

آموزش TypeScript – آموزش اعداد در تایپ اسکریپت (TypeScript)

تایپ اسکریپت مانند جاوا اسکریپت از مقادیر عددی به صورت اشیاء `Number` پشتیبانی می کند. شیء `Number` لفظ های عددی را به نمونه ای از کلاس `Number` تبدیل می کند. این کلاس به عنوان یک `wrapper` عمل کرده و این امکان را فراهم می کند تا بتوان لفظ های عددی را مانند زمانی که شیء باشند، دستکاری کرد.

```
var var_name = new Number(value)
```

اگر آرگومانی غیر عددی به عنوان یک آرگومان به سازنده `Number` داده شود، در این صورت این سازنده `NaN` (`Not-a-Number`) را برگشت می دهد.

در جدول زیر مجموعه ای از مشخصه های شیء `Number` را می توانید مشاهده کنید.

ردیف	مشخصه ها و توضیحات
.1	<code>MAX_VALUE</code> بزرگترین مقداری که یک عدد می تواند در جاوا اسکریپت داشته باشد. <code>1.7976931348623157E+308</code>
.2	<code>MIN_VALUE</code> کمترین مقداری که یک عدد در جاوا اسکریپت داشته باشد. <code>5E-324</code>
.3	<code>NaN</code> برابر با مقداری غیر عددی است.
.4	<code>NEGATIVE_INFINITY</code> مقداری کمتر از <code>MIN_VALUE</code> است.
.5	<code>POSITIVE_INFINITY</code> مقداری بیشتر از <code>MAX_VALUE</code> است.
.6	<code>Prototype</code> مشخصه ی استاتیکی از شیء <code>Number</code> است. از این مشخصه برای تخصیص مشخصه ها و متدهای جدید به شیء <code>Number</code> در سند فعلی استفاده می شود.

Constructor تابعی را برگشت می دهد که نمونه ی این شیء را ساخته است. به صورت پیش فرض این شیء Number است.	7.
---	----

مثال

```

console.log("TypeScript Number Properties: ");
console.log("Maximum value that a number variable can hold: " + Number.MAX_VALUE);
console.log("The least value that a number variable can hold: " + Number.MIN_VALUE);
console.log("Value of Negative Infinity: " + Number.NEGATIVE_INFINITY);
console.log("Value of Negative Infinity: " + Number.POSITIVE_INFINITY);

```

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

و خروجی به صورت زیر نمایش داده می شود.

```

TypeScript Number Properties:
Maximum value that a number variable can hold: 1.7976931348623157e+308
The least value that a number variable can hold: 5e-324
Value of Negative Infinity: -Infinity
Value of Negative Infinity: Infinity

```

مثال : NaN

```

var month = 0
if( month <= 0 || month > 12) {
  month = Number.NaN
  console.log("Month is "+ month)
} else {
  console.log("Value Accepted..")
}

```

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

و خروجی به صورت زیر نمایش داده می شود.

```
Month is NaN
```

مثال : prototype

```
function employee(id:number,name:string) {  
  this.id = id  
  this.name = name  
}  
  
var emp = new employee(123,"Smith")  
employee.prototype.email = "smith@abc.com"  
  
console.log("Employee 's Id: "+emp.id)  
console.log("Employee's name: "+emp.name)  
console.log("Employee's Email ID: "+emp.email)
```

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

```
//Generated by typescript 1.8.10  
function employee(id, name) {  
  this.id = id;  
  this.name = name;  
}  
  
var emp = new employee(123, "Smith");  
employee.prototype.email = "smith@abc.com";
```

```
console.log("Employee's Id: " + emp.id);  
  
console.log("Employee's name: " + emp.name);  
  
console.log("Employee's Email ID: " + emp.email);
```

و خروجی به صورت زیر نمایش داده می شود.

```
Employee's Id: 123  
Employee's name: Smith  
Employee's Email ID: smith@abc.com
```

آموزش متدهای Number در TypeScript

شیء Number تنها شامل متدهای پیش فرضی است که جزئی از تعریف هر شیئی هستند. برخی از متدهای پرکاربرد را میتوانید در جدول زیر مشاهده کنید.

متدها و توضیحات	ردیف
toExponential() عددی را به اجبار به صورت نمایی نمایش می دهد. حتی اگر این عدد در محدوده ای باشد که جاوا اسکریپت در حالت طبیعی آن را به صورت استاندارد نمایش می دهد.	.1
toFixed() عددی را با تعداد مشخصی از ارقام در سمت راست اعشار شکل می دهد.	.2
toLocaleString() نسخه ای از عدد فعلی را به صورت رشته ای بازگشت می دهد. فرمت این عدد ممکن است بر اساس تنظیمات داخلی مرورگر متغیر باشد.	.3
toPrecision() مشخص می کند که به صورت کلی چه تعداد رقم (هم ارقام سمت چپ و هم سمت راست اعشار) از یک عدد باید نمایش داده شود. دقت منفی باعث بروز خطا می شود.	.4
toString() مقدار عدد را به صورت رشته ای برگشت می دهد. به این تابع به عنوان مبنا عدد صحیحی بین 2 تا 36 داده می شود، که با این کار پایه ی مورد استفاده برای ارائه ی مقادیر عددی مشخص می شود.	.5

<p>valueOf() مقدار اولیه ی عدد را برگشت می دهد.</p>	<p>.6</p>
---	-----------