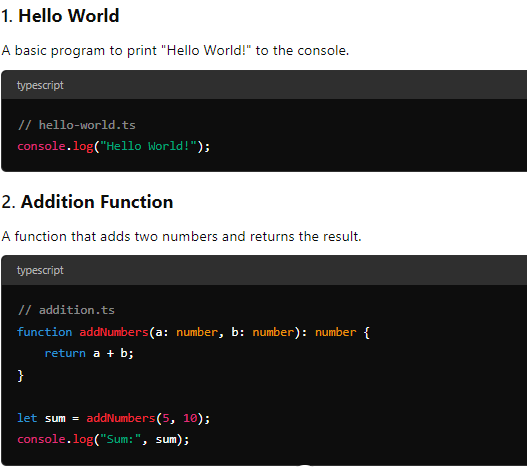
**Installing TypeScript**

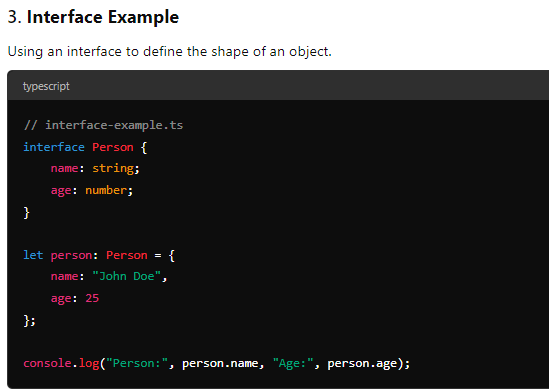
**npm install typescript --save-dev**

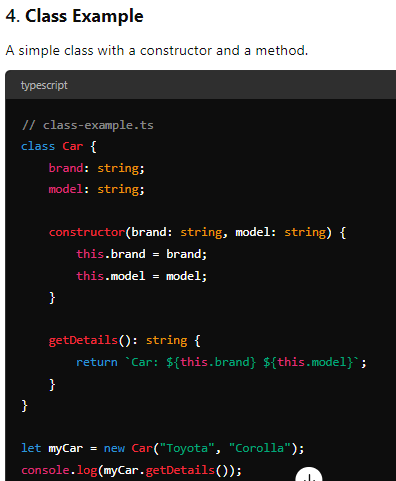
To check the version you are using, you can run the following command:

