

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

## عملگر چیست؟

عملگر توابعی که بر روی داده اجرا می شوند را تعریف می کند. داده هایی که عملگرها بر روی آن کار انجام می دهند، عملوند هستند. عبارت زیر را در نظر بگیرید:

$$7 + 5 = 12$$

در اینجا مقادیر 7، 5 و 12 عملوند و + و = عملگر هستند.

عملگرهای اصلی تایپ اسکریپت را می توان به گروه های زیر تقسیم کرد:

- عملگرهای ریاضی
- عملگرهای منطقی
- عملگرهای نسبی
- عملگرهای Bitwise
- عملگرهای تخصیصی
- عملگر شرطی یا بر مبنای 3
- عملگر رشته ای
- عملگر نوع

## عملگرهای ریاضی

فرض کنید مقادیر متغیرهای a و b به ترتیب برابر با 10 و 5 باشند.

عملگر	توضیحات	مثال
(جمع) +	مجموع عملوند ها را برگشت می دهد	a + b is 15
(تفریق) -	اختلاف مقادیر را برگشت می دهد	a - b is 5
(ضرب) *	حاصلضرب مقادیر را برگشت می دهد	a * b is 50

(تقسیم) /	عمل تقسیم را انجام داده و خارج قسمت را برگشت می دهد	a / b is 2
(پیمانه) %	عمل تقسیم را انجام داده و باقیمانده را برگشت می دهد	a % b is 0
(افزایش پله ای) ++	مقدار متغیر را یک واحد افزایش میدهد	a++ is 11
(کاهش پله ای) --	مقدار متغیر را یک واحد کاهش می دهد	a-- is 9

## عملگرهای نسبی

این عملگرها نوع رابطه ی بین دو هویت را تست یا تعریف می کنند. عملگرهای نسبی مقدار بولی true یا false را برگشت می دهند. فرض کنید مقدار A برابر با 10 و مقدار B برابر با 20 باشد.

عملگر	توضیحات	مثال
>	بزرگتری	(A > B) is False
<	کوچکتری	(A < B) is True
>=	بزرگتر مساوی	(A >= B) is False
<=	کوچکتر مساوی	(A <= B) is True
==	مساوی	(A == B) is false
!=	نامساوی	(A != B) is True

## آموزش عملگرهای منطقی در TypeScript

این عملگرها برای ترکیب دو شرط یا بیشتر استفاده می شوند. عملگرهای منطقی نیز مقدار بولی را برگشت می دهند. فرض کنید مقدار متغیر A برابر با 10 و B برابر با 20 باشد.

عملگر	توضیحات	مثال
&& (And)	این عملگر تنها در صورتی مقدار true را برگشت می دهد که تمامی عبارات های مشخص شده نیز مقدار true را برگشت دهند.	$(A > 10 \ \&\& \ B > 10)$ is False
(OR)	این عملگر در صورتی مقدار true را برگشت می دهد که حداقل یکی از عبارات های مشخص شده مقدار true را برگشت دهند.	$(A > 10 \    \ B > 10)$ is True
! (NOT)	این عملگر عکس نتیجه ی عبارت را برگشت می دهد. برای مثال $(5 >)$ مقدار false را برگشت می دهد.	$!(A > 10)$ is True

## آموزش عملگرهای Bitwise در TypeScript

فرض کنید متغیر  $A=2$  و  $B=3$  باشد.

عملگر	توضیحات	مثال
& (Bitwise AND)	این عملگر عملیات بولی AND را بر روی هر بیت از آرگومان های صحیح آن اجرا می کند.	$(A \ \& \ B)$ is 2

(BitWise OR)	این عملگر عملیات بولی OR را بر روی هر بیت از آرگومان های صحیح آن اجرا می کند.	$(A   B)$ is 3
^ (Bitwise XOR)	این عملگر عملیات انحصاری بولی OR را بر روی هر بیت از آرگومان های صحیح آن اجرا می کند. OR انحصاری یعنی یا عملوند 1 صحیح است یا عملوند 2 ، نه هر دو	$(A \wedge B)$ is 1
~ (Bitwise Not)	یک عملگر یکانی است که کار آن معکوس کردن تمام بیت های عملوند است.	$(\sim B)$ is -4
<< (Left Shift)	تمام بیت های اولین عملوند را بر اساس تعداد جایگاه های مشخص شده ی عملوند دوم به سمت چپ انتقال می دهد. بیت های جدید با صفر پر می شوند. در صورتی که مقداری را به اندازه ی یک جایگاه به سمت چپ انتقال دهیم این عمل با ضرب کردن آن مقدار در 2 برابری می کند. در صورتی که دو جایگاه به سمت چپ انتقال دهیم، مثل آن می ماند که این مقدار را ضرب در 4 کرده ایم و الی آخر.	$(A \ll 1)$ is 4
>> (Right Shift)	عملگر دو دویی است که مقدار عملوند را به سمت راست انتقال می دهد. مقدار عملوند سمت چپ بر اساس تعداد بیت های مشخص شده توسط عملوند سمت راست، به سمت راست انتقال داده می شوند.	$(A \gg 1)$ is 1
>>> (Right shift with Zero)	این عملگر درست مانند عملگر >> است. با این تفاوت که بیت های انتقال داده شده به سمت چپ همیشه صفر هستند.	$(A \ggg 1)$ is 1

