



Funciones en Javascript

Las funciones son bloques de código reutilizables que realizan una tarea específica. En JavaScript, puedes definir funciones con el siguiente formato:

```
// Definición de una función

function nombreDeLaFuncion(parametro1, parametro2) {

    // Cuerpo de la función

    // Puedes realizar operaciones aquí


    // Devolver un valor (opcional)

    return resultado;

}
```

```
// Llamada a la función

let resultado = nombreDeLaFuncion(valor1, valor2);
```

Ejemplos:

1. Función Básica:

/ Función que suma dos números

```
function suma(a, b) {

    return a + b;

}
```

```
// Llamada a la función

let resultadoSuma = suma(3, 5);

console.log("Resultado de la suma: " + resultadoSuma);
```

2. Función con Condición:

```
// Función que determina si un número es par o impar

function esPar(numero) {

    if (numero % 2 === 0) {

        return true;

    } else {

        return false;

    }

}
```

```
// Llamada a la función

let numeroAnalizado = 7;

if (esPar(numeroAnalizado)) {

    console.log(numeroAnalizado + " es par.");

} else {

    console.log(numeroAnalizado + " es impar.");

}
```

3. Función con Valor Predeterminado:

```
// Función que saluda con un mensaje personalizado

function saludar(nombre = "Invitado") {
```

```
    return "Hola, " + nombre + "!";  
}
```

```
// Llamada a la función
```

```
let saludo = saludar("Juan");
```

```
console.log(saludo);
```

```
// Llamada sin argumento
```

```
let saludoPredeterminado = saludar();
```

```
console.log(saludoPredeterminado);
```

4. Función Anónima (Expresión de Función):

```
/ Expresión de función
```

```
let multiplicar = function(a, b) {
```

```
    return a * b;
```

```
};
```

```
// Llamada a la función
```

```
let resultadoMultiplicacion = multiplicar(4, 6);
```

```
console.log("Resultado de la multiplicación: " + resultadoMultiplicacion);
```

5. Función Flecha (ES6+):

```
/ Función flecha que eleva un número al cuadrado
```

```
let cuadrado = (x) => x ** 2;
```

```
// Llamada a la función
```

```
let resultadoCuadrado = cuadrado(3);
```

```
console.log("Resultado del cuadrado: " + resultadoCuadrado);
```

Estos son ejemplos básicos, pero las funciones en JavaScript son extremadamente flexibles y pueden aceptar cualquier número de parámetros, realizar diversas operaciones y devolver resultados. También pueden ser utilizadas como argumentos para otras funciones y asignadas a variables, lo que las hace fundamentales en la programación modular y funcional.