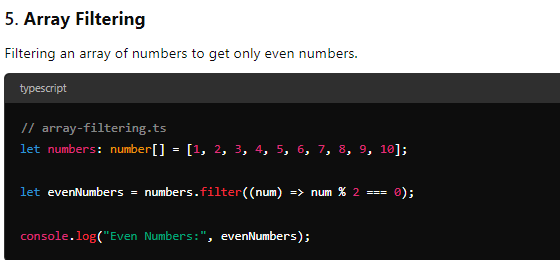
**tsc –v**

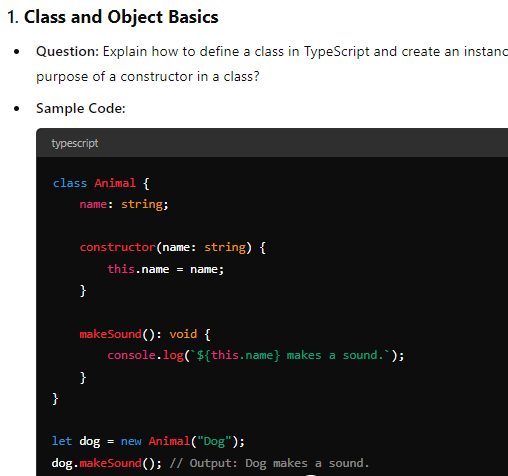
**tsc --init**

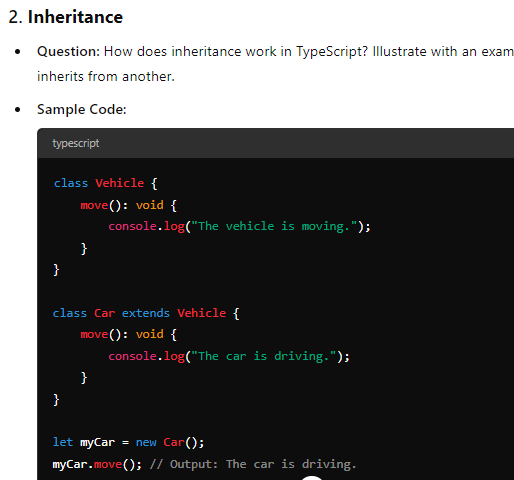




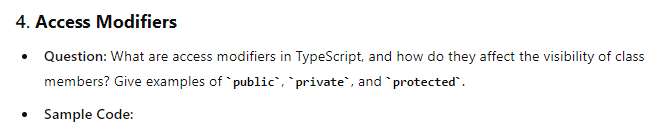










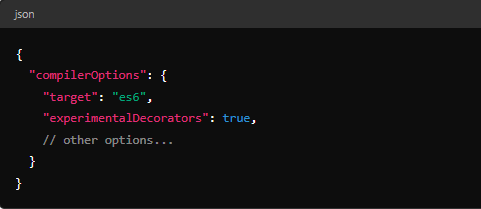




**Decorators** in TypeScript are a special kind of declaration that can be attached to a class, method, accessor, property, or parameter. Decorators allow you to modify the behavior of the item they are attached to, providing a way to add metadata or apply some logic in a declarative manner.

**Enabling Decorators**

To use decorators in TypeScript, you need to enable them in your tsconfig.json by setting the experimentalDecorators option to true.



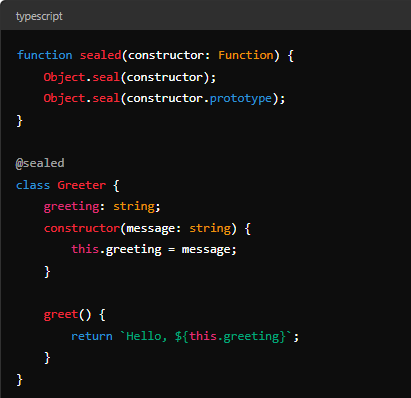
### Types of Decorators

There are several types of decorators you can use:

1. **Class Decorators**
2. **Method Decorators**
3. **Accessor Decorators**
4. **Property Decorators**
5. **Parameter Decorators**

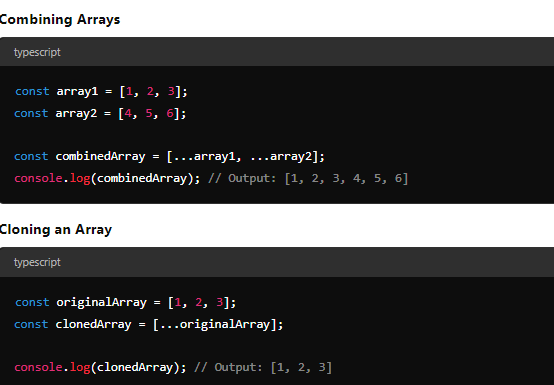
### 1. ****Class Decorators****

A class decorator is a function that takes a class constructor as its only argument and can return a new constructor to replace the original class.

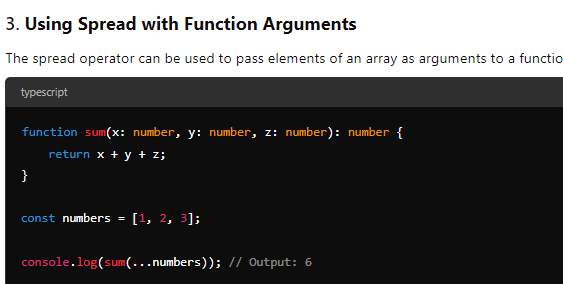


Decorators in TypeScript provide a powerful way to add annotations and modify behavior in a declarative style. They can be used for various purposes, such as adding metadata, logging, validation, or applying cross-cutting concerns like security or caching. Remember that decorators are a TypeScript-specific feature that is not yet part of the JavaScript standard, so using them may require some consideration regarding compatibility and long-term maintenance.

