

# ENUNCIADOS EJERCICIOS CURSO DE JAVASCRIPT DESDE CERO

## Primeros pasos

1. Escribe un comentario en una línea.
2. Escribe un comentario en varias líneas.
3. Declara variables con valores asociados a todos los datos de tipo primitivos.
4. Imprime por consola el valor de todas las variables.
5. Imprime por consola el tipo de todas las variables.
6. A continuación, modifica los valores de las variables por otros del mismo tipo.
7. A continuación, modifica los valores de las variables por otros de distinto tipo.
8. Declara constantes con valores asociados a todos los tipos de datos primitivos.
9. A continuación, modifica los valores de las constantes.
10. Comenta las líneas que produzcan algún tipo de error al ejecutarse.

# Operadores

1. Crea una variable para cada operación aritmética.
2. Crea una variable para cada tipo de operación de asignación, que haga uso de las variables utilizadas para las operaciones aritméticas.
3. Imprime 5 comparaciones verdades con diferentes operadores de comparación.
4. Imprime 5 comparaciones falsas con diferentes operadores de comparación.
5. Utiliza el operador lógico and.
6. Utiliza el operador lógico or.
7. Combina ambos operadores lógicos.
8. Añade alguna negación.
9. Utiliza el operador ternario.
10. Combina operadores aritméticos, de comparación y lógicas.

# Strings

1. Concatena dos cadenas de texto.
2. Muestra la longitud de una cadena de texto.
3. Muestra el primer y último carácter de un string.
4. Convierte a mayúsculas y minúsculas un string.
5. Crea una cadena de texto en varias líneas.
6. Interpola el valor de una variable en un string.
7. Reemplaza todos los espacios en blanco de un string por guiones.
8. Comprueba si una cadena de texto contiene una palabra concreta.
9. Comprueba si dos strings son iguales.
10. Comprueba si dos strings tienen la misma longitud.

# Condicionales

if/else/else if/ternaria

1. Imprime por consola tu nombre si una variable toma su valor.
2. Imprime por consola un mensaje si el usuario y contraseña concide con unos establecidos.
3. Verifica si un número es positivo, negativo o cero e imprime un mensaje.
4. Verifica si una persona puede votar o no (mayor o igual a 18) e indica cuántos años le faltan.
5. Usa el operador ternario para asignar el valor "adulto" o "menor" a una variable dependiendo de la edad .
6. Muestra en que estación del año nos encontramos dependiendo del valor de una variable "mes".
7. Muestra el número de días que tiene un mes dependiendo de la variable del ejercicio anterior.

switch

8. Usa un switch para imprimir un mensaje de saludo diferente dependiendo del idioma.
9. Usa un switch para hacer de nuevo el ejercicio 6.
10. Usa un switch para hacer de nuevo el ejercicio 7.

# Estructuras

1. Crea un array que almacene cinco animales.
2. Añade dos más. Uno al principio y otro al final.
3. Elimina el que se encuentra en tercera posición.
4. Crea un set que almacene cinco libros.
5. Añade dos más. Uno de ellos repetido.
6. Elimina uno concreto a tu elección.
7. Crea un mapa que asocie el número del mes a su nombre.
8. Comprueba si el mes número 5 existe en el map e imprime su valor.
9. Añade al mapa una clave con un array que almacene los meses de verano.
10. Crea un Array, transfórmalo a un Set y almacénalo en un Map.

# Bucles

NOTA: Explora diferentes sintaxis de bucles para resolver los ejercicios.

1. Crea un bucle que imprima los números del 1 al 20.
2. Crea un bucle que sume todos los números del 1 al 100 y muestre el resultado.
3. Crea un bucle que imprima todos los números pares entre 1 y 50.
4. Dado un array de nombres, usa un bucle para imprimir cada nombre en la consola.
5. Escribe un bucle que cuente el número de vocales en una cadena de texto.
6. Dado un array de números, usa un bucle para multiplicar todos los números y mostrar el producto.
7. Escribe un bucle que imprima la tabla de multiplicar del 5.
8. Usa un bucle para invertir una cadena de texto.
9. Usa un bucle para generar los primeros 10 números de la secuencia de Fibonacci.
10. Dado un array de números, usa un bucle para crear un nuevo array que contenga solo los números mayores a 10.

