TS TypeScript TUTORIAL

TypeScript void Type

Summary: in this tutorial, you will learn about the TypeScript `void` type and how to use it as the return type of functions that do not return any value.

Introduction to TypeScript void type

The `void` type denotes the absence of having any type at all. Typically, you use the `void` type as the return type of functions that do not return a value. For example:

```
function log(message): void {
    console.log(messsage);
}
```

It is a good practice to add the `void` type as the return type of a function or a method that doesn't return any value. By doing this, you can gain the following benefits:

- Improve clarity of the code: you do not have to read the whole function body to see if it returns anything.
- Ensure type-safe: you will never assign the function with the `void` return type to a variable.

Notice that if you use the `void` type for a variable, you can only assign `undefined` to that variable. In this case, the `void` type is not useful. For example:

```
let useless: void = undefined;
useless = 1; // error
```

If the `--strictNullChecks` flag is not specified, you can assign the `useless` to `null`.

```
useless = null; // OK if --strictNullChecks is not specified
```

Summary

- Use the `void` type as the return type of functions that do not return any value.