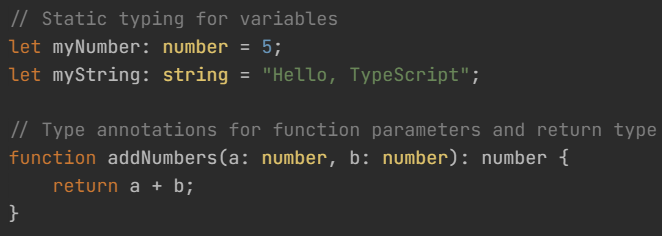
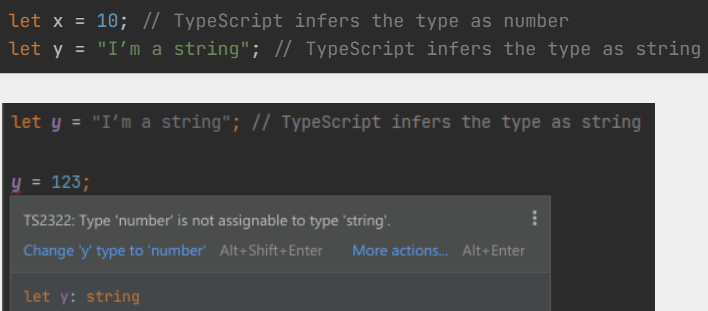
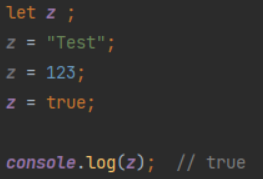
1



