

Veja nesse artigo como trabalhar com Enums no TypeScript

O enum é um dos tipos do TypeScript que nos permite declarar um conjunto de valores/constantas pré-definidos. Para aqueles que já tiveram contato com alguma linguagem orientada a objetos como: C#, Java ... etc, já teve a oportunidade de trabalhar com ele.

Agora para aqueles que estão tendo o seu primeiro contato com ele nesse artigo, imagine o seguinte cenário: nós precisamos definir uma variável informando cada dia da semana. Como fazer isso de uma maneira simples?

Utilizando o *enum* nós podemos criar uma estrutura com os dias da semana como: Segunda, Terça... e um índice para cada um deles: 1,2,3...etc.

Abaixo você tem um exemplo demonstrando um *enum* com os dias da semana:

```
1  export enum DiaDaSemana {
2      Segunda = 1,
3      Terca = 2,
4      Quarta = 3,
5      Quinta = 4,
6      Sexta = 5,
7      Sabado = 6,
8      Domingo = 7,
9  }
```

Essa é uma das formas que nós temos de declarar um *enum*. E para acessar esses valores? Nós podemos buscar um registro pela sua declaração ou pelo seu valor. Abaixo você tem um exemplo com algumas das formas de se acessar esses valores.

```
let dia = DiaDaSemana[1]; // Segunda
let diaNumero = DiaDaSemana[dia]; // 1
let diaString= DiaDaSemana["Segunda"]; // 1
```

Para listar todos valores de um enum nós podemos utilizar o *for*:

```
for (let dia in DiaDaSemana) {
    if (DiaDaSemana.hasOwnProperty(dia) &&
        (isNaN(parseInt(dia)))) {
        console.log(dia);
    }
}
```

```
    }  
}  
Resultado:  
Segunda  
Terça  
Quarta  
Quinta  
Sexta  
Sabado  
Domingo
```

Bom, agora que estamos alinhados, vamos aprofundar um pouco mais nesse assunto. Nós podemos trabalhar com 3 tipos de *enums*:

- Numéricos
- Texto (String)
- Heterogeneous

Numérico

Os enums *numéricos* armazenam strings com valores numéricos. Esse seria o mesmo exemplo anterior que demonstrei com os dias da semana. Para ficar mais claro, vamos criar novo exemplo numérico:

```
1  export enum Cores {  
2      Azul,  
3      Amarelo,  
4      Verde,  
5      Vermelho  
6  }
```

Note que nesse *enum* acima nós não declaramos os seus valores. Os enums numéricos são auto increment, no nosso exemplo anterior será passado o valor 0 para o primeiro registro e incremental para os próximos.

```
let cor = Cores[1]; // Amarelo  
let corNumero = Cores[cor]; // 1  
let corString= Cores["Vermelho"]; // 3
```

Ele também nos permite adicionar um valor maior que 0 fazer um incremental a

partir dele:

```
1 export enum Cores {
2     Azul = 15,
3     Amarelo,
4     Verde,
5     Vermelho
6 }
```

Resultado:

```
console.log(Cores.Azul) // 15
console.log(Cores.Amarelo) // 16
console.log(Cores.Verde) // 17
console.log(Cores.Vermelho) //18
```

Nós também podemos passar valores aleatórios:

```
1 export enum Cores {
2     Azul = 15,
3     Amarelo = 25,
4     Verde = 33,
5     Vermelho = 48
6 }
```

Resultado:

```
console.log(Cores.Azul) // 15
console.log(Cores.Amarelo) // 25
console.log(Cores.Verde) // 33
console.log(Cores.Vermelho) //48
```

String

Os enums *strings* são similares aos numéricos, a diferença que com os strings nós precisamos inicializar ele com uma string e os numéricos são auto increment como nós vimos no exemplo anterior.

```
1 export enum DiaDaSemana {
2     Segunda = "Segunda-feira",
3     Terca = "Terça-feira",
4     Quarta = "Quarta-feira",
5     Quinta = "Quinta-feira",
6     Sexta = "Sexta-feira",
7     Sabado = "Sábado",
8     Domingo = "Domingo",
9 }
```

Acessando esses valores:

```
console.log(DiaDaSemana.Sexta); //Sexta-feira
console.log(DiaDaSemana['Sabado']); //Sábado
```

Heterogeneous

Os *enums Heterogeneous* aceitam os dois valores dos exemplos anteriores: strings e números.

```
1 export enum Heterogeneous {
2     Segunda = 'Segunda-feira',
3     Terca = 1,
4     Quarta,
5     Quinta,
6     Sexta,
7     Sabado,
8     Domingo = 18,
9 }
```

Como nós vimos nos outros exemplos, os *enums* numéricos são auto increment e aceitam um valor aleatório e os strings retornam o valor da key. Abaixo você tem um exemplo de como acessar os valores do enum *Heterogeneous.ts*.

```
1 import { Heterogeneous } from "./enum/heterogeneous";
2
3
4 console.log(Heterogeneous.Segunda); //Segunda-feira
5 console.log(Heterogeneous['Segunda']); //Segunda-feira
6
7 console.log(Heterogeneous['Terca']); //1
8 console.log(Heterogeneous[1]); //Terca
9
10 console.log(Heterogeneous['Quarta']); //2
11 console.log(Heterogeneous['Quinta']); //3
12 console.log(Heterogeneous['Sexta']); //4
13 console.log(Heterogeneous['Sabado']); //5
14 console.log(Heterogeneous['Domingo']); //18
```

Bem legal né?

Os enums são muito utilizados em cenários onde a mudança dos dados não é constante ou não mudam como: cadastro de sexo, dias da semana, cadastro de redes sociais ... etc.

Bom, a ideia desse artigo era de ser algo simples e rápido com foco em demonstrar como nós podemos utilizar o Enum nos nossos códigos .ts sem medo ☐

Espero que tenham gostado e até um próximo artigo pessoal ☐