## آموزش عملگرهای تخصیصی در TypeScript

عملگر	توضیحات	مثال
(تخصیص ساده) =	مقادیر عملوند سمت راست را به عملوند سمت چپ تخصیص می دهند.	$C = A + B$ will assign the value of $A + B$ into $C$
(جمع و تخصیص) +=	عملوند سمت راست را با عملوند سمت چپ جمع زده و حاصل جمع را به عملوند سمت چپ تخصیص می دهد.	$C += A$ is equivalent to $C = C + A$
(تفرق و تخصیص) -=	عملوند سمت راست را از عملوند سمت چپ تفرق کرده و حاصل تفریق را به عملوند سمت چپ تخصیص می دهد.	$C -= A$ is equivalent to $C = C - A$
(ضرب و تخصیص) *=	عملوند سمت راست را در عملوند سمت چپ ضرب کرده و حاصل ضرب را به عملوند سمت چپ تخصیص می دهد.	$C *= A$ is equivalent to $C = C * A$
(تقسیم و تخصیص) /=	عملوند سمت راست را بر عملوند سمت چپ تقسیم کرده و حاصل تقسیم را به عملوند سمت چپ تخصیص می دهد.	

**نکته:** همین قانون در عملگرهای Bitwise نیز صدق می کند. در این حالت این عملگرها به  $\ll=$ ,  $\gg=$ ,  $\>>=$ ,  $\&=$  و  $\^=$  تبدیل می شوند.

## آموزش عملگرهای متفرقه در TypeScript

### عملگر منفی (-)

این عملگر علامت یک مقدار را تغییر می دهد. در این رابطه به مثال زیر توجه کنید:

```
var x:number = 4

var y = -x;

console.log("value of x: ",x); //outputs 4

console.log("value of y: ",y); //outputs -4
```

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

```
//Generated by typescript 1.8.10

var x = 4;

var y = -x;

console.log("value of x: ", x); //outputs 4

console.log("value of y: ", y); //outputs -4
```

و خروجی زیر نمایش داده می شود.

```
value of x: 4
value of y: -4
```

### عملگرهای رشته ای : عملگر الحاقی (+)

زمانی که این عملگر بر روی رشته ها اعمال می شود، باعث می شود که رشته ی دوم به رشته ی اول پیوند زده شود. برای فهم بهتر این مطلب به مثال زیر توجه کنید:

```
var msg:string = "hello"+"world"

console.log(msg)
```

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

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

```
var msg = "hello" + "world";  
console.log(msg);
```

و خروجی زیر نمایش داده می شود.

```
helloworld
```

این عملیات جای خالی بین رشته ها اضافه نمی کند. چندین رشته را می توان در یک دستور به یکدیگر پیوند زد.

## آموزش عملگر شرطی (؟) در TypeScript

این عملگر برای ارائه ی یک عبارت شرطی استفاده می شود. بعضی مواقع به این عملگر، عملگر بر مبنای 3 نیز گفته می شود. سینتکس این عملگر را می توانید در زیر مشاهده کنید:

```
Test ? expr1 : expr2
```

- Test : به عبارت شرطی اشاره دارد.
- expr1 : در صورتی که شرط true باشد، مقدار برگشت داده می شود.
- expr2 : در صورتی که شرط false باشد، مقدار برگشت داده می شود.

بیایید به کد زیر نگاهی بیاندازیم:

```
var num:number = -2  
var result = num > 0 ?"positive":"non-positive"  
console.log(result)
```

خط دوم چک می کند که آیا مقدار متغیر num بزرگتر از صفر است یا خیر. در صورتی که مقدار num بزرگتر از صفر باشد، این کد رشته ی "positive" را برگشت می دهد. در غیر این صورت رشته ی "non-positive" برگشت داده خواهد شد.

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

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

```
var num = -2;
```

```
var result = num > 0 ? "positive" : "non-positive";  
console.log(result);
```

و خروجی زیر نمایش داده می شود.

```
non-positive
```

## عملگرهای نوع

### عملگر typeof

این عملگر، یک عملگر یکانی بوده و نوع داده ی عملوند را برگشت می دهد. به مثال زیر نگاه کنید.

```
var num = 12  
console.log(typeof num); //output: number
```

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

```
//Generated by typescript 1.8.10  
var num = 12;  
console.log(typeof num); //output: number
```

و خروجی زیر نمایش داده می شود.

```
number
```

### instanceof

با کمک این عملگر می توان مشخص کرد که آیا یک شیء دارای نوع مشخصی است یا خیر. به این عملگر در بخش کلاس ها می پردازیم.