Partial / Record

Antonio Espín Herranz

Partial

- **Partial** es una utilidad que transforma un tipo en un tipo donde todas sus propiedades son opcionales.
- Esto es útil si necesitas trabajar con objetos que pueden no tener todas las propiedades definidas.

```
type Partial<T> = {
   [P in keyof T]?: T[P];
};
```

Ejemplo

```
interface Usuario {
   nombre: string;
   edad: number;
   correo: string;
}

const usuarioParcial: Partial<Usuario> = {
   nombre: “Pedro",
   // Puedes omitir edad y correo porque ahora son opcionales
};

console.log(usuarioParcial);
```

Record

- **Record** permite crear un tipo de objeto cuyos claves y valores están estrictamente definidos.
- Es útil para mapear valores sobre un conjunto de claves conocidas.
- type Record<K extends keyof any, T> = {
- [P in K]: T;
- };

Ejemplo

```
type Estado = "activo" | "inactivo" | "pendiente";

const usuariosEstado: Record<Estado, string[]> = {
    activo: ["Antonio", "Maria"],
    inactivo: ["Juan"],
    pendiente: ["Ana"],
};
```

- console.log(usuariosEstado.activo); // ["Antonio", "Maria"]

Record

- Usa Record cuando quieras crear un objeto donde conoces las claves (por ejemplo, estados, identificadores, etc.) y necesitas asignar valores específicos a esas claves.

Comparación

Característica	`Partial`	`Record`
Transformación de tipo	Convierte todas las propiedades de un tipo en opcionales.	Crea un objeto con claves estrictas y valores definidos.
Propósito principal	Trabajar con tipos parciales o incompletos.	Mapear valores sobre un conjunto de claves específicas.
Ejemplo común	Actualizar parcialmente un objeto.	Crear mapas de estado o configuración.

Combinados

```
type UsuarioParcial = Partial<Record<"nombre" | "edad", string>>;
const usuario: UsuarioParcial = {
    nombre: "Antonio",
    // edad es opcional
};
```