

## آموزش TypeScript – آموزش اشیاء در تایپ اسکریپت (TypeScript)

Object ها نمونه هایی هستند که شامل مجموعه ای از جفت مقادیر کلیدی (key value pairs) هستند. این مقادیر می توانند اسکالر، تابع و یا حتی آرایه ای از دیگر اشیاء باشند. سینتکس مربوط به این مطلب در زیر آمده است.

سینتکس

```
var object_name = {
  key1: "value1", //scalar value
  key2: "value",
  key3: function() {
    //functions
  },
  key4:["content1", "content2"] //collection
};
```

همانطور که در بالا مشاهده می کنید، یک شیء می تواند شامل مقادیر اسکالر، توابع و ساختارهایی مانند آرایه ها و تاپل ها باشد.

مثال : نحوه ی نوشتن شیء

```
var person = {
  firstname: "Tom",
  lastname: "Hanks"
};

//access the object values

console.log(person.firstname)

console.log(person.lastname)
```

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

و خروجی به صورت زیر نمایش داده می شود.

```
Tom
Hanks
```

## آموزش Type Template در تایپ اسکریپت (TypeScript)

فرض کنید شیء لفظی مانند زیر در جاوا اسکریپت ایجاد کرده اید.

```
var person = {  
  firstname: "Tom",  
  lastname: "Hanks"  
};
```

اگر بخواهید مقداری را به شیئی اضافه کنید، جاوا اسکریپت این امکان را به شما می دهد تا تغییرات لازم را ایجاد کنید. حالا فرض کنید که بخواهیم تابعی را به شیء person اضافه کنیم، برای انجام این کار بایستی از روش زیر استفاده کرد.

```
person.sayHello = function(){ return "hello";}
```

در صورتی که همین کد را در تایپ اسکریپت استفاده کنید، کامپایلر خطا می دهد. زیرا در تایپ اسکریپت اشیاء واقعی باید دارای یک type template باشند. اشیاء موجود در تایپ اسکریپت بایستی نمونه ای از نوع مشخصی باشند. برای حل این مشکل می توان در اعلان کردن از method template استفاده کرد.

مثال : type template در تایپ اسکریپت

```
var person = {  
  firstName: "Tom",  
  lastName: "Hanks",  
  sayHello: function() { } //Type template  
}  
  
person.sayHello = function() {  
  console.log("hello "+person.firstName)  
}  
  
person.sayHello()
```

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

و خروجی به صورت زیر نمایش داده می شود.

```
hello Tom
```

اشیاء را می توان به عنوان پارامتر نیز به توابع داد.

مثال : اشیاء به عنوان پارامترهای توابع

```
var person = {  
  firstname:"Tom",  
  lastname:"Hanks"  
};  
  
var invokeperson = function(obj: { firstname:string, lastname :string }) {  
  console.log("first name :"+obj.firstname)  
  console.log("last name :"+obj.lastname)  
}  
  
invokeperson(person)
```

در این مثال یک شیء لفظی اعلان شده است. عبارت تابع با دادن شیء `person` احضار می شود.

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

```
//Generated by typescript 1.8.10  
  
var person = {  
  firstname: "Tom",  
  lastname: "Hanks"  
};  
  
var invokeperson = function (obj) {  
  console.log("first name : " + obj.firstname);  
  console.log("last name : " + obj.lastname);  
};
```

```
invokeperson(person);
```

و خروجی به صورت زیر نمایش داده می شود.

```
first name :Tom  
last name :Hanks
```

می توانید شیء بی نامی را ایجاد کرده و آن را به fly بدهید.

مثال : شیء بی نام

```
var invokeperson = function(obj:{ firstname:string, lastname :string}) {  
  
  console.log("first name :"+obj.firstname)  
  
  console.log("last name :"+obj.lastname)  
  
}  
  
invokeperson({firstname:"Sachin",lastname:"Tendulkar"});
```

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

```
//Generated by typescript 1.8.10  
  
var invokeperson = function (obj) {  
  
  console.log("first name : " + obj.firstname);  
  
  console.log("last name : " + obj.lastname);  
  
};  
  
invokeperson({ firstname: "Sachin", lastname: "Tendulkar" });  
  
invokeperson({ firstname: "Sachin", lastname: "Tendulkar" });
```

و خروجی به صورت زیر نمایش داده می شود.

```
first name :Sachin  
last name :Tendulkar
```

## آموزش Duck-typing در TypeScript

در duck-typing دو شیء در صورتی که هر دو مشترکا دارای مشخصات یکسانی باشند، هم نوع در نظر گرفته می شوند. -typing duck برای چک کردن مناسب بودن دو شیء، به جای تایید نوع واقعی این اشیا، حضور مشخصات خاصی از آن ها را تایید می کند. این مفهوم را می توان با استفاده از پاراگراف زیر بهتر بیان کرد:

« زمانی که من پرنده ای ببینم که راه رفتن، شنا کردن و صدایش شبیه به یک اردک باشد، آنگاه اسم آن پرنده را اردک می گذارم. »

کامپایلر تایپ اسکریپت با بهره گیری از سیستم داک تایپینگ امکان ایجاد اشیا در fly در عین حفظ نوع را فراهم کرده است. در مثال زیر چگونگی دادن اشیا به مستقیما از رابطی بهره نمی برند بلکه شامل تمامی اعضای مورد نیاز یک تابع هستند را نشان داده شده است.

### Example

```
interface IPoint {  
  x:number  
  y:number  
}  
  
function addPoints(p1:IPoint,p2:IPoint):IPoint {  
  var x = p1.x + p2.x  
  var y = p1.y + p2.y  
  return {x:x,y:y}  
}  
  
//Valid  
var newPoint = addPoints({x:3,y:4},{x:5,y:1})  
  
//Error  
var newPoint2 = addPoints({x:1},{x:4,y:3})
```