TS TypeScript TUTORIAL

TypeScript Tuple

Summary: in this tutorial, you'll learn about the TypeScript Tuple type and its usage.

Introduction to TypeScript Tuple type

A tuple works like an array with some additional considerations:

- The number of elements in the tuple is fixed.
- The types of elements are known, and need not be the same.

For example, you can use a tuple to represent a value as a pair of a `string` and a `number`:

```
let skill: [string, number];
skill = ['Programming', 5];
```

The order of values in a tuple is important. If you change the order of values of the `skill` tuple to `[5, "Programming"]`, you'll get an error:

```
let skill: [string, number];
skill = [5, 'Programming'];
```

Error:

```
error TS2322: Type 'string' is not assignable to type 'number'.
```

For this reason, it's a good practice to use tuples with data that are related to each other in a specific order.

For example, you can use a tuple to define an RGB color that always comes in a three-number pattern:

```
(r,g,b)
```

For example:

```
let color: [number, number, number] = [255, 0, 0];
```

The `color[0]` , `color[1]` , and `color[2]` would be logically mapped to `Red` , `Green` and `Blue` color values.

Optional Tuple Elements

Since TypeScript 3.0, a tuple can have optional elements specified using the question mark (?) postfix.

For example, you can define an RGBA tuple with the optional alpha channel value:

```
let bgColor, headerColor: [number, number, number, number?];
bgColor = [0, 255, 255, 0.5];
headerColor = [0, 255, 255];
```

> Note that the RGBA defines colors using the red, green, blue, and alpha models. The alpha specifies the opacity of the color.

Summary

- A tuple is an array with a fixed number of elements whose types are known.