

# TypeScript - Arrays

An array is a special type of data type which can store multiple values of different data types sequentially using a special syntax.

TypeScript supports arrays, similar to JavaScript. There are two ways to declare an array:

1. Using square brackets. This method is similar to how you would declare arrays in JavaScript.

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];
```

2. Using a generic array type, Array<elementType>.

```
let fruits: Array<string> = ['Apple', 'Orange', 'Banana'];
```

Both methods produce the same output.

Arrays can contain elements of any data type, numbers, strings, or even objects.

```
let arr = [1, 3, 'Apple', 'Orange', 'Banana', true, false];
```

Arrays can be declared and initialized separately.

JavaScript

```
let fruits: Array<string>;  
fruits = ['Apple', 'Orange', 'Banana'];  
  
let ids: Array<number>;  
ids = [23, 34, 100, 124, 44];
```

An array in TypeScript can contain elements of different data types using a generic array type syntax, as shown below.

JavaScript

```
let values: (string | number)[] = ['Apple', 2, 'Orange', 3, 4, 'Banana'];  
// or  
let values: Array<string | number> = ['Apple', 2, 'Orange', 3, 4, 'Banana'];
```

## Accessing Array Elements:

The array elements can be accessed using the index of an element e.g. `ArrayName[index]`.

The array index starts from zero, so the index of the first element is zero, the index of the second element is one and so on.

JavaScript

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];  
fruits[0]; // returns Apple  
fruits[1]; // returns Orange  
fruits[2]; // returns Banana  
fruits[3]; // returns undefined
```

Use the for loop to access array elements as shown below.

JavaScript

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];  
for(var index in fruits)  
{  
    console.log(fruits[index]); // output: Apple Orange Banana  
}  
  
for(var i = 0; i < fruits.length; i++)  
{  
    console.log(fruits[i]); // output: Apple Orange Banana  
}
```

## TypeScript Array Methods

Method	Description
pop()	Removes the last element of the array and return that element
push()	Adds new elements to the array and returns the new array length
sort()	Sorts all the elements of the array
concat()	Joins two arrays and returns the combined result
indexOf()	Returns the index of the first match of a value in the array (-1 if not found)
copyWithin()	Copies a sequence of elements within the array
fill()	Fills the array with a static value from the provided start index to the end index

shift()	Removes and returns the first element of the array
splice()	Adds or removes elements from the array
unshift()	Adds one or more elements to the beginning of the array
includes()	Checks whether the array contains a certain element
join()	Joins all elements of the array into a string
lastIndexOf()	Returns the last index of an element in the array
slice()	Extracts a section of the array and returns the new array
toString()	Returns a string representation of the array
toLocaleString()	Returns a localized string representing the array

JavaScript

```
var fruits: Array<string> = ['Apple', 'Orange', 'Banana'];

fruits.sort();

console.log(fruits); //output: [ 'Apple', 'Banana', 'Orange' ]

console.log(fruits.pop()); //output: Orange

fruits.push('Papaya');

console.log(fruits); //output: ['Apple', 'Banana', 'Papaya']

fruits = fruits.concat(['Fig', 'Mango']);

console.log(fruits); //output: ['Apple', 'Banana', 'Papaya', 'Fig', 'Mango']

console.log(fruits.indexOf('Papaya')); //output: 2
```