



Save it



Like it



Typescript

Enums



@coder_aishya



What is an enum ?

- Enums are a new data type supported in TypeScript.
- An enum is a group of named constant values.
- Enum stands for enumerated type.

Types of Enum

- Numeric Enum
- String Enum



Numeric Enum

- Numeric enums are number-based enums i.e. they store string values as numbers.
- Enums can be defined using the keyword `enum`.

Enum to store a set of print media types is:



```
enum PrintMedia {  
    Newspaper, —————> = 0  
    Newsletter, —————> = 1  
    Magazine, —————> = 2  
    Book —————> = 3  
}
```



We also have the option to initialize the first numeric value ourselves. For example, we can write the same enum as:



```
enum PrintMedia {  
    Newspaper = 1, —————> = 1  
    Newsletter, —————> = 2  
    Magazine, —————> = 3  
    Book —————> = 4  
}
```



The enum can be used as a function parameter or return type, as shown below:

```
enum PrintMedia {  
    Newspaper = 1,  
    Newsletter,  
    Magazine,  
    Book  
}  
  
function getMedia(mediaName: string): PrintMedia {  
    if ( mediaName === 'Forbes' || mediaName === 'Outlook') {  
        return PrintMedia.Magazine;  
    }  
}  
  
let mediaType: PrintMedia = getMedia('Forbes');  
// returns Magazine
```



String Enum

String enums are similar to numeric enums, except that the enum values are initialized with string values rather than numeric values.

String enum to store print media is :



```
enum PrintMedia {  
    Newspaper = "NEWSPAPER",  
    Newsletter = "NEWSLETTER",  
    Magazine = "MAGAZINE",  
    Book = "BOOK"  
}  
  
// Access String Enum  
PrintMedia.Newspaper; //returns NEWSPAPER  
PrintMedia['Magazine']; //returns MAGAZINE
```



@coder_aishya



String Enum vs Numeric Enum

The difference between numeric and string enums is that numeric enum values are auto-incremented, while string enum values need to be individually initialized.



@coder_aishya