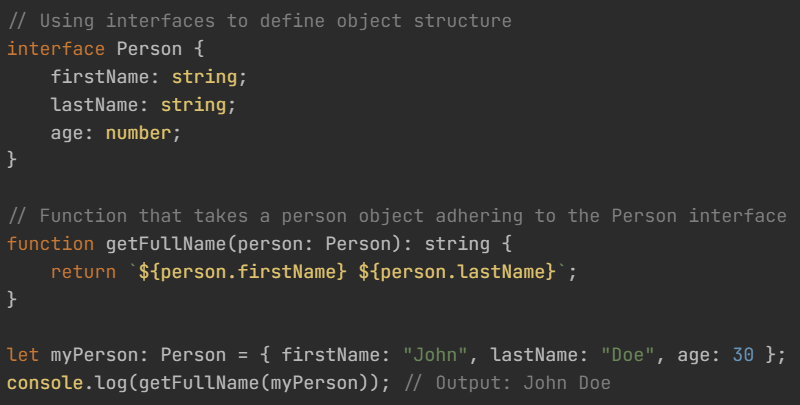
2



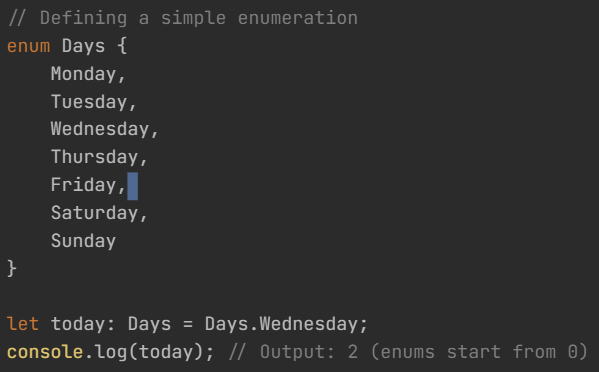
3



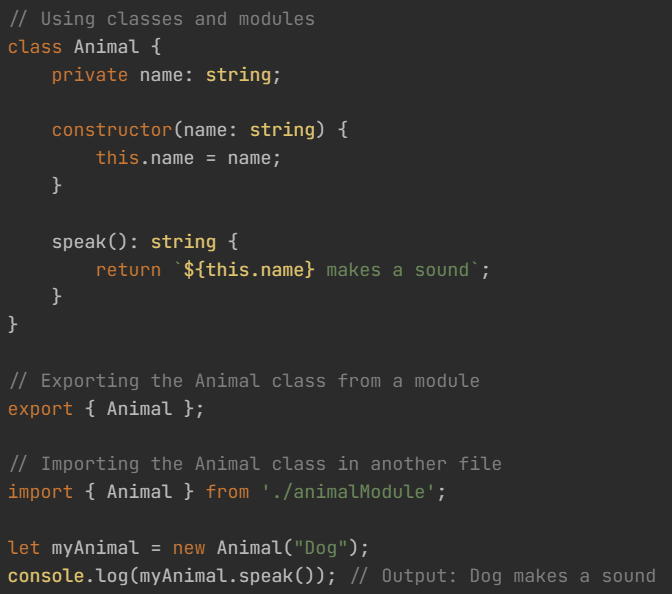
4



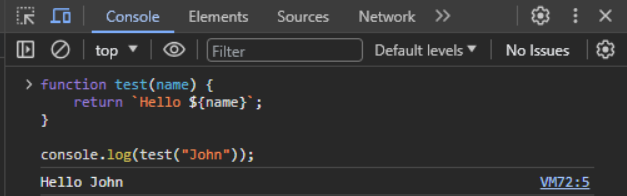
5



6



9



**7. JavaScript vs TypeScript**

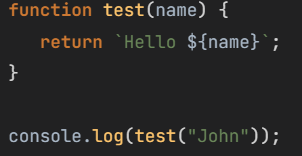
|  |  |  |
| --- | --- | --- |
| **Feature** | **JavaScript** | **TypeScript** |
| **Type System** | Dynamically-typed | Statically-typed (optional static typing) |
| **Platform** | Web browsers, server-side (Node.js) | Primarily for web development (JavaScript runtime) |
| **Type Annotations** | No explicit type annotations | Optional, allows explicit type annotations |
| **Type Inference** | Limited type inference | Powerful type inference system |
| **Interfaces** | No native support | Supports interfaces for defining object structure |
| **Enums** | No native support | Supports enums for named constant values |
| **Classes and Modules** | Supports, but with ES6 (ECMAScript 2015) syntax | Supports, with additional features for object-oriented programming |
| **Compatibility** | Works in any JavaScript runtime | Compatible with JavaScript; TypeScript code can be transpiled to JavaScript |
| **Tooling Support** | Varied; many code editors and IDEs available | Well-supported in Visual Studio Code and other popular editors |
| **Development Process** | Less tooling support for catching errors early | Enhanced tooling support for catching errors during development |
| **Code Structure** | Typically less structured due to dynamic nature | Supports a more structured and modular code organization |
| **Adoption in Frameworks** | Commonly used with frontend frameworks (React, Vue, Angular) | Widely used in Angular development, gaining popularity in other frameworks |

8

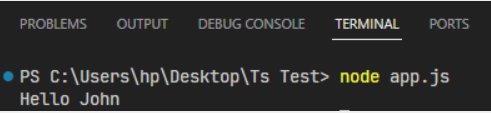
**Java vs TypeScript**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Java** | **TypeScript** |
| **Type System** | Statically-typed | Statically-typed (optional static typing) |
| **Platform** | Cross-platform | Primarily for web development (JavaScript runtime) |
| **Compilation** | Compiled to bytecode (JVM) | Transpiled to JavaScript |
| **Usage** | Enterprise applications, Android, backend | Web development, Node.js, frontend (Angular) |
| **Object-Oriented** | Purely object-oriented | Object-oriented (class-based, prototype-based) |
| **Development Ecosystem** | Mature ecosystem with extensive libraries and frameworks | Benefits from the JavaScript ecosystem, good library support |
| **Tooling** | IDEs like Eclipse, IntelliJ IDEA; Maven, Gradle for builds | Visual Studio Code, lightweight and powerful; good TypeScript support |
| **Memory Management** | Automatic garbage collection | Relies on JavaScript garbage collection |

