

EJERCICIOS JAVASCRIPT

- Crea una función que tome como parámetro un número. Si es un número entero (sin decimales) muestra por consola true. Si no, muestra por consola false.
- Crea una función que tome como parámetros dos strings, puedes llamarlos a y b. Si a contiene b, añade b al principio de a. Si no, añade b al final. Por ejemplo:

("cheese", "cake") => output: "cheesecake"

("Java", "script") => output: "JavaScript"

("therefore I am", "I think") => output: "I think, therefore I am"

- Crea una función que tome como parámetro un número y lo divida en dígitos individuales dentro de un array. Tip: puedes usar el método split(); Por ejemplo:

(10) => output: [1,0]

- Crea una función que tome como parámetro un array. Si todos los elementos del array son estrictamente iguales, muestra por consola true, de lo contrario muestra false. Por ejemplo:

([true, true, true, true]) => output: true

(['10',10,10,10]) => output: false

- Dado un array de números, muestra por consola un array con dichos números ordenados en forma descendente.
- Dado un array de strings, muestra por consola un array con sus valores ordenados alfabéticamente. Por ejemplo:

['b', 'c', 'd', 'a'] => output: ['a', 'b', 'c', 'd']

- Crea una función que tome como parámetro un array de strings y muestra por consola el string más largo. Por ejemplo:

(['help', 'me']) => output: "help"

- Crea una función que tome dos arrays de números como parámetros. Une ambos arrays, remueve valores duplicados (si hay alguno) y ordena los números en orden ascendente.
- Crea una función que tome un objeto como parámetro. Remueve la propiedad "country" y su valor y muestra el resultado por consola. Por ejemplo:

```
const myObj = { city: 'Sacramento', state: 'California', country: 'USA', continent: 'North America' }
```

=> output: { city: 'Sacramento', state: 'California', continent: 'North America' }

- Trabajando con el mismo array anterior, muestra por consola el valor de la propiedad "city"

- Algunas de las propiedades de los siguientes objetos son strings vacíos. Reemplázalos por el valor “null”, empleando para ello una función que tome como parámetro dichos objetos. Muestra por consola el objeto resultante

```
{ a: 'a', b: 'b', c: '' }
```

```
{ a: '', b: 'b', c: '', d: 'd' }
```

```
{ a: 'a', b: 'b ', c: ' ', d: '' }
```

=> output: { a: 'a', b: 'b', c: null }

- Crea una función que tome como parámetros un array de objetos y un string. Agrega a cada uno de los objetos la propiedad “continent” que tome como valor el parámetro de string. Muestra por consola el nuevo array.

```
([{ city: 'Tokyo', country: 'Japan' }, { city: 'Bangkok', country: 'Thailand' }], 'Asia')
```

=> output: [{ city: 'Tokyo', country: 'Japan', continent: 'Asia' }, { city: 'Bangkok', country: 'Thailand', continent: 'Asia' }]

```
([{ city: 'Stockholm', country: 'Sweden' }, { city: 'Paris', country: 'France' }], 'Europe')
```

=> output: [{ city: 'Stockholm', country: 'Sweden', continent: 'Europe' }, { city: 'Paris', country: 'France', continent: 'Europe' }]

- Dado el siguiente HTML, crea una función que, a través de la manipulación del DOM, cambie el color de fondo de los párrafos, cada uno con colores diferentes.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>Set the background color of a paragraph</title>
</head>
<body>
<input type="button" value="Click to set paragraph background
color" onclick="setBackground()">
<p>Paragraph 1</p>
<p>Paragraph 2</p>
</body>
</html>
```

- Dado el siguiente array de objetos, manipula el DOM para imprimir en el html como un h1 los valores de cada una de las propiedades del objeto. Tip: debes hacer uso de appendChild().

```
const dogs = [  
  { name: "balto", type: "chihuahua" },  
  { name: "loli", type: "puddle" },  
];
```

output:

balto es un chihuahua

loli es un puddle