

Guía práctica funciones

Nombre de la declaración de la función.

parámetros

```
1 function ordinary1(a, b, c) {  
2     // implementación de la función  
3 }  
4  
5 const ordinary2 = function (a, b, c)  
6     return a + b + c;  
7 };
```

Las llaves ({ }) y todo lo que hay entre ellas son el **cuerpo de la declaración de la función**.

instrucción devuelve explícitamente un valor de la función

Rol desempeñado por las funciones

```
1 function add(x, y) {  
2     return x + y;  
3 }  
4 console.log(add(x: 2, y: 3));  
5  
6 const objeto = { metodo1 : add }  
7  
8 console.log(objeto.metodo1(x: 4, y: 5));
```

invocada a través de una llamada de la función.

almacenado en una propiedad

invocado a través de una llamada de método.

```
function saludar() {  
    console.log("¡Hola!");  
}
```

En este ejemplo, la función `saludar` no recibe ningún parámetro y no devuelve ningún valor. Simplemente imprime el mensaje "¡Hola!" en la consola cuando se la llama. Para llamar a esta función, puedes hacer lo siguiente:

```
saludar(); // Salida: "¡Hola!"
```

```
function saludarPersona(nombre) {  
  console.log(`¡Hola, ${nombre}!`);  
}
```

En este ejemplo, la función `saludarPersona` recibe un parámetro `nombre`, que se utiliza para personalizar el mensaje que se imprime en la consola. Sin embargo, esta función no devuelve ningún valor.

Puedes llamar a esta función de la siguiente manera:

```
saludarPersona("Juan"); // Salida: "¡Hola, Juan!"  
saludarPersona("María"); // Salida: "¡Hola, María!"
```

```
function sumar(a, b) {  
  const resultado = a + b;  
  return resultado;  
}
```

En este ejemplo, la función `sumar` recibe dos parámetros `a` y `b`. Dentro de la función, se realiza la operación de suma de estos dos valores y se almacena el resultado en la variable `resultado`. Luego, la función utiliza la palabra clave `return` para devolver el valor de `resultado`.

Puedes llamar a esta función de la siguiente manera:

```
const suma1 = sumar(2, 3); // suma1 tendrá el valor 5  
const suma2 = sumar(5, 7); // suma2 tendrá el valor 12  
console.log(suma1); // Salida: 5  
console.log(suma2); // Salida: 12
```

Seguimiento 1 corte 2:

1. **Calcular el factorial de un número:** Crea una función que reciba un número entero positivo y devuelva su factorial. Utiliza un ciclo para calcular el resultado.
2. **Verificar si un número es primo:** Crea una función que reciba un número entero y devuelva `true` si es primo, y `false` en caso contrario. Utiliza un ciclo y una condición para verificar si el número es divisible por otros números distintos de 1 y sí mismo.

3. **Calcular la suma de dígitos de un número:** Crea una función que reciba un número entero y devuelva la suma de sus dígitos. Utiliza un ciclo y una decisión para extraer los dígitos uno por uno y sumarlos.
4. **Generar una serie de Fibonacci hasta un límite:** Crea una función que reciba un número entero n y devuelva un array con los n primeros números de la serie de Fibonacci. Utiliza un ciclo y una decisión para calcular los números de la serie.
5. **Invertir una cadena de texto:** Crea una función que reciba una cadena de texto y devuelva la misma cadena, pero invertida. Utiliza un ciclo para recorrer la cadena de atrás hacia adelante y construir la nueva cadena invertida.
6. **Calcular el máximo común divisor de dos números:** Crea una función que reciba dos números enteros y devuelva el máximo común divisor de ambos. Utiliza un ciclo y una decisión para encontrar el mayor número que divide a ambos números sin dejar resto.
7. **Contar las vocales en una cadena de texto:** Crea una función que reciba una cadena de texto y devuelva el número de vocales que contiene. Utiliza un ciclo y una decisión para recorrer cada carácter de la cadena y verificar si es una vocal.
8. **Calcular la potencia de un número:** Crea una función que reciba dos números enteros base y exponente, y devuelva la base elevada a la exponente. No utilices la función integrada `Math.pow()`. Utiliza un ciclo y una decisión para calcular la potencia.
9. **Verificar si un número es palindrómico:** Crea una función que reciba un número entero y devuelva `true` si es palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda) y `false` en caso contrario. Utiliza un ciclo y una decisión para invertir el número y compararlo con el original.
10. **Calcular la suma de los divisores de un número:** Crea una función que reciba un número entero y devuelva la suma de todos sus divisores (excluyendo al propio número). Utiliza un ciclo para recorrer todos los posibles divisores y una decisión para identificar los divisores reales y sumarlos.

