Ejercicio: Enumeraciones

5 minutos

Vamos a explorar los distintos tipos de datos disponibles en TypeScript y el impacto que tienen en nuestro código.

El tipo de enumeración

Una incorporación útil al conjunto estándar de tipos de datos de JavaScript es el tipo de enumeración, o `enum`.

Las enumeraciones ofrecen una manera sencilla de trabajar con conjuntos de constantes relacionadas. Un elemento `enum` es un nombre simbólico para un conjunto de valores. Las enumeraciones se tratan como tipos de datos y se pueden usar a fin de crear conjuntos de constantes para su uso con variables y propiedades.

Siempre que un procedimiento acepte un conjunto limitado de variables, considere la posibilidad de usar una enumeración. Las enumeraciones hacen que el código sea más claro y legible, especialmente cuando se usan nombres significativos.

El uso de enumeraciones:

- Permite reducir los errores que provoca la transposición o la escritura incorrecta de números.
- Facilita el cambio de valores en el futuro.
- Facilita la lectura del código, lo que significa que es menos probable que se produzcan errores en él.
- Garantiza la compatibilidad con versiones posteriores. Con las enumeraciones, es menos probable que se produzca un error en el código si en el futuro alguien cambia los valores correspondientes a los nombres de miembro.

Creación de una enumeración

Las enumeraciones permiten especificar una lista de opciones disponibles. Son muy útiles

cuando se tiene un conjunto de valores que puede tomar un tipo de variable determinado. Supongamos que tiene un campo en una base de datos externa denominado **ContractStatus**, que contiene los números 1, 2 o 3, que representan los siguientes estados de contacto: **Permanent**, **Temp** y **Apprentice**. Crearemos una enumeración con estos valores y exploraremos la compatibilidad con TypeScript.

1. Abra el área de juegos y quite cualquier código existente.

2. Escriba el código siguiente para crear un objeto `enum` que represente nuestro escenario:

```
TypeScript

enum ContractStatus {
     Permanent,
     Temp,
     Apprentice
}
```

3. Ahora, declare una variable para un nuevo empleado denominada `employeeStatus` del tipo `ContractStatus` y asígnele el nombre `"Temp"`. Muestre el resultado en la consola.

```
TypeScript

let employeeStatus: ContractStatus = ContractStatus.Temp;
console.log(employeeStatus);
```

4. Seleccione **Run** (Ejecutar). Anote el valor que se muestra en la ventana **Log** (Registro). ¿Qué valor se devuelve?

5. De forma predeterminada, los valores `enum` comienzan con un valor de 0, por lo que `Permanent` es 0, `Temp` es 1 y `Apprentice` es 2. Si quiere que los valores empiecen con un valor diferente, en este caso 1, especifíquelo en la declaración `enum`. Realice las ediciones siguientes para que `enum` empiece los valores en 1.

```
TypeScript

enum ContractStatus {
     Permanent = 1,
     Temp,
     Apprentice
}
```

6. Vuelva a ejecutar el código seleccionando **Run** (Ejecutar). Observe que el valor mostrado es ahora **2**.

7. Para mostrar el nombre asociado al valor de enumeración, podemos usar el indexador proporcionado. Agregue lo siguiente en la parte inferior del código:

   TypeScript

   ```
   console.log(ContractStatus[employeeStatus]);
   ```

8. Ejecute el código. Observe que se muestra el valor **Temp**, que es el nombre asociado al valor de enumeración **2**.

Siguiente unidad: Tipos any y unknown en TypeScript

Continuar >

¿Qué tal lo estamos haciendo? ☆ ☆ ☆ ☆ ☆