**TypeScript**

**Void.**

**Documentación:**

**<https://www.typescriptlang.org/docs/handbook/basic-types.html#void>**

[**https://www.typescriptlang.org/docs/handbook/2/everyday-types.html**](https://www.typescriptlang.org/docs/handbook/2/everyday-types.html)

El tipo de dato void es muy interesante, por lo que vale la pena entenderlo. **Void es un poco como lo contrario de “la ausencia de tener cualquier tipo en absoluto”. Por lo general, puede ver esto como el tipo de retorno de funciones que no devuelven un valor <any>.**

Void significa literalmente vació, esto es importante saberlo por que una función en JS que no tenga un retorno implícito, es decir que una función no tenga la palabra **return** al final de su bloque de código, regresara la palabra undefined. Ejemplo:

function saludar(){}

const s: void = saludar();

console.log({s}) // Consola: {s: undefined}

**Y es importante saber que undefined no es lo mismo que null.**

¿Entonces para que nos sirve void?

Void es una manera de indicar en TS que no existe un valor de retorno al ejecutar una función. Usualmente en una función que regrese void, es decir que no regresa nada, sería correcto ponerle el tipo de retorno a la función. Ejemplo:

function saludar(): void {}

const s: void = saludar();

console.log({s}) // Consola: {s: undefined}

Y si regresamos un valor en la función que especificamos como void, TS nos mandaría un error. Ejemplo:

function saludar(): void {

    return 1; // Error: El tipo 'number' no se puede asignar al tipo 'void'.

}

const s: void = saludar();

console.log({s}) // Consola: {s: undefined}

Ejemplo de retornar void en una función de flecha:

const gritar = (): void => {}

const gritar = (): void => {

    return true; // Error: El tipo 'boolean' no se puede asignar al tipo 'void'.

}