Object.defineProperties()

`Object.defineProperties()` 方法直接在一个对象上定义新的属性或修改现有属性，并返回该对象。

语法

```
Object.defineProperties(obj, props)
```

参数

`obj`
在其上定义或修改属性的对象。

`props`
要定义其可枚举属性或修改的属性描述符的对象。对象中存在的属性描述符主要有两种：数据描述符和访问器描述符（更多详情，请参阅 `Object.defineProperty()` ）。描述符具有以下键：

`configurable`
`true` 当且仅当该属性描述符的类型可以被改变并且该属性可以从对应对象中删除。
默认为 **`false`**

`enumerable`
`true` 当且仅当在枚举相应对象上的属性时该属性显现。
默认为 **`false`**

`value`
与属性关联的值。可以是任何有效的JavaScript值（数字，对象，函数等）。
默认为 **`undefined`**.

`writable`
`true` 当且仅当与该属性相关联的值可以用assignment operator改变时。
默认为 **`false`**

`get`

作为该属性的 getter 函数，如果没有 getter 则为 `undefined` 。函数返回值将被用作属性的值。
默认为 **`undefined`**

`set`
作为属性的 setter 函数，如果没有 setter 则为 `undefined` 。函数将仅接受参数赋值给该属性的新值。
默认为 **`undefined`**

返回值

传递给函数的对象。

描述

`Object.defineProperties` 本质上定义了 `obj` 对象上 `props` 的可枚举属性相对应的所有属性。

例子

```
var obj = {};
Object.defineProperties(obj, {
  'property1': {
    value: true,
    writable: true
  },
  'property2': {
    value: 'Hello',
    writable: false
  }
  // etc. etc.
});
```

Polyfill

假设一个原始的执行环境，所有的名称和属性都引用它们的初始值， `Object.defineProperties` 几乎完全等同于（注意 `isCallable` 中的注释）以下JavaScript中的重新实现：

```
function defineProperties(obj, properties) {
  function convertToDescriptor(desc) {
    function hasProperty(obj, prop) {
      return Object.prototype.hasOwnProperty.call(obj, prop);
```

```
    }

    function isCallable(v) {
      // NB: modify as necessary if other values than functions are calla
      return typeof v === 'function';
    }

    if (typeof desc !== 'object' || desc === null)
      throw new TypeError('bad desc');

    var d = {};

    if (hasProperty(desc, 'enumerable'))
      d.enumerable = !!desc.enumerable;
    if (hasProperty(desc, 'configurable'))
      d.configurable = !!desc.configurable;
    if (hasProperty(desc, 'value'))
      d.value = desc.value;
    if (hasProperty(desc, 'writable'))
      d.writable = !!desc.writable;
    if (hasProperty(desc, 'get')) {
      var g = desc.get;

      if (!isCallable(g) && typeof g !== 'undefined')
        throw new TypeError('bad get');
      d.get = g;
    }
    if (hasProperty(desc, 'set')) {
      var s = desc.set;
      if (!isCallable(s) && typeof s !== 'undefined')
        throw new TypeError('bad set');
      d.set = s;
    }

    if (('get' in d || 'set' in d) && ('value' in d || 'writable' in d))
      throw new TypeError('identity-confused descriptor');

    return d;
  }

  if (typeof obj !== 'object' || obj === null)
    throw new TypeError('bad obj');

```

```
  properties = Object(properties);

  var keys = Object.keys(properties);
  var descs = [];

  for (var i = 0; i < keys.length; i++)
    descs.push([keys[i], convertToDescriptor(properties[keys[i]])]);

  for (var i = 0; i < descs.length; i++)
    Object.defineProperty(obj, descs[i][0], descs[i][1]);

  return obj;
}
```

规范

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) Object.defineProperties	ST Standard	Initial definition. Implemented in JavaScript 1.8.5
ECMAScript 2015 (6th Edition, ECMA-262) Object.defineProperties	ST Standard	
ECMAScript Latest Draft (ECMA-262) Object.defineProperties	LS Living Standard	

浏览器兼容性

Desktop Mobile

Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
Basic Support	5	(Yes)	4	9	11.6	5

相关链接

- `Object.defineProperty()`
- `Object.keys()`
- 属性的可枚举性和所有权