

1. Escribe una función que escriba "Hola" 3 veces.
2. Escribe una función que escriba "Hola" tantas veces como el número que le pasamos de parámetro. Llámala dos veces, primero con 5 y después con 3.
3. Escribe una función que devuelva el resultado de sumar los dos parámetros que le pasamos. Muestra el resultado en la consola.
4. Escribe una función que devuelva el resultado de sumar los dos parámetros que le pasamos. Pide al usuario 2 números. Usa esos 2 números como parámetros de la función y después asigna el valor que devuelve la función a una tercera variable. Finalmente, muestra el valor de la variable en la consola.
5. Escribe una función a la que le damos dos números y que devuelva true si el primer número es más grande que el segundo número y false en el caso contrario. Pide dos números al usuario, y pásalos a la función. Muestra en consola "El primer número es más grande" o "El primer número no es más grande".
6. Escribe un programa que pida al usuario dos números. Usa esos 2 números en una función que divide el primero por el segundo y devuelve el resultado. Muestra el resultado que devuelve la función en un alert. Después vuelve a pedir los números y hacer todo otra vez hasta que el usuario introduzca un 0 como segundo número.
7. Diseña una función esPar() que reciba como parámetro un número y devuelva si ese número es par o no. Utilízalo en un programa que lea un número por prompt y determine si es par o no y muestre un mensaje al respecto en un alert.
8. Crea una aplicación que nos calcule el factorial de un número pedido por teclado, lo realizará mediante una función al que le pasamos el número como parámetro. Para calcular el factorial, se multiplica los números anteriores hasta llegar a uno. Por ejemplo, si introducimos un 5, realizará esta operación $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$. Muestra el resultado en un alert.
9. Crea una aplicación que nos cuente el número de cifras de un número entero positivo (hay que confirmar que no haya introducido un número negativo y deberemos quitar los decimales) pedido por teclado. Crea un método que realice esta acción: pasando el número por parámetro, devolverá el número de cifras. Después mostrar ese número en un alert.
10. Crea una aplicación que nos calcule el área de un círculo, cuadrado o triángulo. Pediremos de qué figura queremos calcular su área y según lo introducido pedirá los valores necesarios para calcular el área. Crea un método por cada figura para calcular cada área, este devolverá un número entero(sin decimales). Muestra el resultado por pantalla
 1. Círculo: $(\text{radio}^2) \cdot \pi$
 2. Triángulo: $(\text{base} \cdot \text{altura}) / 2$
 3. Cuadrado: $\text{lado} \cdot \text{lado}$
11. Crea una aplicación que nos convierta una cantidad de euros introducida por teclado a otra moneda, estas pueden ser a dólares, yenes o libras. El método tendrá como parámetros, la cantidad de euros y la moneda a pasar que será una cadena, este no devolverá ningún valor, mostrará un mensaje indicando el cambio.
 1. 0.86 libras es un 1 €
 2. 1.28611 \$ es un 1 €
 3. 129.852 yenes es un 1 €
12. Escribe una función a la que le pasamos un año y devuelve true si es bisiesto. En el programa principal, pídele al usuario un número y muestra en pantalla "El año ... es bisiesto" o "El año ... no es bisiesto" y sigue pidiendo números y diciendo si es o no bisiesto hasta que introduzca un 0. Un año es bisiesto si es divisible entre 4, pero no si es divisible entre 100 pero si es divisible entre 400 sí lo es.

-
1. Crea una función llamada `temporizador` que reciba un número de segundos y un callback. La función debe utilizar `setTimeout` para esperar los segundos indicados y luego ejecutar el callback que muestra un mensaje en la consola.
 2. Crea una función `descargarDatos` que devuelva una Promesa. La promesa debe simular la descarga de datos usando un `setTimeout` de 2 segundos. Si el proceso tiene éxito, resuelve la promesa con un mensaje de éxito. Si hay un error (simulado con una probabilidad del 30%), rechaza la promesa con un mensaje de error.
 3. Crea tres funciones: `operacion1`, `operacion2` y `operacion3`. Cada función devuelve una Promesa que se resuelve después de 1 segundo con un mensaje indicando el resultado de la operación. Encadena las tres funciones usando `.then` y muestra los mensajes en orden.
 4. Dada una función que utiliza un callback para manejar tareas asíncronas, conviértela para que utilice una promesa en su lugar.

```
javascript
function leerArchivo(nombreArchivo, callback) {
  setTimeout(() => {
    if (nombreArchivo) {
      callback(null, `Contenido del archivo: ${nombreArchivo}`);
    } else {
      callback("Error: No se proporcionó un nombre de archivo", null);
    }
  })
}
```

```
    }, 1000);  
  }
```

```
// Ejemplo de uso con callback leerArchivo("archivo.txt", (error, contenido) => { if (error) { console.error(error); } else {  
console.log(contenido); } });
```