

[CGA-JAVA-WFDA] Web Front-end Development Angular 2.1

[Dashboard](#) / [My courses](#) / [CGA-JAVA-WFDA-2.1](#) / [2. Typescript](#) / [\[Bài đọc\] Biến và kiểu dữ liệu](#)

[Bài đọc] Biến và kiểu dữ liệu

Biến trong TypeScript

Mọi chương trình JavaScript đều là một chương trình TypeScript

Do đó TypeScript vẫn tuân thủ cách đặt biến giống như JavaScript.

Biến trong TypeScript vẫn được khai báo bởi các từ khoá như var, let và const.

Kiểu dữ liệu trong TypeScript

TypeScript hỗ trợ tất cả các kiểu dữ liệu của JavaScript và bổ sung thêm một số kiểu dữ liệu mới để hỗ trợ các lập trình viên làm việc dễ dàng hơn. Và dưới đây là một số kiểu dữ liệu mới trong TypeScript:

Boolean

Với những dữ liệu có giá trị trả về là true/false thì TypeScript cung cấp cho ta kiểu dữ liệu boolean để lưu trữ chúng.

Ví dụ: `let isShow: boolean = true;`

Number

Cũng giống như JavaScript tất cả số trong TypeScript đều là số thực và được lưu trữ với kiểu dữ liệu là number. Ngoài ra các lập trình viên cũng có thể khai báo dữ liệu dạng số thập phân, hệ thập lục phân, hệ nhị phân và hệ bát phân.

Ví dụ:

```
let decimal: number = 10;
```

```
let hex: number = 0xf00d;
```

```
let binary: number = 0b1010;
```

```
let octal: number = 0o744;
```

String

Để lưu trữ dữ liệu dạng chuỗi trong TypeScript lập trình viên có thể khai báo kiểu dữ liệu string

```
let x = 'CodeGym'
```

Array

Cũng như JavaScript, Typescript cho phép làm việc với mảng các giá trị. Array có thể được khai báo bằng hai cách sau:

```
let list: number[] = [1, 2, 3, 4, 5];
```

```
let list: Array<number> = [1, 2, 3, 4, 5];
```

Tuple

Kiểu dữ liệu Tuple cho phép khai báo mảng với các giá trị có kiểu dữ liệu đã biết ví dụ như sau:

```
// Khai báo kiểu dữ liệu Tuple
```

```
let x: [string, number];
```

```
// Khởi tạo dữ liệu cho kiểu Tuple
```

```
x = ["CodeGym", 2021]; // Đúng
```

```
x = [2021, "CodeGym"]; // Sai
```

Từ ví dụ trên ta thấy rằng để sử dụng kiểu dữ liệu Tuple cần lưu ý việc khởi tạo các dữ liệu theo đúng thứ tự kiểu dữ liệu đã khai báo.

Enum

TypeScript cũng cung cấp thêm một kiểu mới đó là Enum, Enum được sử dụng để cho phép một biến có thể là một tập hợp các hằng số được định sẵn.

Ví dụ:

```
enum City {HaNoi, HoChiMinh, DaNang};
```

```
let city: City = City.HaNoi;
```

Any

TypeScript còn cung cấp một kiểu dữ liệu any giúp các lập trình viên có thể giải quyết được vấn đề không biết chắc chắn kiểu dữ liệu của nó.

Ví dụ:

```
let x: any = 2021;

x = 'CodeGym';

x = true;

x = {

  id: 1,

  name: 'CodeGym'

};
```

Last modified: Saturday, 20 February 2021, 11:51 AM

Previous Activity

Jump to...

Next Activity

CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

Follow Us

