

Ejercicios Unidad 1: Sintaxis básica del lenguaje.

EJERCICIO: u1e01_tablaMultiplicar

Crea un programa que pida al usuario un número y escriba en pantalla la tabla de multiplicar de ese número, utilizando `document.write()`.

Para pedir un valor del usuario utilizamos el siguiente código:

```
var respuesta = prompt("Mensaje");
```

EJERCICIO: u1e02_edad

Crea un programa que pida al usuario que introduzca una edad y muestre el siguiente mensaje en función del número introducido:

- 0-12: Niño
- 13-26: Joven
- 26-60: Adulto
- >60: Jubilado

Si la edad es menor que 0, dará un mensaje de error.

EJERCICIO: u1e03_area

Escribir un script que permita calcular el área y el perímetro de cada una de las siguientes figuras:

a) Un cuadrado b) Un triángulo d) Un círculo

Para ello el script pedirá primero al usuario cual es la figura con la que desea trabajar. El usuario responderá: "S" para el cuadrado (square), "T" para el triángulo (triangle) y "C" para el círculo (circle). Una vez se sepa la figura, el script pedirá al usuario los datos que necesite (se repetirá la pregunta tantas veces como datos se necesite).

Tener en cuenta que:

- Área del cuadrado: lado^2 ;
- Área del triángulo: $(\text{base} * \text{altura})/2$;
- Área del círculo: $\text{PI} * \text{radio}^2$
- Perímetro del círculo: $2 * \text{PI} * \text{radio}$
- El valor de PI se obtiene con `Math.PI` (PI es una constante del objeto `Math`).

Finalmente, el script escribirá en la página web: los datos que le ha facilitado el usuario y el resultado pedido.

Ejemplo (para el cuadrado):

Se pregunta: "¿Que figura desea calcular?"

El usuario responde: "C"

Se pregunta: "¿Cuál es la longitud del lado?"

El usuario responde: "2"

Respuesta:

Cuadrado

lado: 2 cm

área: 4 cm²

perímetro: 8 cm

EJERCICIO: u1e04_mediaHora

Crea un programa que genere una página con un listado de horas que vayan desde las 9 hasta las 21:30 de 30 minutos en 30 minutos. Ej.: 9:00, 9:30,

EJERCICIO: u1e05_cincoMinutos

Modifica el ejercicio anterior, pero en este caso el intervalo será de 5 minutos. Los minutos se deben escribir siempre con 2 cifras. Ej.: 9:00, 9:05, 9:10, ...

EJERCICIO: u1e06_presiApellido

Crea un control parental para que no se pueda entrar en una página. Se pedirá al usuario que escriba el apellido del primer presidente de la democracia. Mientras el usuario no introduzca el valor correcto, el mensaje seguirá saliendo.

EJERCICIO: u1e07_presiNombreApellido

Modifica el ejercicio anterior de la siguiente forma:

- En primer lugar, se carga un prompt donde pregunta: "¿Cuál fue el primer presidente de la democracia española?"
- En caso de que el usuario introduzca mal nombre y apellido, muestre como mensaje: "ERROR. Inténtelo de nuevo. ¿Cuál fue el primer presidente de la democracia española?"
- En caso de que el usuario introduzca el nombre del presidente (solamente) el prompt muestre como mensaje "Te falta el apellido. ¿Cuál fue el primer presidente de la democracia española?"
- En caso de que el usuario introduzca el apellido del presidente (solamente) el prompt muestre como mensaje "Te falta el nombre. ¿Cuál fue el primer presidente de la democracia española?"

EJERCICIO: u1e08_multiplos23

Realizar un script que escriba en la página los números del 1 al 100. Junto a ellos, escribirá el mensaje "es múltiplo de 2" y "es múltiplo de 3" SOLO junto a aquellos números en que se cumpla esta condición. En el caso de los números que sean múltiplos de ambos, se mostrarán ambos mensajes.

LISTADO DE NUMEROS DEL 1 AL 100

```
El 1
El 2 es múltiplo de 2
El 3 es múltiplo de 3
El 4 es múltiplo de 2
El 5
El 6 es múltiplo de 2 es múltiplo de 3
El 7
El 8 es múltiplo de 2
El 9 es múltiplo de 3
El 10 es múltiplo de 2
```

EJERCICIO: u1e09_edad

Crea una nueva versión del ejercicio u1e03_edad, pero utilizando el operador ternario.

EJERCICIO: u1e10_triangulo

Crea un bucle que haga siete llamadas a `console.log`, para obtener el siguiente triángulo:

```
#  
##  
###  
####  
#####  
#####  
#####
```

Puede ser útil saber que puedes obtener la longitud de una cadena escribiendo `length` después de ella.

```
var cad = "cad"; console.log(cad.length); // → 3
```

EJERCICIO: u1e11_fizzbuzz

Escribe un programa que utilice `console.log` para imprimir todos los números de 1 a 100, con tres excepciones. Para los números divisibles por 3, escribirá "Fizz" en lugar del número, y para los números divisibles por 5 (pero no por 3), escribirá "Buzz" en su lugar. Además, para los números divisibles por 3 y por 5, escribirá "FizzBuzz".

