de números y devuelva la suma de los números impares.

Solución:

- ```
function sumaImpares(arr) {
 return arr.reduce((total, num) => num % 2 !== 0 ? total + num :
 total, 0);
}
```

```
console.log(sumaImpares([1, 2, 3, 4, 5, 6])); // Output: 9
```

- Escribe una función que reciba dos arreglos y devuelva un nuevo arreglo que contenga los elementos comunes entre ambos.

#### **Solución:**

- ```
function elementosComunes(arr1, arr2) {  
    return arr1.filter(num => arr2.includes(num));  
}
```

```
console.log(elementosComunes([1, 2, 3], [3, 4, 5])); // Output: [3]
```

- Escribe una función que reciba un arreglo de números y devuelva un nuevo arreglo con los elementos en orden inverso.

Solución:

- ```
function invertirArreglo(arr) {
 return arr.reverse();
}
```

```
console.log(invertirArreglo([1, 2, 3, 4])); // Output: [4, 3, 2, 1]
```

- Escribe una función que reciba un número y devuelva su representación en binario.

### Solución:

```
• function decimalABinario(num) {
 return num.toString(2);
}

console.log(decimalABinario(10)); // Output: "1010"
```

- Escribe una función que reciba un arreglo de strings y devuelva un nuevo arreglo con las mismas palabras pero en orden alfabético.

### Solución:

```
• function ordenarPalabras(arr) {
 return arr.sort();
}

console.log(ordenarPalabras(["perro", "gato", "pez"])); // Output:
["gato", "pez", "perro"]
```

- Escribe una función que reciba un arreglo de objetos y devuelva un nuevo arreglo con solo los objetos que tienen una propiedad específica.

### Solución:

```
• function filtrarPorPropiedad(arr, propiedad) {
 return arr.filter(objeto => objeto.hasOwnProperty(propiedad));
}

const personas = [
 { nombre: 'Juan', edad: 25 },
 { nombre: 'María', telefono: '123456789' },
 { nombre: 'Pedro', edad: 20 }
];

console.log(filtrarPorPropiedad(personas, 'edad')); // Output: [{
nombre: 'Juan', edad: 25 }, { nombre: 'Pedro', edad: 20 }]
```

- Escribe una función que reciba un arreglo de números y devuelva un nuevo arreglo con los números duplicados eliminados.

### Solución:

```
• function eliminarDuplicados(arr) {
 return Array.from(new Set(arr));
}

console.log(eliminarDuplicados([1, 2, 2, 3, 4, 4, 5])); // Output: [1,
2, 3, 4, 5]
```

- Escribe una función que reciba un arreglo de números y devuelva la diferencia entre el número más grande y el más pequeño.

### Solución:

- ```
function diferenciaMaxMin(arr) {  
    return Math.max(...arr) - Math.min(...arr);  
}
```

```
console.log(diferenciaMaxMin([10, 4, 1, 8, 20])); // Output: 19
```

- Escribe una función que reciba un arreglo de números y devuelva un nuevo arreglo con los números en orden ascendente.

Solución:

- ```
function ordenarAscendente(arr) {
 return arr.sort((a, b) => a - b);
}
```

```
console.log(ordenarAscendente([5, 2, 8, 1, 3])); // Output: [1, 2, 3,
5, 8]
```

- Escribe una función que reciba un string y devuelva el número de vocales que contiene.

### **Solución:**

- ```
function contarVocales(str) {  
    const vocales = ['a', 'e', 'i', 'o', 'u'];  
    return str.split('').filter(letra =>  
        vocales.includes(letra.toLowerCase())).length;  
}
```

```
console.log(contarVocales("Murciélago")); // Output: 4
```

- Escribe una función que reciba un arreglo de números y un número objetivo, y devuelva true si hay un par de números en el arreglo que sumen ese número objetivo.

Solución:

```
function sumaParObjetivo(arr, objetivo) {  
    const set = new Set();  
    for (let num of arr) {  
        if (set.has(objetivo - num)) {  
            return true;  
        }  
        set.add(num);  
    }  
    return false;  
}
```

```
console.log(sumaParObjetivo([1, 2, 3, 4], 7)); // Output: true
```