




Objetos y tipos personalizados en TypeScript



Objetos literales

Para definir un objeto literal en TypeScript basta con declarar una variable e inicializarla con llaves { }, y dentro de ellas agregar las propiedades y sus respectivos valores o métodos.



```
1  let task = {  
2      description : "Complete the TypeScript course",  
3      reference: "Typescript for beginners",  
4      completed: false,  
5  };
```

Una vez declaramos el objeto TypeScript le asigna el valor como si fuera un tipo de dato.

*Esto quiere decir que si quiero reasignarle valores a la variable, tengo que declarar de nuevo el objeto y las propiedades. Por ejemplo para cambiar valores a la variable **task**:*



```
1 task = {  
2     description : "Complete the JavaScript Course",  
3     reference : "JavaScript Course",  
4     completed : true,  
5 }
```

¿Cómo crear objetos con tipos específicos?

Para crear objetos con un tipo de dato específico basta con declarar nuestra variable, seguido de eso poner dos puntos y unas llaves, por ejemplo:

`"let user : { }"`

Y dentro de las llaves poner las propiedades y el tipo de dato que quiero que tengan:

`"let user : { name : string, nickname: string }"`

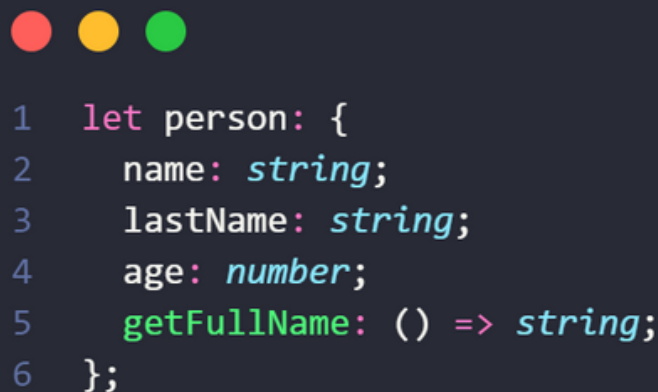
Una vez teniendo esto, ya puedo asignarle valores a las propiedades de mi objeto.



```
1  let user : {name: string, nickname : string } = {  
2      name : "Juan Pérez",  
3      nickname : "@juan_perez",  
4  }
```

Métodos dentro de los objetos

Un método es una función la cual es propiedad de un objeto, para indicarle a nuestro objeto que tendrá un método basta con agregárselo dentro de nuestras llaves que indican el tipo de dato:



```
1 let person: {  
2     name: string;  
3     lastName: string;  
4     age: number;  
5     getFullName: () => string;  
6 };
```

*En este ejemplo el método es la función **getFullName**, la cual retorna un **string** y no recibe argumentos.*

Por lo que ahora ya puedo asignarle valores a las propiedades de mi objeto:




```
1  person = {  
2    name: "Juan",  
3    lastName: "Pérez",  
4    age: 21,  
5    getFullName() {  
6      return `${this.name} ${this.lastName}`;  
7    },  
8  };
```

Tipos personalizados

*En TypeScript puedo crear mis propios tipos de datos para definir objetos, solo con utilizar la palabra reservada **type**, en seguida coloco el nombre que le quiero asignar. Por ejemplo :*

type User = { }

Y dentro de las llaves ya puedo agregar los valores, las propiedades y métodos de mi objeto.



```
1  type User = {  
2      name : string,  
3      lastName : string,  
4      isPremium : boolean,  
5      getFullName : () => string,  
6  };
```

*Y mediante el tipo que definí, ya puedo crear diferentes objetos utilizando las propiedades y métodos de **User**:*



```
1  let newUser : User = {  
2      name: "Juan",  
3      lastName : "Pérez",  
4      isPremium : false,  
5      getFullName() {  
6          return `${this.name} ${this.lastName}`  
7      }  
8  };
```