💻 Lenguaje Typescript

Tipos de dato

Nombre	Ejemplo
`number`	gestion=2025, edad=15
`string`	nombre='Juan', cena='Pollo'
`boolean`	habilitado=false, top=true

Variables

Se puede asignar valores luego de ser definida

```
let anio: number = 2025;
let latitud: number = 34.8514;
let mes: string = 'Agosto';
let repobado: boolean = false;
```

Constantes

No pueden cambiar su valor una vez definida

```
const nombre: string = 'Juan';
const edad: number = 15;
const peso: number = 46.500;
const bueno: boolean = false;
```

En los siguientes ejemplos se producirán errores porque se declara una **constante** y luego se intenta cambiar su valor, lo cual no está permitido

```
const nombre: string = 'Ana';
nombre = 'Anita';

const notaFinal: number = 95;
notaFinal = 100;
```

Para corregir el error, se debe cambiar la palabra `const` por `let`

Estructura de una funcion

Nombre: Este nombre debe ser descriptivo para indicar qué hace la función.

Parámetros: Son cero, uno o más valores de entrada, que la función utiliza para realizar su tarea.

Sentencias/Bloque de codigo: Es el conjunto de instrucciones que se ejecutan cuando la función es llamada.

Valor de retorno: Valor como resultado de la tarea.

Llamada a la función: Para usar una función, se "llama" por su **nombre** seguido de paréntesis. Si la función requiere parámetros, estos se colocan dentro de los paréntesis.

Declaracion de funciones

Funcion con **valor de retorno** `number`

```
function sumar(numX: number, numY: number): number {
  return 0;
}
```

Funcion con **valor de retorno** `string`

```
function enviarSaludo(usuario: string): string {
  return ‘Hola’;
}
```

Funcion con **valor de retorno** `boolean`

```
function registrarUsuario(persona: string): boolean {
  return false;
}
```

Funcion con **valor de retorno** `void`

```
function destruirEnemigo(cantidadBombas: number): void {

}
```

Ejercicios de examen