GENÉRICOS

GENÉRICOS

enlace a la documentación

¿Por qué usar **genéricos** y no tipo "**any**"? Mira la respuesta aquí

Fundamentos TS

- Tipos paramétricos <X> en funciones, clases, interfaces

```
function add<T>(arg1: T, arg2: T): T { return arg1 + arg2; }
let result: number = add<number>(2, 3);
```

```
class Box<T> {
 private data: T;
  get(): T { return this.data; }
  set(val: T) { this.data = val; }
}
let b: Box<string> = new Box<string>();
b.set('hola');
```

```
interface Comparable<T> { biggerThan(val: T): boolean }
class MyNumber implements Comparable<number> {
  constructor (private num: number) { }
  biggerThan(val: number): boolean { return this.num > val.num; }
}
let num: Comparable<number> = new MyNumber(10);
console.log(num.biggerThan(5));   // true
```