

# Enum in Typescript

---

## AGENDA

1. Number enum
  2. String enum
  3. When to use enum
  4. Bonus: How enum compiled to Javascript
- 

- **What:** Tập hợp các giá trị cùng nhóm.
- **Why:** Dễ dàng quản lý và truy xuất.
- **When:** Sử dụng cho data một chiều, định nghĩa enum --> rồi sử dụng. (chi tiết bên dưới)

## 1. Number enum

```
enum Status {  
  PENDING,      // 0  
  IN_PROGRESS,  // 1  
  DONE,         // 2  
  CANCELLED,    // 3  
}
```

```
enum Status {  
  PENDING = 3,   // 3  
  IN_PROGRESS,   // 4  
  DONE,         // 5  
  CANCELLED,    // 6  
}
```

```
enum Status {  
  PENDING = 3,  
  IN_PROGRESS = 5,  
  DONE = 8,  
  CANCELLED = 10,  
}
```

```
enum Status {  
  PENDING,    // 0  
  IN_PROGRESS, // 1  
  DONE,       // 2  
  CANCELLED,  // 3  
}  
  
// can assign any number to your enum variable  
const stats1: Status = Status.PENDING;  
const stats2: Status = 1;  
const stats3: Status = -1;  
  
// number enum --> support reverse mapping  
console.log(Status[0]); // 'PENDING'  
console.log(Status['DONE']); // 2
```

## 2. String enum

- Don't have reverse mapping for string enum.

```
enum MediaTypes {  
  JSON = "application/json",  
  XML = "application/xml",  
}  
  
fetch("https://example.com/api/endpoint", {  
  headers: {  
    Accept: MediaTypes.JSON,  
  },  
}).then(response => {  
  // ...  
});
```

Source: <https://mariusschulz.com/blog/string-enums-in-typescript>

## 3. When to use enum

- Static data on Frontend, not the data from API response.
- Not required to reverse map.
- Prefer **union types** for data returned from API.

## 4. Bonus: How enum compiled to Javascript

- Playground: [link](#)

```
enum Status {
  PENDING,      // 0
  IN_PROGRESS,  // 1
  DONE,         // 2
  CANCELLED,    // 3
}

// compiled to javascript
"use strict";
var Status;
(function (Status) {
  Status[Status["PENDING"] = 0] = "PENDING";
  Status[Status["IN_PROGRESS"] = 1] = "IN_PROGRESS";
  Status[Status["DONE"] = 2] = "DONE";
  Status[Status["CANCELLED"] = 3] = "CANCELLED";
})(Status || (Status = {}));

Status['PENDING']; // 0
Status[0]; // 'PENDING'
```

```
enum MediaTypes {
  JSON = "application/json",
  XML = "application/xml",
}

// compiled to javascript
"use strict";
var MediaTypes;
(function (MediaTypes) {
  MediaTypes["JSON"] = "application/json";
  MediaTypes["XML"] = "application/xml";
})(MediaTypes || (MediaTypes = {}));

MediaTypes["JSON"]; // "application/json"
MediaTypes["application/json"]; // undefined
```

## Tham khảo

- <https://mariusschulz.com/blog/string-enums-in-typescript>
- <https://blog.logrocket.com/writing-readable-code-with-typescript-enums-a84864f340e9/>

---

## Series - Typescript cơ bản 🎉

- Tác giả: **Hậu Nguyễn**
- Được phát hành trên kênh youtube **Easy Frontend**.
- Tài liệu pdf và videos đều có bản quyền thuộc về Easy Frontend.
- Videos được phát hành cho fan cứng trước, public sau.
- [Đăng ký fan cứng](#) để xem series này đầy đủ và sớm nhất nhé.

### 📞 Kết nối với mình

- ☒ Follow Facebook: <https://www.facebook.com/nvhauesmn/>
- ☒ Like Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ☒ Youtube Channel: <https://www.youtube.com/easyfrontend>