

How do I remove a key from a JavaScript object? [duplicate]

Asked 9 years, 2 months ago Active 4 months ago Viewed 837k times



1146

This question already has an answer here:

[How do I remove a property from a JavaScript object?](#) 41 answers



Let's say we have an object with this format:



124

```
var thisIsObject= {  
  'Cow' : 'Moo',  
  'Cat' : 'Meow',  
  'Dog' : 'Bark'  
};
```

I wanted to do a function that removes by key:

```
removeFromObjectByKey('Cow');
```

javascript

edited Mar 11 '17 at 13:50



Peter Mortensen

14.5k 19 89 118

asked Aug 11 '10 at 4:59



Martin Ongtango

8,089 13 52 82

marked as duplicate by [Paul Roub](#) javascript Dec 14 '15 at 21:32

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

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

- 1 Yeah just some confusion with terminology I think, ie it's Javascript not JQuery, and it's an object not array (OP may come from other languages with associative arrays). – [thomasrutter](#) Aug 11 '10 at 5:12

3 Answers



1999



The [delete](#) operator allows you to remove a property from an object.

The following examples all do the same thing.

```
// Example 1
var key = "Cow";
delete thisIsObject[key];

// Example 2
delete thisIsObject["Cow"];

// Example 3
delete thisIsObject.Cow;
```

If you're interested, read [Understanding Delete](#) for an in-depth explanation.

edited Apr 3 '14 at 15:19

answered Aug 11 '10 at 5:01



[jessegavin](#)

55k 25 131 160

- 28 If you are looping over keys in an object, and delete them if they match a certain value, does this affect the key index while you are looping over it? – [CMAury](#) Feb 4 '13 at 15:57
- 10 It seems to me that running through that object with `for (var key in object)` will still match that key and trying to use `object[key]` in the loop will throw an error. (or return `undefined`) – [Sami](#) Mar 3 '13 at 21:47
- 7 Beware that IE8 may throw an exception when using `delete` in certain circumstances. See stackoverflow.com/questions/1073414/ – [nullability](#) Apr 9 '13 at 20:58
- 4 except it leaves an undefined in the array which makes it difficult to iterate. – [Josh Bedo](#) Jul 24 '14 at 15:53
- 3 Does anyone know the runtime of this operation? – [AjaxLeung](#) May 31 '16 at 18:31

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



146

If you are using Underscore.js or Lodash, there is a function 'omit' that will do it.

<http://underscorejs.org/#omit>

```
var thisIsObject= {
  'Cow' : 'Moo',
  'Cat' : 'Meow',
  'Dog' : 'Bark'
};
_.omit(thisIsObject, 'Cow'); //It will return a new object

=> {'Cat' : 'Meow', 'Dog' : 'Bark'} //result
```

If you want to modify the current object, assign the returning object to the current object.

```
thisIsObject = _.omit(thisIsObject, 'Cow');
```

With pure JavaScript, use:

```
delete thisIsObject['Cow'];
```

Another option with pure JavaScript.

```
thisIsObject.cow = undefined;

thisIsObject = JSON.parse(JSON.stringify(thisIsObject ));
```

edited Feb 8 '17 at 4:24



Pang

7,274 16 68 108

answered Mar 1 '15 at 19:02



Mohammed Safeer

13k 6 59 70

23 Downvoted. The thing is that both in underscore and lodash `_.omit` returns new object, does not modify the current one. So, this is slightly a different thing. – [shabunc](#) Jul 1 '15 at 1:36

5 @shabunc The same with pure javascript. `delete o.property` does a lot more harm than good behind the scenes, as it changes `o`'s hidden class and makes it a generic slow object. – [Mateusz Nowak](#) Dec 20 '15 at 10:09

1 @MateuszNowak no it is not the same. You are talking about implementation details which can vary, but in javascript per se, `delete` modifies object

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

35 Upvoted. A new immutable object is preferable in many cases. – [Dominic](#) Aug 13 '17 at 19:50



It's as easy as:

75

```
delete object.keyname;
```



or

```
delete object["keyname"];
```

edited Jun 6 at 13:22



[kalehmann](#)

3,347 6 15 29

answered Dec 15 '13 at 21:38



[ANIL MIRGE](#)

769 5 2

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