

آموزش TypeScript – آموزش آرایه ها در تایپ اسکریپت (TypeScript)

استفاده از متغیرها جهت ذخیره سازی مقادیر، محدودیت های زیر را به همراه دارد.

- متغیرها ذاتا اسکالر هستند. به بیان دیگر اعلان یک متغیر در یک زمان مشخص تنها می تواند شامل یک واحد باشد. این یعنی برای ذخیره سازی n مقدار در یک برنامه به n اعلان متغیر نیاز است. در نتیجه در صورتی که کسی بخواهد گروه های بزرگتری از مقادیر را ذخیره کند، استفاده از متغیرها در دسترس می شود.
 - متغیرهای یک برنامه حافظه های اختصاص داده شده ای هستند که ترتیب مشخصی ندارند. به همین دلیل دستیابی و خواندن مقادیر براساس ترتیب اعلان شده ی این متغیرها کار سختی می باشد.
- تایپ اسکریپت برای حل همین مشکلات بود که مفهوم آرایه ها را معرفی کرد. یک آرایه مجموعه ای همگن از مقادیر است. به بیان ساده تر آرایه مجموعه ای از مقادیر است که نوع داده ی آن ها یکسان است. نوع آرایه توسط کاربر تعریف می شود.

آموزش ویژگی های آرایه در TypeScript

در زیر لیستی از ویژگی های یک آرایه را مشاهده می کنید:

- اعلان آرایه بلوک های مرتبی از حافظه را به برنامه اختصاص می دهد.
- آرایه ها استاتیک هستند. این یعنی بعد از اینکه آرایه مقدار دهی اولیه شود، دیگر نمی توان سائز آن را تغییر داد.
- هر بلوک حافظه بیانگر المانی از آرایه می باشد.
- المان های آرایه توسط عدد صحیح منحصر به فردی به نام subscript یا index مربوط به المان شناسایی می شوند.
- آرایه ها درست مانند متغیرها قبل از اینکه بتوان آن ها را استفاده کرد، باید اعلان شوند. برای این کار می توان از عبارت کلیدی var استفاده کرد.
- مقدار دهی اولیه به یک آرایه همان پر کردن المان های آرایه است. مقادیر المان آرایه را می توان به روزرسانی و اصلاح کرد. اما امکان پاک کردن آن ها وجود ندارد.

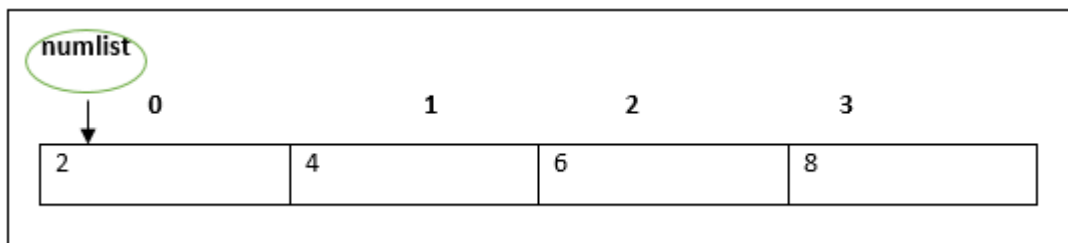
آموزش مقداردهی اولیه و اعلان آرایه ها در TypeScript

برای اعلان و دادن مقدار اولیه به آرایه در تایپ اسکریپت می توان از سینتکس زیر استفاده کرد.

سینتکس

```
var array_name[:datatype]; //declaration
array_name = [val1,val2,valn...] //initialization
```

در صورتی که طی اعلان آرایه نوع داده مشخص نشود، فرض بر این گذاشته می شود که نوع داده any است. نوع چنین آرایه ای از نوع داده ی اولین المان آرایه طی مقدار دهی اولیه استنتاج می شود. برای مثال اعلان آرایه ای مانند `var numlist:number[] = [2,4,6,8]` زیر را ایجاد می کند.



نشانهگر آرایه به صورت پیش فرض به اولین المان اشاره دارد.

کار مقداردهی اولیه و اعلان آرایه ها را می توان در یک دستور انجام داد. سینتکس این کار در زیر آمده است:

```
var array_name[:data type] = [val1,val2...valn]
```

نکته : به جفت [] اندازه ی آرایه گفته می شود.

