

در این پست به تشریح انواع داده در زبان TypeScript و ذکر مثال در این زمینه می‌پردازیم.

تعریف متغیرها و انواع داده

در TypeScript هنگام تعریف متغیرها باید نوع داده ای آن‌ها را مشخص کنیم. در TypeScript پنج نوع داده ای وجود دارد که در زیر با ذکر مثال تعریف شده اند. مفاهیم ماژول، کلاس و تابع در پست بعدی به تشریح توضیح داده خواهند شد.

number : معادل نوع داده ای number در JavaScript است. برای ذخیره سازی اعداد صحیح و اعشاری استفاده می‌شود.
یک مثال:

```
class NumberTypeOfTypeScript {
    MyFunction()
    {
        var p: number;
        p = 1;
        var q = 2;
        var r = 3.33;
        alert("Value of P=" + p + " Value of q=" + q + " Value of r=" + r);
    }
}

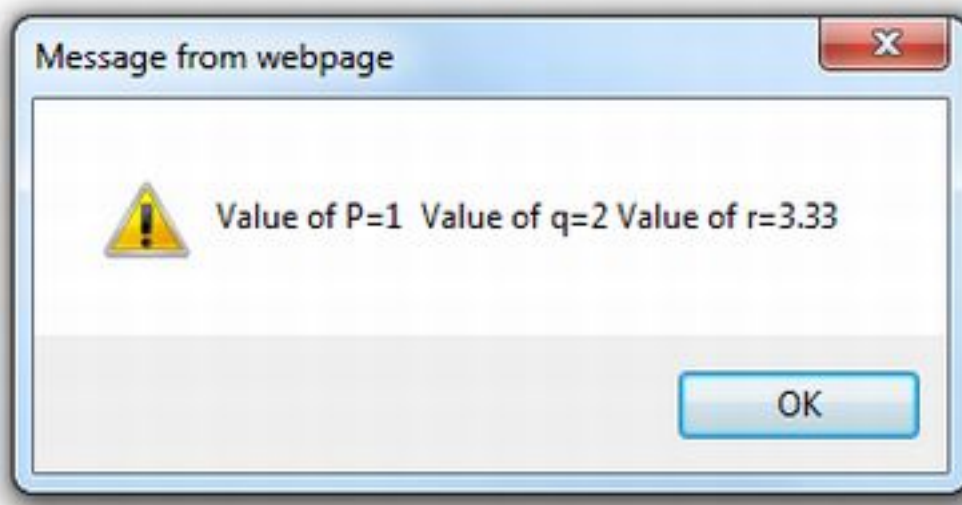
window.onload = () =>{
    var value = new NumberTypeOfTypeScript();
    value.MyFunction();
}
```

حال باید یک فایل Html برای استفاده از این کلاس داشته باشیم. به صورت زیر:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>TypeScript HTML App</title>
    <link rel="stylesheet" href="app.css" type="text/css" />
    <script src="app.js"></script>
</head>
<body>
    <h1>Number Type in TypeScript</h1>
    <div id="content"/>
</body>
</html>
```

بعد از اجرای پروژه خروجی به صورت زیر خواهد بود:

Number Type in TypeScript



string : معادل نوع داده ای رشته ای است و برای ذخیره سازی مجموعه ای از کاراکترها از نوع UTF-16 استفاده می شود.

یک مثال:

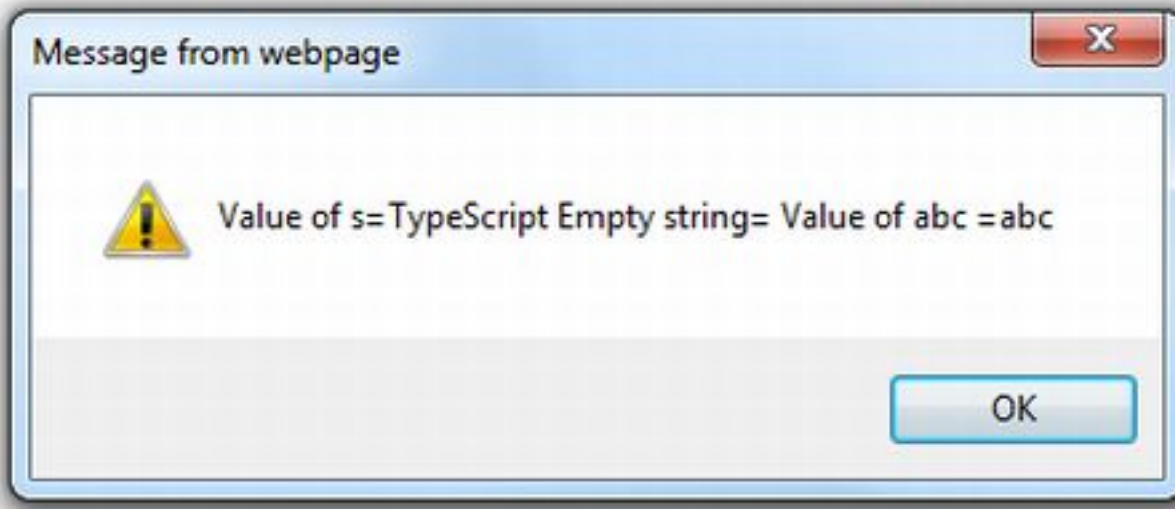
```
class StringTypeOfTypeScript {
  Myfunction() {
    var s: string;
    s="TypeScript"
    var empty = "";
    var abc = "abc";
    alert("Value of s="+ s+" Empty string="+ empty+" Value of abc =" +abc) ;
  }
}
window.onload = () =>{
  var value = new StringTypeOfTypeScript();
  value.Myfunction();
}
```

کد کامپایل شده و تبدیل آن به JavaScript:

