

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

بعضی مواقع ممکن است که نیاز باشد مجموعه ای از مقادیر که نوع آن ها متفاوت است را ذخیره کنیم. آرایه ها در چنین موقعیت هایی کاربرد ندارند. تایپ اسکریپت نوع داده ای چندتایی (tuple) را در اختیار ما می گذارد، تا بتوانیم در این راستا از آن استفاده کنیم.

تاپل بیانگر مجموعه ای ناهمگن از مقادیر است. به بیان دیگر، با کمک تاپل می توان فیلهای متعددی را ذخیره سازی کرد که نوع داده های آن ها مختلف باشد. تاپل ها را می توان به عنوان پارامتر به توابع داد.

سینتکس

```
var tuple_name = [value1,value2,value3,...value n]
```

مثال

```
var mytuple = [10,"Hello"];
```

همچنین می توانید در تایپ اسکریپت یک تاپل خالی را اعلان کرده و بعدا به آن مقدار اولیه بدهید.

```
var mytuple = [];  
mytuple[0] = 120  
mytuple[1] = 234
```

آموزش دسترسی به مقادیر در تاپل ها در TypeScript

به تک تک مقادیر تاپل، آیتم گفته می شود. تاپل ها مبتنی بر ایندکس هستند. این یعنی با استفاده از ایندکس عددی متناظر هر آیتم می توان به آیتم های تاپل دسترسی پیدا کرد. این ایندکس ها از صفر شروع شده و تا $n-1$ ادامه دارند (n اندازه ی تاپل است).

سینتکس

```
tuple_name[index]
```

مثال : تاپل ساده

```
var mytuple = [10,"Hello"]; //create a tuple  
console.log(mytuple[0])
```

```
console.log(mytuple[1])
```

در مثال بالا، تاپلی به نام mytuple اعلان شده است. این تاپل به ترتیب شامل نوع های داده ای عددی و رشته ای است.

بعد از کامپایل کردن کد بالا، همین کد در جاوا اسکریپت ایجاد می شود.
و خروجی به صورت زیر نمایش داده می شود.

```
10  
Hello
```

مثال : تاپل خالی

```
var tup = []  
  
tup[0] = 12  
tup[1] = 23  
  
console.log(tup[0])  
console.log(tup[1])
```

بعد از کامپایل کردن کد بالا، همین کد در جاوا اسکریپت ایجاد می شود.
و خروجی به صورت زیر نمایش داده می شود.

```
12  
23
```

عملیات های تاپل

تاپل ها در تایپ اسکریپت از عملیات های مختلفی مانند push کردن آیتمی جدید، حذف کردن یک آیتم از تاپل و .. پشتیبانی می کنند.

مثال

```
var mytuple = [10,"Hello","World","typeScript"];  
  
console.log("Items before push "+mytuple.length) // returns the tuple size
```

```
mytuple.push(12) // append value to the tuple

console.log("Items after push "+mytuple.length)

console.log("Items before pop "+mytuple.length)

console.log(mytuple.pop()+" popped from the tuple") // removes and returns the last item

console.log("Items after pop "+mytuple.length)
```

- push() آیتمی را به تاپل پیوست می کند.
- pop() آخرین مقدار تاپل را حذف کرده و آن را برگشت می دهند.

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

و خروجی به صورت زیر نمایش داده می شود.

```
Items before push 4
Items after push 5
Items before pop 5
12 popped from the tuple
Items after pop 4
```

آموزش آپدیت کردن تاپل ها در TypeScript

تاپل ها mutable هستند. به این معنی که می توان مقادیر المان های تاپل را آپدیت کرده و یا تغییر داد.

مثال

```
var mytuple = [10,"Hello","World","typeScript"]; //create a tuple

console.log("Tuple value at index 0 "+mytuple[0])

//update a tuple element

mytuple[0] = 121

console.log("Tuple value at index 0 changed to "+mytuple[0])
```

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

و خروجی به صورت زیر نمایش داده می شود.

```
Tuple value at index 0 10  
Tuple value at index 0 changed to 121
```

آموزش نابود کردن تاپل در TypeScript

به شکستن ساختار یک هویت نابود کردن گفته می شود. در صورتی که این فرآیند در متن تاپل استفاده شود، تایپ اسکریپت از آن پشتیبانی می کند.

مثال

```
var a =[10,"hello"]  
  
var [b,c] = a  
  
console.log( b )  
  
console.log( c )
```

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

```
//Generated by typescript 1.8.10  
  
var a = [10, "hello"];  
  
var b = a[0], c = a[1];  
  
console.log(b);  
  
console.log(c);
```

و خروجی به صورت زیر نمایش داده می شود.

```
10  
hello
```