MDN Plus now available in your country! Support MDN and make it your own. Learn more

mdn web docs

Object.values()

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

Try it

JavaScript Demo: Object.values()

```
const object1 = {
  a: 'somestring',
  b: 42,
  c: false
};

console.log(Object.values(object1));
// expected output: Array ["somestring", 42, false]

```

Run ›

Reset

```
> Array ["somestring", 42, false]
```

Syntax

```
Object.values(obj)
```

Parameters

`obj`

The object whose enumerable own property values are to be returned.

Return value

An array containing the given object's own enumerable property values.

Description

`Object.values()` returns an array whose elements are the enumerable property values found on the object. The ordering of the properties is the same as that given by looping over the property values of the object manually.

Examples

Using Object.values

```
const obj = { foo: 'bar', baz: 42 };
console.log(Object.values(obj)); // ['bar', 42]

// Array-like object
const arrayLikeObj1 = { 0: 'a', 1: 'b', 2: 'c' };
console.log(Object.values(arrayLikeObj1 )); // ['a', 'b', 'c']

// Array-like object with random key ordering
// When using numeric keys, the values are returned in the keys' numerical order
const arrayLikeObj2 = { 100: 'a', 2: 'b', 7: 'c' };
console.log(Object.values(arrayLikeObj2 )); // ['b', 'c', 'a']

// getFoo is property which isn't enumerable
const myObj = Object.create({}, { getFoo: { value() { return this.foo; } } });
myObj.foo = 'bar';
console.log(Object.values(myObj)); // ['bar']

// non-object argument will be coerced to an object
console.log(Object.values('foo')); // ['f', 'o', 'o']
```

Specifications

Specification
ECMAScript Language Specification # sec-object.values

Browser compatibility

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Chrome Android	Firefox for Android	Opera Android
values	Chrome 54	Edge 14	Firefox 47	Internet Explorer No	Opera 41	Safari 10.1	Chrome Android 54	Firefox for Android 47	Opera Andr

Full support No support User must explicitly enable this feature.

See also

- Polyfill of `Object.values` in `core-js`

- Enumerability and ownership of properties
- `Object.keys()`
- `Object.entries()`
- `Object.prototype.propertyIsEnumerable()`
- `Object.create()`
- `Object.getOwnPropertyNames()`

Last modified: Aug 6, 2022, by MDN contributors