TypeScript 声明数组的语法格式如下所示：

var array\_name[:datatype];        //声明

array\_name = [val1,val2,valn..]   //初始化

// 或者直接在声明时初始化：

var array\_name[:datatype] = [val1,val2…valn]

**实例**

var numlist:number[] = [2,4,6,8]

**数组解构**

我们也可以把数组元素赋值给变量，如下所示：

var arr:number[] = [12,13]

var[x,y] = arr // 将数组的两个元素赋值给变量 x 和 y

console.log(x)

console.log(y)

**数组迭代**

我们可以使用 for 语句来循环输出数组的各个元素：

var j:any;

var nums:number[] = [1001,1002,1003,1004]

for(j in nums) {

console.log(nums[j])

}

**多维数组**

var arr\_name:datatype[][]=[ [val1,val2,val3],[v1,v2,v3] ]

**数组方法**

|  |  |  |
| --- | --- | --- |
| **序号** | **方法 & 描述** | **实例** |
| 1. | concat()  连接两个或更多的数组，并返回结果。 | var alpha = ["a", "b", "c"];  var numeric = [1, 2, 3];  var alphaNumeric = alpha.concat(numeric);  console.log("alphaNumeric : " + alphaNumeric ); // a,b,c,1,2,3 |
| 2. | every()  检测数值元素的每个元素是否都符合条件。 | function isBigEnough(element, index, array) {  return (element >= 10);  }    var passed = [12, 5, 8, 130, 44].every(isBigEnough);  console.log("Test Value : " + passed ); // false |
| 3. | filter()  检测数值元素，并返回符合条件所有元素的数组。 | function isBigEnough(element, index, array) {  return (element >= 10);  }    var passed = [12, 5, 8, 130, 44].filter(isBigEnough);  console.log("Test Value : " + passed ); // 12,130,44 |
| 4. | forEach()  数组每个元素都执行一次回调函数。 | let num = [7, 8, 9];  num.forEach(function (value) {  console.log(value);  });  编译成 JavaScript 代码：  var num = [7, 8, 9];  num.forEach(function (value) {  console.log(value); // 7 8 9  }); |
| 5. | indexOf()  搜索数组中的元素，并返回它所在的位置。 | var index = [12, 5, 8, 130, 44].indexOf(8);  console.log("index is : " + index ); // 2 |
| 6. | join()  把数组的所有元素放入一个字符串。 | var arr = new Array("First","Second","Third");    var str = arr.join();  console.log("str : " + str ); // First,Second,Third    var str = arr.join(", ");  console.log("str : " + str ); // First, Second, Third    var str = arr.join(" + ");  console.log("str : " + str ); // First + Second + Third |
| 7. | lastIndexOf()  返回一个指定的字符串值最后出现的位置，在一个字符串中的指定位置从后向前搜索。 | var index = [12, 5, 8, 130, 44].lastIndexOf(8);  console.log("index is : " + index ); // 2 |
| 8. | map()  通过指定函数处理数组的每个元素，并返回处理后的数组。 | var numbers = [1, 4, 9];  var roots = numbers.map(Math.sqrt);  console.log("roots is : " + roots ); // 1,2,3 |
| 9. | pop()  删除数组的最后一个元素并返回删除的元素。 | var numbers = [1, 4, 9];    var element = numbers.pop();  console.log("element is : " + element ); // 9    var element = numbers.pop();  console.log("element is : " + element ); // 4 |
| 10. | push()  向数组的末尾添加一个或更多元素，并返回新的长度。 | var numbers = new Array(1, 4, 9);  var length = numbers.push(10);  console.log("new numbers is : " + numbers ); // 1,4,9,10  length = numbers.push(20);  console.log("new numbers is : " + numbers ); // 1,4,9,10,20 |
| 11. | reduce()  将数组元素计算为一个值（从左到右）。 | var total = [0, 1, 2, 3].reduce(function(a, b){ return a + b; });  console.log("total is : " + total ); // 6 |
| 12. | reduceRight()  将数组元素计算为一个值（从右到左）。 | var total = [0, 1, 2, 3].reduceRight(function(a, b){ return a + b; });  console.log("total is : " + total ); // 6 |
| 13. | reverse()  反转数组的元素顺序。 | var arr = [0, 1, 2, 3].reverse();  console.log("Reversed array is : " + arr ); // 3,2,1,0 |
| 14. | shift()  删除并返回数组的第一个元素。 | var arr = [10, 1, 2, 3].shift();  console.log("Shifted value is : " + arr ); // 10 |
| 15. | slice()  选取数组的的一部分，并返回一个新数组。 | var arr = ["orange", "mango", "banana", "sugar", "tea"];  console.log("arr.slice( 1, 2) : " + arr.slice( 1, 2) ); // mango  console.log("arr.slice( 1, 3) : " + arr.slice( 1, 3) ); // mango,banana |
| 16. | some()  检测数组元素中是否有元素符合指定条件。 | function isBigEnough(element, index, array) {  return (element >= 10);    }    var retval = [2, 5, 8, 1, 4].some(isBigEnough);  console.log("Returned value is : " + retval ); // false    var retval = [12, 5, 8, 1, 4].some(isBigEnough);  console.log("Returned value is : " + retval ); // true |
| 17. | sort()  对数组的元素进行排序。 | var arr = new Array("orange", "mango", "banana", "sugar");  var sorted = arr.sort();  console.log("Returned string is : " + sorted ); // banana,mango,orange,sugar |
| 18. | splice()  从数组中添加或删除元素。 | var arr = ["orange", "mango", "banana", "sugar", "tea"];  var removed = arr.splice(2, 0, "water");  console.log("After adding 1: " + arr ); // orange,mango,water,banana,sugar,tea  console.log("removed is: " + removed);    removed = arr.splice(3, 1);  console.log("After removing 1: " + arr ); // orange,mango,water,sugar,tea  console.log("removed is: " + removed); // banana |
| 19. | toString()  把数组转换为字符串，并返回结果。 | var arr = new Array("orange", "mango", "banana", "sugar");  var str = arr.toString();  console.log("Returned string is : " + str ); // orange,mango,banana,sugar |
| 20. | unshift()  向数组的开头添加一个或更多元素，并返回新的长度。 | var arr = new Array("orange", "mango", "banana", "sugar");  var length = arr.unshift("water");  console.log("Returned array is : " + arr ); // water,orange,mango,banana,sugar  console.log("Length of the array is : " + length ); // 5 |