TypeScript Array Type

Summary: in this tutorial, you'll learn about the TypeScript `array` type and its basic operations.

Introduction to TypeScript array type

A TypeScript `array` is an ordered list of data. To declare an array that holds values of a specific type, you use the following syntax:

```
let arrayName: type[];
```

For example, the following declares an array of strings:

```
let skills: string[] = [];
```

And you can add one or more strings to the array:

```
skills[0] = "Problem Solving";
skills[1] = "Programming";
```

or use the `push()` method:

```
skills.push('Software Design');
```

The following declares a variable and assigns an array of strings to it:

```
let skills = ['Problem Sovling','Software Design','Programming'];
```

In this example, TypeScript infers the `skills` array as an array of strings. It is equivalent to the following:

```
let skills: string[];
skills = ['Problem Sovling','Software Design','Programming'];
```

After you define an array of a specific type, TypeScript will prevent you from adding incompatible values. For example, the following will cause an error:

```
skills.push(100);
```

… because we're trying to add a number to the string array.

Error:

```
Argument of type 'number' is not assignable to parameter of type 'string'.
```

When you extract an element from the array, TypeScript infers the type of the array element. For example:

```
let skill = skills[0];
console.log(typeof(skill));
```

Output:

```
string
```

In this example, we extract the first element of the `skills` array and assign it to the `skill` variable.

Since an element in a string array is a string, TypeScript infers the type of the `skill` variable to string as shown in the output.

TypeScript array properties and methods

TypeScript arrays have the same properties and methods as JavaScript arrays. For example, the following uses the `length` property to get the number of elements in an array:

```
let series = [1, 2, 3];
console.log(series.length); // 3
```

You can use all the useful array methods such as forEach(), map(), reduce(), and filter(). For example:

```
let series = [1, 2, 3];
let doubleIt = series.map(e => e* 2);
console.log(doubleIt);
```

Output:

```
[ 2, 4, 6 ]
```

Storing values of mixed types

The following illustrates how to define an array that holds both strings and numbers:

```
let scores = ['Programming', 5, 'Software Design', 4];
```

In this case, TypeScript infers the `scores` array as an array of `string | number`. It's equivalent to the following:

```
let scores : (string | number)[];
scores = ['Programming', 5, 'Software Design', 4];
```

Summary

- In TypeScript, an array is an ordered list of values.
- Use the `let arr: type[]` syntax to declare an array of a specific type. Adding a value of a different type to the array will result in an error.

- An array can store values of mixed types. Use the `arr: (type1 | type2) []` syntax to declare an array of values with mixed types (`type1` , and `type2`)