

Introducción

JavaScript es un lenguaje de programación interpretado (El navegador lee directamente el código, sin necesidad de terceros) de alto nivel. Es el lenguaje de programación encargado de dotar de mayor interactividad y dinamismo a las páginas web.



¿Cuáles son los principales objetivos de estos ejercicios?

- Repasar todo lo visto de JS hasta ahora.

Ejercicios Condiciones y Objetos/Arrays

- Crea un objeto en la variable “ordenador” de tal modo que el siguiente código no muestre por consola 'suspendedes':

```
// TEST 1

if (typeof ordenador === "object") {

    console.log("apruebas");

} else {

    console.log("suspendedes");

}


// TEST 2

if (ordenador.marca === "Asus") {

    console.log("apruebas");

} else {

    console.log("suspendedes");

}


// TEST 3

if (ordenador.tipo === "portátil") {

    console.log("apruebas");

} else {

    console.log("suspendedes");

}
```

```
}

// TEST 4

if (ordenador.perifericos.touchpad === true) {

    console.log("apruebas");

} else {

    console.log("suspendes");

}


// TEST 5

if (ordenador.almacenamiento.discos[0] === "SSD") {

    console.log("apruebas");

} else {

    console.log("suspendes");

}


// TEST 6

if (ordenador.procesador.velocidad === "2.5 GHz") {

    console.log("apruebas");

} else {

    console.log("suspendes");

}
```

Ejercicios Bucles

- Imprimir los números del 21 al 34 en la consola.
- Sumar los números del 1 al 10 y mostrar el resultado en la consola.
- Dado un array de números, imprimir en la consola la suma de todos los números.
- Dado un número, encontrar su factorial.
- Dado un array de números, imprimir en la consola solo los números impares.
- Dado un array de objetos con propiedades "nombre" y "edad", imprimir en la consola solo los nombres de las personas mayores de 18 años.
- Dado un número, imprimir en la consola si es primo o no.
- Utiliza la siguiente array para resolver los próximos ejercicios:

```
let animales = [  
  {  
    nombre: "León",  
    especie: "Felino",  
    edad: 6,  
  },  
  {  
    nombre: "Cebra",  
    especie: "Herbívoro",  
    edad: 4,  
  },  
  {  
    nombre: "Tigre",  
    especie: "Felino",  
    edad: 3,  
  },  
]
```

```
    },  
    {  
      nombre: "Elefante",  
      especie: "Herbívoro",  
      edad: 8,  
    },  
    {  
      nombre: "Jirafa",  
      especie: "Herbívoro",  
      edad: 5,  
    },  
    {  
      nombre: "Oso",  
      especie: "Omnívoro",  
      edad: 2,  
    },  
    {  
      nombre: "Pingüino",  
      especie: "Ave",  
      edad: 1,  
    },  
  ],  
];
```

- Crea un array con los animales que son felinos y mayores de 4 años. No utilices el método filter de las arrays y muéstralo por consola.
- Crea un array con los animales que son herbívoros y tienen una edad par. No utilices el método filter de las arrays y muéstralo por consola.

- Crea un array con los animales que son omnívoros o aves y menores de 3 años. No utilices el método filter de las arrays y muéstralo por consola.
- Crea un array con los animales que su especie empieza por "H". No utilices el método filter de las arrays y muéstralo por consola.
- Crea un array con los animales cuyo nombre tenga más de 4 letras y su especie sea felino o herbívoro. No utilices el método filter de las arrays y muéstralo por consola.

Ejercicios Funciones

- Crea una función llamada eliminarDuplicados(array) que reciba como parámetro un array de elementos y devuelva un nuevo array con los elementos únicos, es decir, sin elementos duplicados. Puedes utilizar un ciclo for...of para recorrer el array original.

```
//salida
eliminarDuplicados([1, 2, 3, 3, 4, 5, 5, 6]); //
Devuelve [1, 2, 3, 4, 5, 6]
eliminarDuplicados(["hola", "adios", "hola",
"hasta", "luego"]); // Devuelve ["hola", "adios",
"hasta", "luego"]
```

- Crea una función llamada sumarElementos(array) que reciba como parámetro un array de números y devuelva la suma de todos sus elementos. Puedes utilizar un ciclo for...of para recorrer el array y sumar sus elementos.

```
sumarElementos([1, 2, 3, 4, 5]); // Devuelve 15
sumarElementos([-5, 10, -15, 20]); // Devuelve 10
```

- Crea una función en contarCaracteres(array) que reciba como parámetro un array de cadenas de caracteres y devuelva la cantidad total de caracteres que hay en todas las cadenas juntas. Puedes utilizar un ciclo for...of para recorrer el array y contar los caracteres.

```
contarCaracteres(["Hola", "Mundo"]); // Devuelve 9  
contarCaracteres(["JavaScript", "es",  
"genial"]); // Devuelve 18
```

- Crea una función llamada numeroMasGrande(array) que reciba como parámetro un array de números y devuelva el número más grande de ese array. Puedes utilizar un ciclo for...of para recorrer el array y condicionales para encontrar el número más grande.

```
numeroMasGrande([1, 2, 3, 4, 5]); // Devuelve 5  
numeroMasGrande([10, -5, 20, -15]); // Devuelve 20
```

- Crea una función llamada cadenaMasLarga(array) que reciba como parámetro un array de cadenas de caracteres y devuelva la cadena más larga de ese array. Puedes utilizar un ciclo for...of para recorrer el array y condicionales para encontrar la cadena más larga.

```
cadenaMasLarga(["Hola", "Mundo"]); // Devuelve  
"Mundo"  
cadenaMasLarga(["JavaScript", "es",  
"genial"]); // Devuelve "JavaScript"
```

Ejercicio Suscripción a Newsletter

Crea una vista "Suscripción a Newsletter" que contenga un formulario con los siguientes campos: Nombre, Dirección de correo electrónico, Intereses (checkboxes) y Tipo de suscripción (radio buttons). El objetivo de este formulario es recoger la información del usuario y guardarla en localStorage.

Para garantizar que se recopilan todos los datos necesarios, se debe implementar una validación que obligue al usuario a completar todos los campos del formulario. También es importante realizar una validación de la dirección de correo electrónico, para asegurarse de que se ha introducido una dirección válida.

Si alguna de las validaciones no se cumple, se debe mostrar un mensaje de error durante 3 segundos y luego desaparecer. Si el usuario completa correctamente todos los campos del formulario, se debe mostrar un mensaje de éxito durante 3 segundos y redirigir a la vista de confirmación de suscripción. Para mostrar los mensajes, se pueden utilizar los alertas de Bootstrap.

Ejercicio mostrar Newsletter

Crea una vista Suscripciones y muestra en esa página la suscripción que habías guardado previamente en el localStorage

Ejercicio Newsletter nivel 2

Modifica tu formulario de Suscripción para que puedas guardar varias suscripciones. Cada vez que escribas una nueva suscripción, guárdala en un array junto con las anteriores.

Después, muestra todas las suscripciones en la vista de Suscripciones, de forma que se puedan leer todas ellas.

Entregables

- Subir el ejercicio a un repositorio en github
- Dejar el enlace del repositorio adjuntado en Classroom