# Funciones

NOTA: Explora diferentes sintaxis de funciones para resolver los ejercicios.

1. Crea una función que reciba dos números y devuelva su suma.
2. Crea una función que reciba un array de números y devuelva el mayor de ellos.
3. Crea una función que reciba un string y devuelva el número de vocales que contiene.
4. Crea una función que reciba un array de strings y devuelva un nuevo array con las strings en mayúsculas.
5. Crea una función que reciba un número y devuelva true si es primo, y false en caso contrario.
6. Crea una función que reciba dos arrays y devuelva un nuevo array que contenga los elementos comunes entre ambos.
7. Crea una función que reciba un array de números y devuelva la suma de todos los números pares.
8. Crea una función que reciba un array de números y devuelva un nuevo array con cada número elevado al cuadrado.
9. Crea una función que reciba una cadena de texto y devuelva la misma cadena con las palabras en orden inverso.
10. Crea una función que calcule el factorial de un número dado.

# Objetos

1. Crea un objeto con 3 propiedades.
2. Accede y muestra su valor.
3. Agrega una nueva propiedad.
4. Elimina una de las 3 primeras propiedades.
5. Agrega una función e invócala.
6. Itera las propiedades del objeto.
7. Crea un objeto anidado.
8. Accede y muestra el valor de las propiedades anidadas.
9. Comprueba si los dos objetos creados son iguales.
10. Comprueba si dos propiedades diferentes son iguales.



# Desestructuración y propagación

1. Usa desestructuración para extraer los dos primeros elementos de un array.
2. Usa desestructuración en un array y asigna un valor predeterminado a una variable.
3. Usa desestructuración para extraer dos propiedades de un objeto.
4. Usa desestructuración para extraer dos propiedades de un objeto y asígnalas a nuevas variables con nombres diferentes.
5. Usa desestructuración para extraer dos propiedades de un objeto anidado.
6. Usa propagación para combinar dos arrays en uno nuevo.
7. Usa propagación para crear una copia de un array.
8. Usa propagación para combinar dos objetos en uno nuevo.
9. Usa propagación para crear una copia de un objeto.
10. Combina desestructuración y propagación.

# Clases

1. Crea una clase que reciba dos propiedades.
2. Añade un método a la clase que utilice las propiedades.
3. Muestra los valores de las propiedades e invoca a la función.
4. Añade un método estático a la primera clase.
5. Haz uso del método estático.
6. Crea una clase que haga uso de herencia.
7. Crea una clase que haga uso de getters y setters.
8. Modifica la clase con getters y setters para que use propiedades privadas.
9. Utiliza los get y set y muestra sus valores.
10. Sobrescribe un método de una clase que utilice herencia.

# Manejo de errores

1. Captura una excepción utilizando try-catch.
2. Captura una excepción utilizando try-catch y finally.
3. Lanza una excepción genérica.
4. Crea una excepción personalizada.
5. Lanza una excepción personalizada.
6. Lanza varias excepciones según una lógica definida.
7. Captura varias excepciones en un mismo try-catch.
8. Crea un bucle que intente transformar a float cada valor y capture y muestre los errores.
9. Crea una función que verifique si un objeto tiene una propiedad específica y lance una excepción personalizada.
10. Crea una función que realice reintentos en caso de error hasta un máximo de 10.

# Console

1. Crea un función que utilice error correctamente.
2. Crea una función que utilice warn correctamente.
3. Crea una función que utilice info correctamente.
4. Utiliza table.
5. Utiliza group.
6. Utiliza time.
7. Valida con assert si un número es positivo.
8. Utiliza count.
9. Utiliza trace.
10. Utiliza clear.

# Módulos

1. Exporta una función.
2. Exporta una constante.
3. Exporta una clase.
4. Importa una función.
5. Importa una constante.
6. Importa una clase.
7. Exporta una función, una constante y una clase por defecto (en caso de que lo permita).
8. Importa una función, una constante y una clase por defecto (en caso de que lo permita).
9. Exporta una función, una constante y una clase desde una carpeta.
10. Importa una función, una constante y una clase desde un directorio diferente al anterior.