آموزش دسترسی به المان های آرایه در TypeScript

اسم آرایه ای که به دنبال subscript آورده شود، در اشاره کردن به المانی از آرایه کاربرد دارد. سینتکس آن در زیر آمده است.

```
array_name[subscript] = value
```

مثال : آرایه ی ساده

```
var alphas:string[];  
alphas = ["1","2","3","4"]  
console.log(alphas[0]);  
console.log(alphas[1]);
```

بعد از کامپایل کردن کد بالا، کد جاوا اسکریپت زیر ایجاد می شود.

```
//Generated by typescript 1.8.10
```

```
var alphas;  
alphas = ["1", "2", "3", "4"];
```

```
console.log(alphas[0]);  
console.log(alphas[1]);
```

و خروجی به صورت زیر نمایش داده می شود.

```
1  
2
```

مثال : اعلان و مقداردهی اولیه در یک دستور

```
var nums:number[] = [1,2,3,3]  
console.log(nums[0]);  
console.log(nums[1]);  
console.log(nums[2]);  
console.log(nums[3]);
```

بعد از کامپایل کردن کد بالا، کد جاوااسکریپت زیر ایجاد می شود.

```
//Generated by typescript 1.8.10  
var nums = [1, 2, 3, 3];  
console.log(nums[0]);  
console.log(nums[1]);  
console.log(nums[2]);  
console.log(nums[3]);
```

و خروجی به صورت زیر نمایش داده می شود.

```
1  
2  
3  
3
```

آموزش شیء آرایه در TypeScript

آرایه را می توان با استفاده از شیء آرایه نیز ایجاد کرد. سازنده ی آرایه را می توان به تابع داد. مقداری عددی که بیانگر اندازه ی آرایه باشد، یا لیستی از مقادیر که با استفاده از کاما جدا شده اند. در مثال زیر نحوه ی ایجاد آرایه با استفاده از این روش نشان داده شده است.

مثال

```
var arr_names:number[] = new Array(4)

for(var i = 0;i<arr_names.length;i++) {

    arr_names[i] = i * 2

    console.log(arr_names[i])

}
```

بعد از کامپایل کردن کد بالا، کد جاوا اسکریپت زیر ایجاد می شود.

```
//Generated by typescript 1.8.10

var arr_names = new Array(4);

for (var i = 0; i < arr_names.length; i++) {

    arr_names[i] = i * 2;

    console.log(arr_names[i]);

}
```

و خروجی به صورت زیر نمایش داده می شود.

```
0
2
4
6
```

مثال : سازنده ی آرایه مقادیر جدا شده توسط کاما را می پذیرد

```
var names:string[] = new Array("Mary","Tom","Jack","Jill")
```

```
for(var i = 0;i<names.length;i++) {
  console.log(names[i])
}
```

بعد از کامپایل کردن کد بالا، کد جاوا اسکریپت زیر ایجاد می شود.

```
//Generated by typescript 1.8.10
var names = new Array("Mary", "Tom", "Jack", "Jill");
for (var i = 0; i < names.length; i++) {
  console.log(names[i]);
}
```

و خروجی به صورت زیر نمایش داده می شود.

```
Mary
Tom
Jack
Jill
```

آموزش متدهای آرایه در TypeScript

لیستی از متدهای شیء آرایه به همراه مشخصات آن ها را می توانید در جدول زیر مشاهده کنید.

متد و توضیحات	ردیف
<p>concat()</p> <p>آرایه ی جدیدی متشکل از این آرایه به همراه آرایه های دیگر و (یا) مقدار یا مقادیر دیگر را برگشت می دهد.</p>	1.
<p>every()</p> <p>در صورتی که تمام المان های این آرایه تابع آزمایشی ارائه شده را با موفقیت پشت سر بگذارند، مقدار true را برگشت می دهد.</p>	2.
<p>filter()</p> <p>آرایه ی جدیدی به همراه تمام المان های این آرایه را ایجاد می کند. برای چنین حالتی تابع filtering ارائه شده مقدار true را برگشت می دهد.</p>	3.

