## **Funciones con Parámetros por Defecto en TypeScript**

En TypeScript, como en JavaScript, puedes definir funciones con **parámetros por defecto**. Esto significa que si al llamar a la función no se proporciona un valor para un parámetro determinado, se utilizará el valor por defecto asignado en la definición de la función.

**¿Por qué usar parámetros por defecto?**

* **Mayor flexibilidad:** Permite llamar a la función de diferentes maneras, simplificando su uso.
* **Valores predeterminados:** Establece valores por defecto para los parámetros, evitando errores si no se proporcionan.
* **Mejor legibilidad:** Hace el código más claro y conciso.

**Sintaxis:**

TypeScript

function nombreFuncion(parametro1: tipo, parametro2: tipo = valorPorDefecto): tipo { // Cuerpo de la función }

Use code [with caution.](file:///C:/faq#coding)

**Ejemplos:**

### **Ejemplo 1: Saludo Personalizado (saludoPersonalizado.ts)**

TypeScript

function saludar(nombre: string, mensaje: string = 'Hola') { // Si no se proporciona un mensaje, se utiliza 'Hola' por defecto. console.log(`${mensaje}, ${nombre}!`); } saludar('Juan'); // Imprime: Hola, Juan! saludar('Ana', '¡Buenos días!'); // Imprime: ¡Buenos días!, Ana!

Use code [with caution.](file:///C:/faq#coding)

### **Ejemplo 2: Cálculo de Área de un Rectángulo (calcularAreaRectangulo.ts)**

TypeScript

function calcularArea(base: number, altura: number = 1) { // Si no se proporciona la altura, se asume que es 1. return base \* altura; } let area1 = calcularArea(5); // Imprime: 5 (base 5, altura 1 por defecto)let area2 = calcularArea(3, 4); // Imprime: 12

Use code [with caution.](file:///C:/faq#coding)

### **Ejemplo 3: Generador de Contraseñas (generarContrasena.ts)**

TypeScript

function generarContrasena(longitud: number = 8, incluirNumeros: boolean = true, incluirMayusculas: boolean = true) { // Genera una contraseña aleatoria con las opciones especificadas. // ... (implementación de la lógica de generación de contraseñas) return contrasena; } let contrasena1 = generarContrasena(); // Contraseña de 8 caracteres, con números y mayúsculaslet contrasena2 = generarContrasena(12, false, true); // Contraseña de 12 caracteres, solo mayúsculas

Use code [with caution.](file:///C:/faq#coding)

### **Ejemplo 4: Función para Sumar Números (sumarNumeros.ts)**

TypeScript

function sumar(...numeros: number[]): number { // Suma todos los números proporcionados. return numeros.reduce((total, numero) => total + numero, 0); } let resultado1 = sumar(1, 2, 3); // Imprime: 6let resultado2 = sumar(10, 20); // Imprime: 30

Use code [with caution.](file:///C:/faq#coding)

**Explicación de los ejemplos:**

* **saludoPersonalizado.ts:** Demuestra cómo usar un parámetro por defecto para el mensaje de saludo.
* **calcularAreaRectangulo.ts:** Calcula el área de un rectángulo, asumiendo una altura de 1 si no se proporciona.
* **generarContrasena.ts:** Genera contraseñas aleatorias con diferentes opciones de personalización.
* **sumarNumeros.ts:** Suma un número variable de argumentos utilizando el operador rest (...).

**En resumen:**

Los parámetros por defecto en TypeScript te permiten crear funciones más flexibles y fáciles de usar. Al asignar valores predeterminados a los parámetros, puedes simplificar las llamadas a la función y evitar errores comunes.