

Téc em Desenvolvimento
de Sistemas Bilíngue

Desenvolver Código

Orientado a Objetos

UC4 | Prof. Leonardo de Souza

Estrutura de Dados

ENUMS

O que são

- Enum é uma abreviação de "enumeration", que em português seria "enumeração". Basicamente, é uma maneira de nomear um conjunto de constantes relacionadas. Em TypeScript, um enum é uma maneira de declarar um conjunto fixo de valores que têm um significado específico em um contexto.

O que são

- São como uma lista de palavras-chave que você pode usar para representar valores específicos em seu código. Cada palavra-chave tem um valor associado a ela.

Estrutura de Dados

Características

Características

- **Palavras-chave Descritivas:** Os enumeradores permitem usar palavras-chave significativas para representar valores específicos.

Características

- **Valores Automáticos ou Personalizados:** Os valores dos enumeradores podem ser atribuídos automaticamente começando de 0 ou podem ser definidos manualmente para cada palavra-chave

Características

- Suporte a Valores Numéricos e de Texto: Os enumeradores podem conter tanto valores numéricos quanto de texto.

Características

- **Fácil de Iterar:** Você pode facilmente percorrer todas as palavras-chave de um enumerador usando loops ou métodos de iteração.

Estrutura de Dados

Sintaxe

Sintaxe

- Para criar um enumerador em TypeScript, você usa a palavra-chave `enum` seguida por uma lista de palavras-chave separadas por vírgulas.

```
enum NomeDoEnum {  
    PalavraChave1,  
    PalavraChave2,  
    PalavraChave3,  
}
```

Estrutura de Dados

Exemplos

Exemplos

- Vamos ver alguns exemplos para entender melhor como os enumeradores funcionam:

```
enum DiasDaSemana {  
    Segunda,  
    Terça,  
    Quarta,  
    Quinta,  
    Sexta,  
    Sábado,  
    Domingo,  
}
```

```
console.log(DiasDaSemana.Segunda); // Saída: 0
```



```
enum Cores {  
    Vermelho = "#FF0000",  
    Verde = "#00FF00",  
    Azul = "#0000FF",  
}  
  
console.log(Cores.Verde); // Saída: "#00FF00"
```

```
enum Direcoes {
```

```
    Norte,
```

```
    Sul,
```

```
    Leste,
```

```
    Oeste,
```

```
}
```

```
for (let direcao in Direcoes) {
```

```
    console.log(direcao);
```

```
}
```

Atenção

- Quando criamos um enum, damos um nome a ele, e as palavras-chave são as constantes. Elas nunca mudam de valor, por isso chamamos elas de constante. Elas também devem ter algum tipo de significado em comum para a aplicação.

Atenção

- Os valores das constantes podem ser declarados das seguintes formas:

```
//Numérico e implícito:  
export enum Level {  
  BLUE, // 0  
  YELLOW, // 1  
  ORANGE, // 2  
  RED // 3  
}
```

```
//Numérico e explícito  
export enum Level {  
  BLUE = 0,  
  YELLOW = 10,  
  ORANGE = 20,  
  RED = 30  
}
```

```
//Numérico e explícito incremental  
export enum Level {  
  BLUE = 1, // 1  
  YELLOW, // 2  
  ORANGE, // 3  
  RED // 4  
}
```



```
//String:  
enum Level {  
    BLUE = "Blue",  
    YELLOW = "Yellow",  
    ORANGE = "Orange",  
    RED = "Red"  
}
```

Quando usar ENUMS?

- Você pode usar enums sempre que tiver um conjunto fixo e limitado de valores relacionados que precisam ser representados de forma clara e concisa no seu código. Eles são úteis em situações como status, opções de menu, dias da semana, meses, e assim por diante.

Estrutura de Dados

Exercício

Enum Level

Desenvolva um sistema para uma pizzeria em TypeScript. Você deve definir um enum chamado SaborPizza que representará os sabores disponíveis de pizzas.

Crie uma classe chamada Pizza que vai ter os parâmetros sabor, tamanho e preço. Crie nesta classe um método chamado descrição que retorna uma string contendo a descrição da pizza no formato "Pizza [sabor], Tamanho: [tamanho], Preço: R\$ [preco]".

Crie três instâncias de pizzas com diferentes sabores, tamanhos e preços e exiba no console.