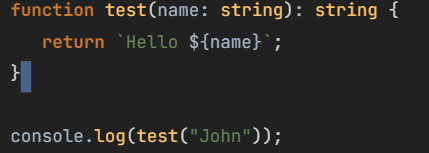
10



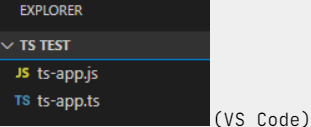
11



12

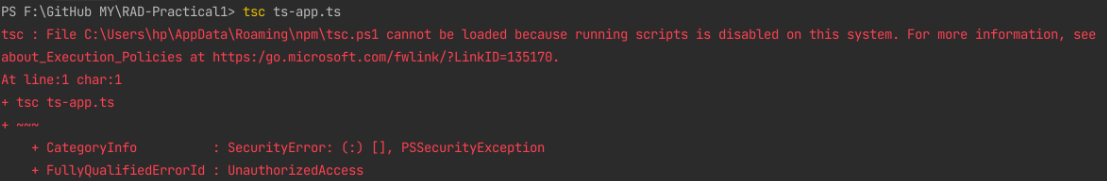


13

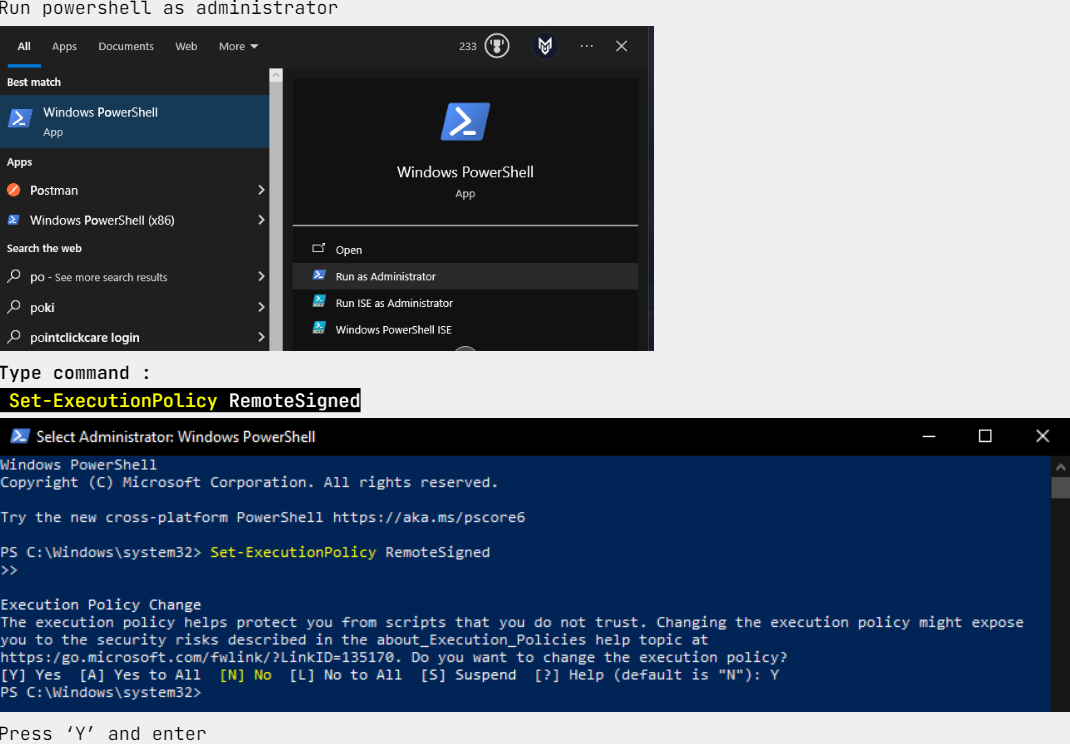


14

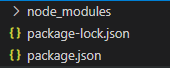
15



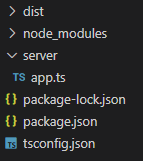
16



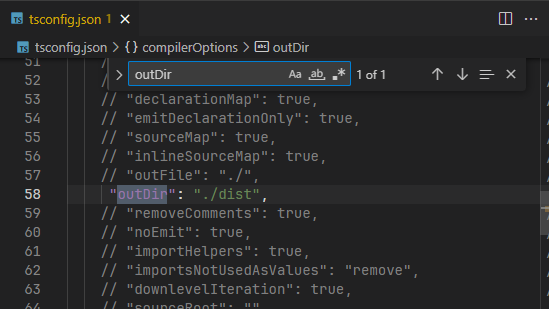
17



18



19

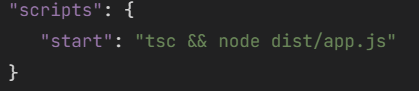


20

A screen shot of a computer program

Description automatically generated

21



22



23