

Object.keys()

Share:  Twitter  Facebook  Google+

The **Object.keys()** method returns an array of a given object's own enumerable properties, in the same order as that provided by a **for...in** loop (the difference being that a for-in loop enumerates properties in the prototype chain as well).

Syntax

```
Object.keys(obj)
```

Parameters

obj
The object whose enumerable own properties are to be returned.

Description

Object.keys() returns an array whose elements are strings corresponding to the enumerable properties found directly upon **object**. The ordering of the properties is the same as that given by looping over the properties of the object manually.

Examples

```
1  var arr = ['a', 'b', 'c'];
2  console.log(Object.keys(arr)); // console: ['0', '1', '2']
3
4  // array like object
5  var obj = { 0: 'a', 1: 'b', 2: 'c' };
6  console.log(Object.keys(obj)); // console: ['0', '1', '2']
7
8  // array like object with random key ordering
9  var an_obj = { 100: 'a', 2: 'b', 7: 'c' };
10 console.log(Object.keys(an_obj)); // console: ['2', '7', '100']
11
12 // getFoo is property which isn't enumerable
13 var my_obj = Object.create({}, { getFoo: { value: function() { return this.foo; } } });
14 my_obj.foo = 1;
15
16 console.log(Object.keys(my_obj)); // console: ['foo']
```

If you want all properties, even not enumerables, see [Object.getOwnPropertyNames\(\)](#).

Notes

In ES5, if the argument to this method is not an object (a primitive), then it will cause a **TypeError**. In ES6, a non-object argument will be coerced to an object.

```
1  Object.keys("foo");
2  // TypeError: "foo" is not an object (ES5 code)
3
4  Object.keys("foo");
5  // ["0", "1", "2"] (ES6 code)
```

Polyfill

To add compatible **Object.keys** support in older environments that do not natively support it, copy the following snippet:

```
1  // From https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/keys
```

```
2  if (!Object.keys) {
3      Object.keys = (function() {
4          'use strict';
5          var hasOwnProperty = Object.prototype.hasOwnProperty,
6              hasDontEnumBug = !({ toString: null }).propertyIsEnumerable('toString'),
7              dontEnums = [
8                  'toString',
9                  'toLocaleString',
10                 'valueOf',
11                 'hasOwnProperty',
12                 'isPrototypeOf',
13                 'propertyIsEnumerable',
14                 'constructor'
15             ],
16             dontEnumsLength = dontEnums.length;
17
18         return function(obj) {
19             if (typeof obj !== 'object' && (typeof obj !== 'function' || obj === null)) {
20                 throw new TypeError('Object.keys called on non-object');
21             }
22
23             var result = [], prop, i;
24
25             for (prop in obj) {
26                 if (hasOwnProperty.call(obj, prop)) {
27                     result.push(prop);
28                 }
29             }
30
31             if (hasDontEnumBug) {
32                 for (i = 0; i < dontEnumsLength; i++) {
33                     if (hasOwnProperty.call(obj, dontEnums[i])) {
34                         result.push(dontEnums[i]);
35                     }
36                 }
37             }
38             return result;
39         };
40     }());
41 }
```

Please note that the above code includes non-enumerable keys in IE7 (and maybe IE8), when passing in an object from a different window.

For a simple Browser Polyfill, see [JavaScript - Object.keys Browser Compatibility](#).

Specifications

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) The definition of 'Object.keys' in that specification.	<div><div></div>Standard</div>	Initial definition. Implemented in JavaScript 1.8.5.
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'Object.keys' in that specification.	<div><div></div>Standard</div>	

Browser compatibility

	Desktop	Mobile				
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari	
Basic support	5	4.0 (2.0)	9	12	5	

See also

- [Enumerability and ownership of properties](#)
- [Object.prototype.propertyIsEnumerable\(\)](#)
- [Object.create\(\)](#)

- `Object.getOwnPropertyNames()`