4.	forEach() به ازای هر یک از المان های آرایه داده ای را فراخوانی می کند.
5.	indexOf() اولین (کمترین) ایندکس یک المان را برابر با مقداری مشخص و در صورتی که مقداری پیدا نشود برابر با -1 ، برگشت می دهد.
6.	join() تمامی المان های آرایه را در یک رشته به یکدیگر پیوند می زند.
7.	lastIndexOf() آخرین (بزرگترین) ایندکس یک المان را برابر با مقداری مشخص و در صورتی که مقداری پیدا نشود برابر با -1 ، برگشت می دهد.
8.	map() در تمامی المان های این آرایه، آرایه ای جدید به همراه نتایج فراخوانی تابع ارائه شده را ایجاد می کند
9.	pop() آخرین المان را از یک آرایه حذف کرده و همان المان را برگشت می دهد.
10.	push() یک یا چند المان را به انتهای یک آرایه اضافه کرده و طول جدید آرایه را برگشت می دهد.
11.	reduce() به صورت همزمان تابعی را بر روی دو مقدار این آرایه (از چپ به راست) اعمال می کند. برای اینکه این آرایه را به تنها یک مقدار کاهش دهد.
12.	reduceRight() به صورت همزمان تابعی را بر روی دو مقدار این آرایه (از راست به چپ) اعمال می کند. برای اینکه این آرایه را به تنها یک مقدار کاهش دهد.
13.	reverse() ترتیب المان های یک آرایه را برعکس می کند. به این صورت که اولین المان جای خود را به آخرین المان و آخرین المان جای خود را به اولین المان می دهد.
14.	shift() اولین المان را از یک آرایه حذف کرده و همان المان را برگشت می دهد.
15.	slice() بخشی از یک آرایه را استخراج کرده و آرایه ی جدیدی را برگشت می دهد.
16.	some() اگر حداقل یکی از المان های این آرایه تابع آزمایشی ارائه شده را با موفقیت پشت سر بگذارد، مقدار true را برگشت می دهد.
17.	sort() المان های آرایه را مرتب می کند.

splice() المان هایی را از یک آرایه حذف کرده یا به آن اضافه می کند.	.18
toString() رشته ای را برگشت می دهد، به گونه ای که این رشته بیانگر آرایه و المان های آن است.	.19
unshift() یک یا چند المان را به جلوی یک آرایه اضافه کرده و طول جدید آرایه را برگشت می دهد.	.20

آموزش نابود کردن آرایه در TypeScript

این فرآیند به شکستن ساختار یک هویت گفته می شود. در صورتی که این کار در متن یک آرایه انجام شود، تایپ اسکریپت از فرآیند نابود کردن آرایه پشتیبانی می کند.

مثال

```
var arr:number[] = [12,13]

var [x,y] = arr

console.log(x)

console.log(y)
```

بعد از کامپایل کردن کد بالا، کد جاوا اسکریپت زیر ایجاد می شود.

```
//Generated by typescript 1.8.10

var arr = [12, 13];

var x = arr[0], y = arr[1];

console.log(x);

console.log(y);
```

و خروجی به صورت زیر نمایش داده می شود.

```
12
13
```

آموزش پیمایش آرایه با استفاده از حلقه ی for...in در TypeScript

برای پیمایش آرایه می توان از حلقه ی for...in استفاده کرد.

```

var j:any;

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

for(j in nums) {

  console.log(nums[j])

}

```

این حلقه کار پیمایش را براساس ایندکس آرایه انجام می دهد.
بعد از کامپایل کردن کد بالا، کد جاوا اسکریپت زیر ایجاد می شود.

```

//Generated by typescript 1.8.10

var j;

var nums = [1001, 1002, 1003, 1004];

for (j in nums) {

  console.log(nums[j]);

}

```

و خروجی به صورت زیر نمایش داده می شود.

```

1001
1002
1003
1004

```

آموزش آرایه ها در تایپ اسکریپت (TypeScript)

تایپ اسکریپت از مفاهیم زیر در آرایه ها پشتیبانی می کند.

مفهوم و توضیحات	ردیف
<p>Multi-dimensional arrays</p> <p>تایپ اسکریپت از آرایه های چندبعدی پشتیبانی می کند. ساده ترین شکل این آرایه ها، آرایه های دوبعدی هستند.</p>	1.

Passing arrays to functions	.2
می توانید با مشخص کردن اسم آرایه بدون هیچ ایندکسی، این آرایه را به تابع بدهید.	
Return array from functions	.3
این امکان را به تابع می دهد تا آرایه ای را برگشت دهد.	