BỘ CÂU TRẢ LỜI AUDIT MODULE 5

1. TypeScript là gì?

* Là 1 superste của JS. Được phát triển bởi Microsoft
* Có thể chuyển đổi mã nguồn thành code JS để chạy trên mt của Browser hoặc NodeJS
* Tuân thủ chặt chẽ các kỹ thuật mà ECMAScript (ES6) đề ra, do đó tất cả những đoạn code hợp lệ trong JS thì sẽ hợp lệ ở TS
* TypeScript đang được hỗ trợ rất mạnh, từ cộng đồng, IDE/Editor, đến

các library/framework.

1. Tại sao nên sử dụng Typescript, Ưu điểm của Typescript

Bởi vì JS nó k có các quy tắc chặt chẽ, nó có thể khai báo biến tự do, chỉ dùng cho những dự án nhỏ còn TS thì nó tuân thủ những quy tắc chặt chẽ hơn và khi làm nhóm thì sẽ dễ dàng hơn trong việc sửa chửa

\*Ưu điểm:

- Tạo sự nhất quán trong coding style cho team nhiều người

- Support nhiều tính năng nâng cao của ES6

- Dễ dàng gợi ý code trong tool editor/IDE

- Hỗ trợ debug

- Support nhiều kiểu dl mới

1. Làm sao để cài đặt TypeScript

B1: vào nodejs để cài đặt nodejs về máy

B2: Sau khi cài đặt xong vào cmd kiểm tra version của npm và nodejs

Gõ lệnh để kiểm tra version npm như bên dưới : npm –v

Gõ lệnh kiểm tra version nodejs như bên dưới: node –v

B3: Cài đặt typescript: npm install --global typescript

1. Có thể chuyển mã TypeScript thành mã Javascript không, bằng cách nào?

Có thể chuyển mã TypeScript thành mã Javascript.

* Bước 1: Tạo file main.ts, sau đó điền code mẫu ở page sau.
* Bước 2: Di chuyển terminal/cmd vào thư mục chứa file main.ts
* Bước 3: Chạy lệnh tsc main.ts
* Bước 4: Kiểm tra file JS được transpilera.
* Bước 5: Chạy lệnh nodemain.js (js,không phải ts)để nhìn thấy kết quả

1. Kể tên các kiểu dữ liệu trong Typescript

Boolean, Number, String, Array, Tuple, Enum, Any

1. Kể tên các vòng lặp được typescript hỗ trợ

* Trong TypeScript hỗ trợ chúng ta 3 loại vòng lặp for khác nhau lần lượt là for, for of và for in.
* Vòng lặp while và do-while

1. Phân biệt sự khác nhau giữa var, let, const

|  |  |
| --- | --- |
| Var | Let/Const |
| * Function scope * Có thể khai báo 1 biến trong cùng 1 scope với từ khóa var * var thì hoisted lên đầu scope | * Block scope * Không thể khai báo lại biến trong cùng 1 scope với let hoặc const * Khi khai báo biến với let hoặc const thì biến đó không được hoisted lên đầu scope, dẫn đến muốn dùng biến thì bắt buộc phải khai báo trước. |