

# TYPESCRIPT

---

Es un superset de JavaScript desarrollado por Microsoft. Lenguaje de programación con una serie de mejoras. Es hasta el 85% JavaScript.

**Importante:** TypeScript compila el código que hayamos desarrollado con nuevas funcionalidades que podrían no funcionar en todos los navegadores y lo convierte en JavaScript que sea compatible con todos los navegadores, y esto lo hace de forma automática.

Es un lenguaje interpretado en JavaScript. Extensión: archivo.ts

## HOLA MUNDO CON TYPESCRIPT

---

El contenido del archivo TS no se mostrará directamente en el navegador. Para ello se necesitará que todo lo codificado en el archivo TS se pase a un archivo JS a través de la terminal

En la terminal se debe colocar la Ruta de la carpeta e ir hacia donde se encuentra el archivo.ts y colocar:

tsc archivo.ts --> Esto creará automáticamente 1 archivo JS copia del archivo.ts

Para que los cambios en ambos archivos sean automáticos en la terminal se debe ir a la ruta donde se encuentra el archivo.ts y se debe colocar lo siguiente:

tsc -w \*.ts

## TIPOS DE DATOS EN TYPESCRIPT

```
// String
let cadena: string = "victorrobles.web";
console.log(cadena);

// Number
let numero: number = 12;

// Booleano
let verdadero_falso: boolean = true;

// Any (se puede escribir cualquier tipo de dato)
let cualquiera: any = "hola";
cualquiera = 77;

// Arrays
let lenguajes: Array<string> = ["PHP", "12", "CSS"]; // Forma 1
let years: number[] = [12, 13, 14]; // Forma 2

// Múltiples Tipos de datos
let cadena: string | number = "victorrobles.web";
```

```
// Crear un nuevo Tipo de Dato
type alfanumerico = string | number;
```

---

## LET Y VAR EN TYPESCRIPT

Diferencia entre Let y Var radica en el alcance de la variable. Let: Variable a nivel de bloque. Var: Variable a nivel global.

```
var numero1 = 10; // Variable global
var numero2 = 12; // Variable global

if(numero1 == 10){
    let numero1 = 44; // Variable local (solo funcionará dentro del If)
    var numero2 = 55; // Variable global (como es de caracter global. reemplazará
    a todo valor que s le haya asignado a la variable con anterioridad)

    console.log(numero1, numero2); // Mostrará 44 y 55
}

console.log(numero1, numero2); // Mostrará 10 y 55
```

---

## FUNCIONES Y TIPADO FUERTE

Se pueden escribir funciones como en JavaScript

```
function getNumero(numero= 12){
    return "El número es: " + numero;
}
console.log(getNumero(55));
```

Pero en TypeScript también puedes colocar el tipo de dato de entrada y el tipo de Dato de salida.

```
function getNumero(numero: number = 12):string{
    return "El número es: " + numero;
}
console.log(getNumero(55));
```

**IMPORTANTE:** Si los tipos de datos introducidos como entrada o como de salida no coinciden, la mayoría de veces el resultado se mostrará en el navegador, pero en la *Consola* lanzará un error.