```
var StringTypeOfTypeScript = (function () {
  function StringTypeOfTypeScript() {}
  StringTypeOfTypeScript.prototype.Myfunction = function () {
    var s;
    s = "TypeScript";
    var empty = "";
    var abc = "abc";
    alert("Value of s=" + s + " Empty string=" + empty + " Value of abc =" + abc);
  };
  return StringTypeOfTypeScript;
})();
window.onload = function () {
  var value = new StringTypeOfTypeScript();
  value.Myfunction();
};
```

خروجی به صورت زیر است:

String Type in TypeScript



boolean : برای ذخیره سازی مقادیر true یا false می باشد.
مثال:

```
class booleanTypeofTypeScript {
  MyFunction() {
    var lie: bool;
    lie = false;
    var a = 12;
    if (typeof (lie) == "boolean" && typeof (a) == "boolean") {
      alert("Both is boolean type");
    }

    if (typeof (lie) == "boolean" && typeof (a) != "boolean") {
      alert("lie is boolean type and a is not!");
    }
    else {
      alert("a is boolean type and lie is not!");
    }
  }
}

window.onload =()=> {
  var access = new booleanTypeofTypeScript();
  access.MyFunction();
}
```

کد کامپایل شده و تبدیل آن به JavaScript:

```
var booleanTypeofTypeScript = (function () {
  function booleanTypeofTypeScript() {}
  booleanTypeofTypeScript.prototype.MyFunction = function () {
    var lie;
    lie = false;
    var a = 12;
    if(typeof (lie) == "boolean" && typeof (a) == "boolean") {
      alert("Both is boolean type");
    }
    if(typeof (lie) == "boolean" && typeof (a) != "boolean") {
      alert("lie is boolean type and a is not!");
    }
    else {
      alert("a is boolean type and lie is not!");
    }
  }
})();
```

```

    return booleanTypeofTypeScript;
})();
window.onload = function () {
    var access = new booleanTypeofTypeScript();
    access.MyFunction();
};

```

null : همانند دات نت هنگامی که قصد داشته باشیم مقدار یک متغیر را null اختصاص دهیم از این کلمه کلیدی استفاده می‌کنیم.
مثال:

```

class NullTypeinTypeScript {
    MyFunction() {
        var p: number = null;
        var x = null;
        if (p== null) {
            alert("p has null value!");
        }
        else { alert("p has a value"); }
    }
}
window.onload = () =>{
    var value = new NullTypeinTypeScript();
    value.MyFunction();
}

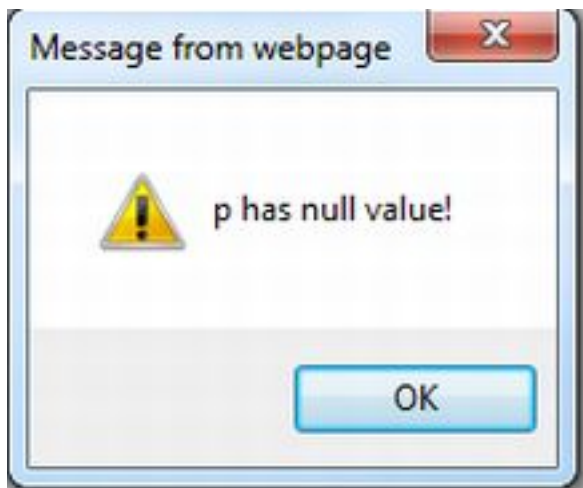
```

کد کامپایل شده و تبدیل آن به JavaScript:

```

var NullTypeinTypeScript = (function () {
    function NullTypeinTypeScript() {}
    NullTypeinTypeScript.prototype.MyFunction = function () {
        var p = null;
        var x = null;
        if(p == null) {
            alert("p has null value!");
        } else {
            alert("p has a value");
        }
    };
    return NullTypeinTypeScript;
})();
window.onload = function () {
    var value = new NullTypeinTypeScript();
    value.MyFunction();
};

```



undefined: معادل نوع undefined در Javascript است. اگر به یک متغیر مقدار اختصاص ندهید مقدار آن undefined خواهد بود.

مثال:

```
class UndefinedTypeOfTypeScript {
  Myfunction() {
    var p: number;
    var x = undefined;
    if (p == undefined && x == undefined) {
      alert("p and x is undefined");
    }
    else { alert("p and c cannot undefined"); }
  }
}
window.onload = () =>{
  var value = new UndefinedTypeOfTypeScript();
  value.Myfunction();
}
```

کد کامپایل شده و تبدیل آن به JavaScript:

```
var UndefinedTypeOfTypeScript = (function () {
  function UndefinedTypeOfTypeScript() {}
  UndefinedTypeOfTypeScript.prototype.Myfunction = function () {
    var p;
    var x = undefined;
    if(p == undefined && x == undefined) {
      alert("p and x is undefined");
    } else {
      alert("p and c cannot undefined");
    }
  };
  return UndefinedTypeOfTypeScript;
})();
window.onload = function () {
  var value = new UndefinedTypeOfTypeScript();
  value.Myfunction();
};
```

خروجی این مثال نیز به صورت زیر است:



ادامه دارد...