**The spread operator** (...) in TypeScript (and JavaScript) allows you to spread out elements of an array or object into another array or object. It's a very useful tool for combining arrays, cloning objects, and passing arguments to functions.

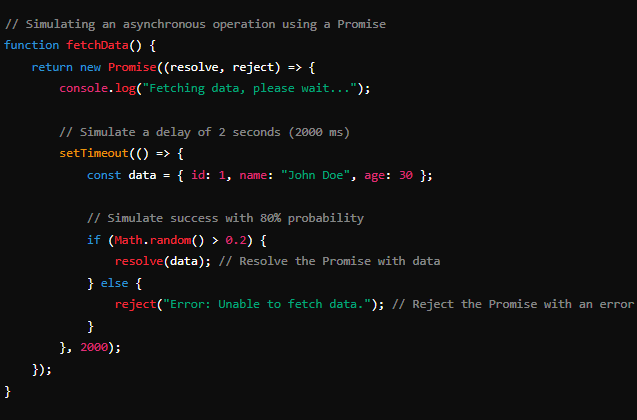


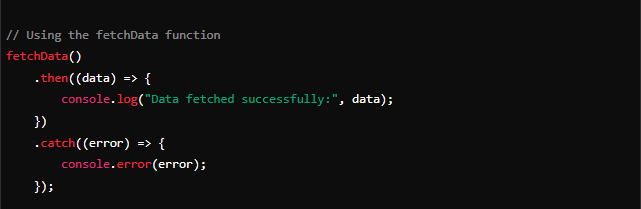


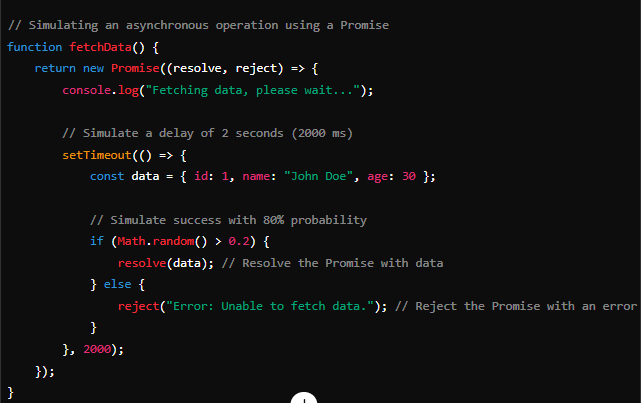


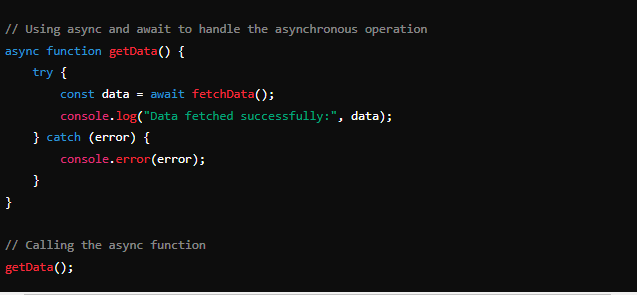
The spread operator in TypeScript is a versatile tool that simplifies working with arrays and objects. It can be used to combine, clone, or merge data structures and is especially useful in function argument passing. Its simplicity and power make it a staple in modern TypeScript (and JavaScript) development.

**Asynchronous Programming in ES6**









Console.log(“start exe”)

setTimeout(()=>{

console.log(“logic exe”)

},2000)

Console.log(“start exe”)

