

Square Brackets Javascript Object Key

Asked 4 years ago Active 2 months ago Viewed 19k times



Can anyone explain how the why/how the below method of assigning keys in javascript works?

77

```
a = "b"  
c = {[a]: "d"}
```



return:



16

```
Object {b: "d"}
```

javascript

edited Sep 11 '15 at 4:47



Tushar

70.6k

11

106

134

asked Sep 11 '15 at 4:40



lcharbon

734

1

6

14

@Tushar: There is no call to `Object` here. That's completely unrelated. – [Felix Kling](#) Sep 11 '15 at 4:52

@FelixKling Array is treated as object, `typeof []` – [Tushar](#) Sep 11 '15 at 4:53

@Tushar: arrays are objects, but there is no array here and, again, no call to `Object`. – [Felix Kling](#) Sep 11 '15 at 4:54



Yes, but while `typeof [] == "object"`, `Array.prototype.constructor != Object.prototype.constructor`, nor is the behavior the same. – [Sean Vieira](#) Sep 11 '15 at 5:04

4 Answers



It's the new ES2015 (the EcmaScript spec formally known as ES6) [computed property name syntax](#). It's a shorthand for the

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
var a = "b"  
var c = {[a]: "d"}
```

is syntactic sugar for:

```
var a = "b"  
var c = {}  
c[a] = "d"
```

answered Sep 11 '15 at 4:46



[Sean Vieira](#)

122k 26 244 241

It's not only ES6 – [Tushar](#) Sep 11 '15 at 4:47

@Tushar - where else does this work? It throws for me in a normal ES5 environment (Node 0.10). – [Sean Vieira](#) Sep 11 '15 at 4:51

Really the use of `[]` gives an excellent way to use actual value of variable as **key** / **property** while creating JavaScript **objects**.

11

I'm pretty much statisfied with the above answer and I appreciate it as it allowed me to write this with a little example.

I've executed the code line by line on **Node REPL** (Node shell).

```
> var key = "fullName";    // Assignment  
undefined  
>  
> var obj = {key: "Rishikesh Agrawani"}    // Here key's value will not be used  
undefined  
> obj    // Inappropriate, which we don't want  
{ key: 'Rishikesh Agrawani' }  
>  
> // Let's fix  
undefined  
> var obj2 = {[key]: "Rishikesh Agrawani"}  
undefined  
...
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Nov 2 '18 at 17:08



hygull

5,031

2

17

34

2 Very good examples! They explain this feature better. – JCF Apr 27 at 22:19

1 Thank you very much @JCF. I appreciate your words. – hygull Apr 28 at 3:24



3



```
const animalSounds = {cat: 'meow', dog: 'bark'};
```

```
const animal = 'lion';
```

```
const sound = 'roar';
```

```
{...animalSounds, [animal]: sound};
```

The result will be

```
{cat: 'meow', dog: 'bark', lion: 'roar'};
```

edited Aug 1 at 7:56



teunbrand

3,393

4

23

answered Aug 1 at 6:16



monireh d

62

5



0



Also, only condition to use `[]` notation for accessing or assigning stuff in objects when we don't yet know what it's going to be until evaluation or runtime.

answered Jun 2 at 0:37



Eldiyar Talantbek

602

6

8

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).