EJERCICIO: u1e12_tablero

Escribe un programa que crea una cadena que representa una rejilla de 8 x 9, utilizando el carácter "nueva línea" para separar líneas. En cada posición de la rejilla, habrá un espacio o el carácter "#". Los caracteres formarán un tablero de ajedrez.

Al escribir esa cadena por `console.log`, se mostrará algo similar a esto:

```
# # # #  
# # #  
# # # #  
# # #
```

Cuando funcione, define una variable `tam = 8`; y cambia el programa para que funcione para cualquier tamaño, mostrando una rejilla del ancho y alto dados.

EJERCICIO: u1e13_minimo

La función estándar `Math.min` admite varios argumentos y devuelve el que tiene menor valor. Escribe una función **min** que recibe dos argumentos y devuelve su valor menor.

EJERCICIO: u1e14_contarBs

Se puede obtener el carácter *n*ésimo de una cadena con `"string"[n]`. El valor devuelto será una cadena que solo tiene una letra, por ejemplo la "b". El primer carácter tiene la posición 0.

Además, las cadenas tienen una propiedad `length`, que contiene el número de caracteres del string. Por tanto, el último carácter estará en la posición `"string".length-1`.

Escribe una función **contarBs** que tome una cadena como argumento, y devuelva el número de "B" mayúsculas que hay en la cadena.

A continuación, escribe una función **contarCaracter**, similar a `contarBs`, pero que su segundo argumento indica el carácter que se quiere contar (en lugar de contar "B" mayúsculas).

EJERCICIO: u1e15_pares

El operador resto (%) se puede utilizar para comprobar si un número es par o impar, utilizando % 2, para saber si es divisible por dos.

Hay otro modo de definir si un número positivo entero es par o impar:

- 0 es par
- 1 es impar
- Para cualquier otro número N, es par si lo es N-2; o impar si lo es N-2.

Define una función recursiva **esPar** de acuerdo con esta definición. La función debería aceptar un solo parámetro y devolver un boolean.

EJERCICIO: u1e16_factorial

El factorial de un número se puede definir como:

- factorial (0) = 1
- factorial (n) = n * factorial (n-1) //si n>0

Crea una función **factorialIter**, que calcule el factorial de un número, utilizando estructuras iterativas.

Crea otra función **factorialRecur**, que lo calcule utilizando recursividad.

EJERCICIO: u1e17_fibonacci

La serie de Fibonacci se forma de la siguiente forma:

- Fibonacci(0) = 0
- Fibonacci(1) = 1
- Fibonacci(n) = Fibonacci(n-1)+Fibonacci(n-2) // si n>1

Crea una función **fibolter**, que escriba los números de la serie de fibonacci, hasta esa posición (empezamos por cero), utilizando estructuras iterativas.

Crea otra función **fiborRecur**, que lo calcule utilizando recursividad.

EJERCICIO: u1e18_binario

Crea una función binario, que pida un número y escriba ese número en binario. Para ello, habrá que pasarlo a binario dividiendo entre 2, y construir una cadena con 0 y 1.

EJERCICIO: u1e19_literales

Haz una página web en la que muestres (por consola) todos los tipos de literales que existen en JavaScript.

Puedes ver más información en:

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar_and_types#Data_structures_and_types (apartado Literales).

EJERCICIO: u1e20_caracteresEspeciales

Haz una página web en la que muestre un alert con una cadena que contenga los caracteres especiales ', ", \, salto de línea, el símbolo copyright referenciándolo en octal, hexadecimal y Unicode.

Puedes ver más información en:

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Grammar_and_types#Data_structures_and_types (apartado Uso de caracteres especiales en cadenas).

EJERCICIO: u1e21_numeros

Crea un programa en el que crees 5 variables numéricas (entero, decimal, científico, octal y hexadecimal).

A las variables les asignarás los siguientes números: 1357, 135.7, 135e7, 01357 y 0x1357.

Muestra con 5 alerts su valor, escribiendo la siguiente sentencia:

```
alert ("Número entero: " + entero);
```

EJERCICIO: u1e22_cadenas

Crea un programa en el que se definan 4 variables (2 cadenas y 2 números), con los siguientes valores: tu nombre, tu apellido, tu edad y tu año de nacimiento.

- Muestra en un alert una frase que incluya comillas simples.
- Muestra en un alert tu nombre y apellidos separados por un salto de línea.
- Muestra en un alert la suma de las variables edad y año de nacimiento.
- Muestra en un alert la suma de todas las variables.

EJERCICIO: u1e23_comparacion

Crea un programa en el que muestres el resultado de varias operaciones mediante alert, mostrando el texto exacto de la operación realizada y su resultado.

Ej:

- `var operacion1 = (10 == 10);`
- `alert ("La operación 10 == 10 es " + operacion1)`

Las operaciones a realizar son:

- `10 == 10`
- `10 === 10`
- `10 === 10.0`
- `"Ada" == "ada"`
- `"Ada" > "ada"`
- `"Ada" < "ada"`
- `"123" == 123`
- `"123" === 123`
- `parseInt("123") === 123`