

### Object.is

```
Object.is(0, ''); //false
Object.is(null, undefined); //false
Object.is([1], true); //false
Object.is(NaN, NaN); //true
```

### Object.is

x	y	==	===	Object.is
undefined	undefined	true	true	true
null	null	true	true	true
true	true	true	true	true
false	false	true	true	true
'foo'	'foo'	true	true	true
0	0	true	true	true
+0	-0	true	true	false
0	false	true	false	false
==	false	true	false	false
==	0	true	false	false
'0'	0	true	false	false
'17'	17	true	false	false
[1, 2]	'1,2'	true	false	false
new String('foo')	'foo'	true	false	false
null	undefined	true	false	false
null	false	false	false	false
undefined	false	false	false	false
{ foo: 'bar' }	{ foo: 'bar' }	false	false	false
new String('foo')	new String('foo')	false	false	false
0	null	false	false	false
0	NaN	false	false	false
'foo'	NaN	false	false	false
NaN	NaN	false	false	true

### Async & Await

زمانی که لازم است برای رسیدن پاسخی از سمت سرور صبر کند؛ عملیات ناهنگام است

#### Callback

یکی از رایج‌ترین و قدیمی‌ترین راه‌های مدیریت عملیات ناهنگام استفاده از توابع کالباک هست

وقتی عملیات ناهنگام به پایان رسید، نتیجه‌ی این عملیات برای پردازش می‌تونه به یک تابع فرستاده بشه که به این تابع میگن کالباک

#### Callback

خب اینجا یکم کار سخت میشه. چون وقتی یک یا چند تابع کالباک درون هم دیگه قرار می‌گیرن، علاوه بر اینکه ساختار کدهای ما زشت میشن، کدها تو در تو میشن و کار با اطلاعات و متغیرها و نهایتاً مدیریت کد سخت میشه که به این قضیه میگن جهنم کالباک

#### (Callback Hell)

### Pro

#### Closure

```
function init() {
  var counter = 0;

  function jump_counter() {
    counter++;

    return counter;
  }

  return jump_counter;
}
```

```
var jump = init();
```

```
jump();
```

```
jump();
```

```
jump();
```

کلاژر، یک تابعی هست که توی یک تابع دیگه تعریف میشه که می‌تونه علاوه بر متغیرهای حوزه خودش، به متغیرهای حوزه تابع بیرونی هم دسترسی داشته باشه.

#### this

فقط نحوه فراخوانی اون تابع هست که مقدار دیس رو تعیین می‌کنه

#### this

```
var name = "David";
```

```
var person = {
  name: "Ali",
  getName: function() {
    console.log(this.name);
  }
}
```

```
var getName = person.getName;
```

```
getName(); // David
```

```
person.getName(); // Ali
```

### Call() & Apply()

توی کال پارامترها رو باید بصورت تک تک پاس بدیم و توی اپلای همیشه بصورت آرایه پاس داد.

### this

```
"use strict";
```

در این حالت دیس دیگر گلوبال نیس و آندیفاین است

#### Arrow Function

مقدار دیس توی توابع معمولی به آبجکتی اشاره می‌کنه که داره اون تابع رو اجرا می‌کنه. اما توی ارو فانکشن ها مقدار دیس به صاحب (حوزه) آبجکتی اشاره می‌کنه که ارو فانکشن توی اون تعریف شده



By **Sadaf Niknam**  
(sadaf.niknaam)  
[cheatography.com/sadaf-niknaam/](https://cheatography.com/sadaf-niknaam/)

Not published yet.  
Last updated 23rd March, 2020.  
Page 2 of 2.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>