

Tipos Avanzados y Funciones en TypeScript

by @nicobytes





Nicolas Molina

@nicobytes

Google Developer Expert
Dev and Teacher at **Platzi**



TypeScript

ESNext

JS

Proyecto

Configuración del proyecto con ts-node



Enums

Define un conjunto de variables



Tuplas

Un array fuertemente tipado



```
const [state, setState] = useState(initialState);
```

A yellow square logo with the letters 'JS' in black, representing JavaScript.

```
const [state, setState] = useState<string>('');
```



Unknown type

Una mejor forma de evitar any



Never type

Funciones que nunca terminan



Parámetros opcionales

Y nullish-coalescing



Parámetros por defecto

Al modo TypeScript



Parámetros rest

Flexibilidad en los parámetros



Sobrecarga de funciones

Ayudando a TypeScript en las funciones




```
/* WRONG */
```

```
declare function fn(x: unknown): unknown;  
declare function fn(x: HTMLElement): number;  
declare function fn(x: HTMLDivElement): string;  
var myElem: HTMLDivElement;  
var x = fn(myElem); // x: unknown, wat?
```



```
/* OK */
```

```
declare function fn(x: HTMLDivElement): string;
```

```
declare function fn(x: HTMLElement): number;
```

```
declare function fn(x: unknown): unknown;
```

```
var myElem: HTMLDivElement;
```

```
var x = fn(myElem); // x: string, :)
```



```
/* WRONG */
```

```
interface Example {  
    diff(one: string): number;  
    diff(one: string, two: string): number;  
    diff(one: string, two: string, three: boolean): number;  
}
```



```
/* OK */  
interface Example {  
    diff(one: string, two?: string, three?: boolean): number;  
}
```



```
/* WRONG */
```

```
interface Moment {  
    utcOffset(): number;  
    utcOffset(b: number): Moment;  
    utcOffset(b: string): Moment;  
}
```



```
/* OK */
```

```
interface Moment {  
    utcOffset(): number;  
    utcOffset(b: number | string): Moment;  
}
```



Interfaces

Una mejor forma de definir objetos



Estructuras complejas

Creando nuestra app



Extender interfaces

Aplicando el concepto de herencia



Propiedades de solo lectura

Una forma de evitar sobrescrituras



Creando un CRUD

Preparando nuestro proyecto



Type: Omit & Pick

Nuestros DTOs para creación




```
{  
  "id": "c6f7d3f2",  
  "title": "My Product",  
  "price": 100,  
  "createdAt": "2022-02-03T20:50:53"  
  ...  
}
```

```
{  
  "title": "My Product",  
  "price": 100,  
  ...  
}
```

```
{  
  "id": "c6f7d3f2",  
  "title": "My Product",  
  "price": 100,  
  "createdAt": "2022-02-03T20:50:53"  
  ...  
}
```

Type: Partial & Required

Nuestros DTOs para actualización



Readonly Type

Nuestros DTOs para búsqueda



Acceder al tipado por índice

Manteniendo la consistencia



ReadonlyArray

Evitar las mutaciones en los arrays

