

Tổng quan TypeScript - Cài đặt môi trường

TypeScript là gì?

TypeScript là một ngôn ngữ lập trình mã nguồn mở do Microsoft phát triển, mở rộng từ JavaScript bằng cách thêm vào tính năng kiểu tĩnh (static typing). Điều này giúp phát hiện lỗi ngay trong quá trình viết code thay vì chờ đến lúc runtime. TypeScript biên dịch (transpile) xuống JavaScript, do đó có thể chạy trên bất kỳ trình duyệt, máy chủ hay ứng dụng nào hỗ trợ JavaScript.

Tổng quan về TypeScript

- **Kiểu tĩnh (Static Typing):** TypeScript cho phép khai báo kiểu dữ liệu cho các biến, tham số hàm, và giá trị trả về. Điều này giúp ngăn chặn nhiều lỗi do không tương thích kiểu dữ liệu.
- **Interface:** Giúp định nghĩa cấu trúc của đối tượng, hỗ trợ việc kiểm tra kiểu dữ liệu một cách chặt chẽ hơn.
- **Enum:** Cung cấp một cách để định nghĩa các hằng số dễ đọc hơn.
- **Generics:** Cho phép tạo các components có thể hoạt động với nhiều loại dữ liệu khác nhau mà vẫn giữ được tính an toàn về kiểu.
- **Type Inference:** TypeScript có khả năng suy đoán kiểu dữ liệu dựa trên giá trị ban đầu được gán cho biến.
- **Namespaces và Modules:** Hỗ trợ tổ chức code theo cách dễ dàng hơn, tránh xung đột tên và tạo điều kiện thuận lợi cho việc tái sử dụng code.

Ứng dụng thực tế của TypeScript

- **Phát triển ứng dụng web:** TypeScript đặc biệt phổ biến trong việc phát triển các ứng dụng web lớn, nơi mà việc kiểm soát và quản lý code trở nên phức tạp. Ví dụ, nhiều framework như Angular được xây dựng hoàn toàn trên TypeScript.

- **Phát triển ứng dụng di động:** TypeScript được sử dụng trong các framework như React Native để phát triển ứng dụng di động. Với khả năng hỗ trợ kiểu tĩnh, TypeScript giúp việc viết code trở nên chính xác và an toàn hơn.
- **Phát triển backend:** TypeScript cũng có thể được sử dụng để phát triển backend, thường là với Node.js. Điều này giúp đồng bộ hóa giữa code frontend và backend trong cùng một dự án.
- **Tăng cường chất lượng code:** Trong các dự án lớn, TypeScript giúp giảm thiểu lỗi bằng cách bắt lỗi ngay khi viết code, đồng thời cung cấp công cụ tự động hóa như IntelliSense để tăng hiệu suất lập trình.
- **Dự án mã nguồn mở:** Nhiều dự án mã nguồn mở hiện nay đang chuyển sang sử dụng TypeScript, điển hình là các thư viện JavaScript phổ biến như React, Vue, hay D3.
- **Phát triển hệ thống phức tạp:** Với khả năng hỗ trợ các công cụ kiểm thử và phân tích code mạnh mẽ, TypeScript thường được sử dụng trong các hệ thống đòi hỏi độ tin cậy cao, như các ứng dụng tài chính, quản lý chuỗi cung ứng, hay hệ thống y tế.

Cài đặt môi trường chạy TypeScript

Cài đặt NodeJS

Trước tiên, bạn cần cài đặt Node.js vì TypeScript chạy trên nền tảng Node.js. Node.js bao gồm Node Package Manager (NPM), giúp bạn dễ dàng cài đặt TypeScript và các công cụ khác.

- **Tải Node.js:** Truy cập vào trang [Node.js](https://nodejs.org/), tải và cài đặt phiên bản LTS (Long Term Support) mới nhất phù hợp với hệ điều hành của bạn.
- **Kiểm tra cài đặt Node.js và NPM:** Sau khi cài đặt xong, mở terminal hoặc command prompt và chạy các lệnh sau để kiểm tra:

```
node -v
npm -v
```

Nếu hai lệnh trên trả về phiên bản của Node.js và NPM, thì bạn đã cài đặt thành công.

Cài đặt TypeScript

Sử dụng NPM để cài đặt TypeScript trên toàn hệ thống:

```
npm install -g typescript
```

Kiểm tra cài đặt TypeScript: Kiểm tra bằng cách chạy lệnh sau:

```
tsc -v
```

Nếu lệnh này trả về phiên bản của TypeScript, việc cài đặt đã thành công.

Thiết lập Dự Án TypeScript

Tạo thư mục dự án:

```
mkdir my-typescript-project  
cd my-typescript-project
```

Khởi tạo dự án với `package.json`:

```
npm init -y
```

Lệnh này sẽ tạo file `package.json` mặc định cho dự án.

Tạo file cấu hình `tsconfig.json`:

```
tsc --init
```

File `tsconfig.json` giúp cấu hình các tùy chọn biên dịch TypeScript. Bạn có thể mở file này và điều chỉnh các cài đặt như `target` (phiên bản JavaScript đích), `module` (loại module), v.v.

Viết và Biên dịch TypeScript

Tạo file TypeScript

Tạo một file `index.ts` trong thư mục dự án của bạn:

```
const greeting: string = "Hello, TypeScript!";  
console.log(greeting);
```

Biên dịch TypeScript

Biên dịch file

`index.ts` thành JavaScript bằng lệnh sau:

```
tsc
```

Lệnh này sẽ tạo ra file `index.js` trong cùng thư mục.

Chạy file JavaScript:

Bạn có thể chạy file JavaScript vừa biên dịch bằng Node.js:

```
node index.js
```

Sử dụng Watch Mode (Tuỳ chọn)

Để tự động biên dịch khi bạn thay đổi mã nguồn, bạn có thể sử dụng chế độ watch:

```
tsc --watch
```

Chế độ này giúp tiết kiệm thời gian khi phát triển ứng dụng bằng TypeScript.