

02 – TYPESCRIPT

RECUERDA GRABAR LA CLASE

AGENDA

- Types e Interfaces
- Genéricos
- Predicates & Type Guards
- Pick & Partial
- Clases

TYPES E INTERFACES

TYPE

Un type puede ser usado para describir o abstraer objetos / entidades

```
type User =  
  id: number | string  
  username: string  
  location: Location  
}
```

INTERFACES

Una interfaz puede ser usado para describir o abstraer objetos / entidades

```
interface User
  id: number | {string
  username: string
  location: Location
}
```

SIMILITUDES

- Ambas sirven para representar entidades o crear objetos complejos
- Ambas soportan extensión
- Son compatibles: Puedo extender un type a una intefaz y viceversa

DIFERENCIAS

- La interfaz admite sobrecarga
- La interfaz se usa para implementación en clases
- Diferencia de sintaxis para extensión
- El type se usa para definir tipo a valores primitivos: `type MiNumero = number;`

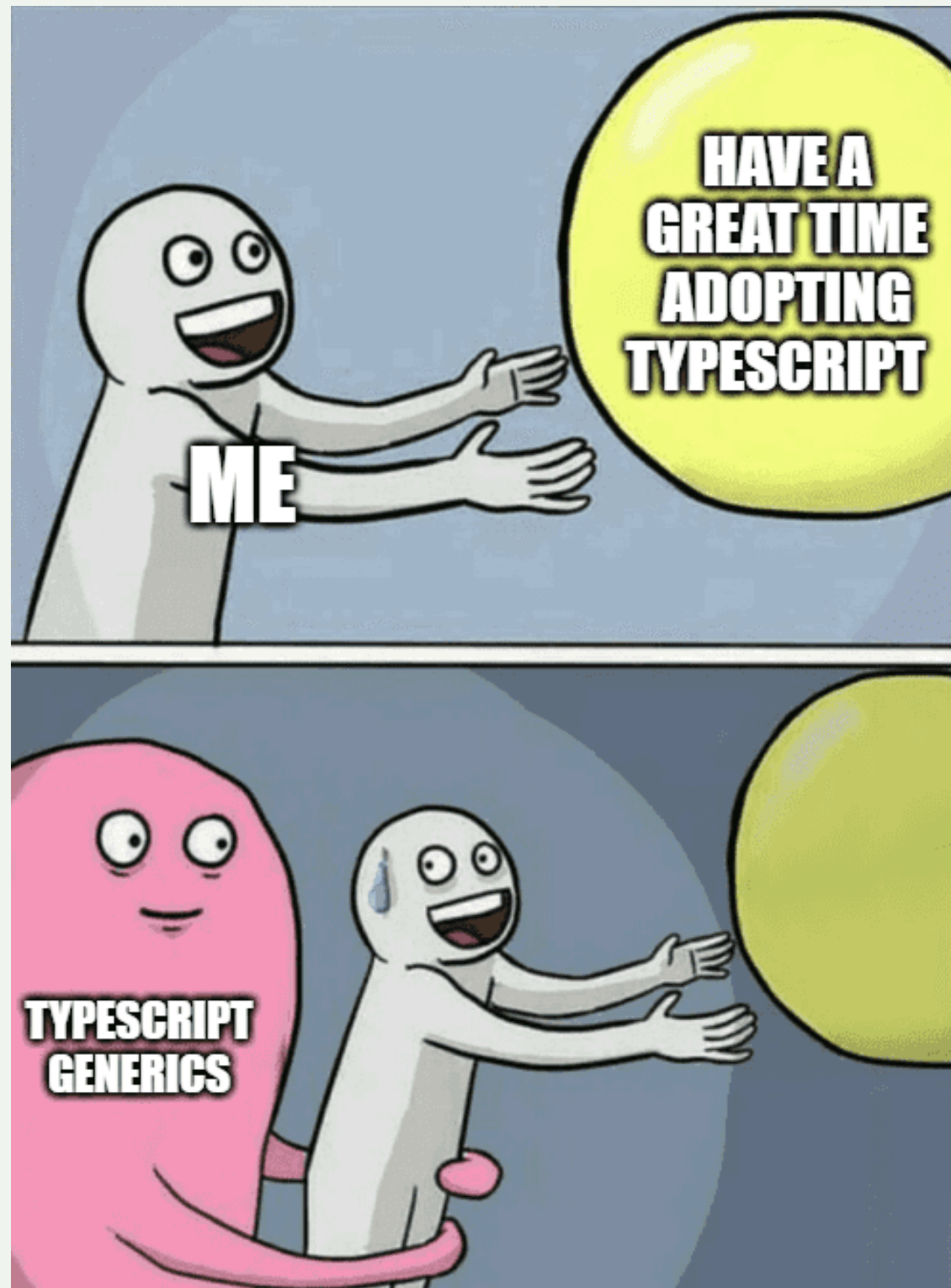


GENÉRICOS

Son plantillas de código. Sirve para aprovechar código sin tener que duplicarlo por causa de cambio de tipos y evitando el any

```
function display(valor: number): number {
  console.log(valor);
  return valor;
}

function displayString(valor: string): string {
  console.log(valor);
  return valor;
}
```



UTILITY TYPES



PARTIAL

Cambia todas las propiedades de un objeto a opcionales

```
type User = {  
  name: string  
  password: string  
  address: string  
  phone: string  
}  
  
type PartialUser = {  
  name?: string  
  password?: string  
  address?: string  
  phone?: string  
}
```

REQUIRED

Cambia todas las propiedades de un objeto a obligatorias

OMIT

Elimina las keys sobre un objeto

```
interface Todo {  
  title: string  
  description: string  
  completed: boolean  
}  
  
type TodoPreview = Omit<Todo, 'description'>  
  
const todo: TodoPreview = {  
  title: 'Clean room',  
  completed: false  
}
```

RETURN TYPE

```
function greet(): string {  
    return 'Hello'  
}  
  
type Greeting = ReturnType<typeof greet>
```

GREETING = STRING

A CODEAR