# 对象

## 对象的三种类型的属性

①命名数据属性

一个确定值的属性

②命名访问器属性

通过getter和setter进行读取和赋值

③内部属性

由js引擎内部使用的属性，不能通过js代码直接访问到，不过可以通过一些间接的读取和设置。

如每个对象都有的内部属性[[Prototype]],可以通过Object.getPrototypeOf()方法间接读取。

如 [[value]]

[[Writable]]

[[Get]]

[[Set]]

[[Enumerable]]

[[Configurable]]

[[Prototype]] 对象的原型

[[Extensible]] 对象是否可以扩展（是否能添加新属性）

[[DefineOwnProperty]] 定义一个属性的内部方法

[[Put]] 为一个属性赋值的内部方法

## 属性的特性

对于命名数据属性和命名访问器属性，有四个客观

1. 命名数据属性特有的特性

属性的值的[[Value]]特性和控制属性的值是否可以修改的[[Writable]]特性

1. 命名访问器属性特有的特性

存储着getter方法的[[Get]]和存储着setter方法的[[Set]]

1. 两种属性都有的特性

如果一个属性是不可枚举的，则一些操作下，这个属性是不可见的，比如for...in和Object.keys，那么可以通过[[Enumerable]]特性来设置；如果一个属性是不可配置的，则该属性的所有特性（[[Value]]）都不可改变，那么可以通过[[Configurable]]特性来设置

Object.defineProperty()语法

## 属性描述符

分为数据描述符 和 存取描述符

描述符必须是两者之一，而不能同时都是。

|  |  |  |
| --- | --- | --- |
| 属性名 | 默认值 |  |
| value | undefined |  |
| get | undefined |  |
| set | undefined |  |
| writable | false | 是否可写 |
| enumerable | false | 是否可枚举 |
| configurable | false | 决定属性是否能被删除，以及除值修改（writable value）外是否可以被修改 |

通过Object.defineProperty指定的属性，不设定，则取上表的默认值，但是注意，对同一个属性多次define，只会对响应描述符覆盖。

通过字面量， [] 或 . 指定的属性描述符默认都是true。（数据描述符）

除了在Object.defineProperty(obj, prop, {

enumerable: false,

configurable: false,

writable: false,

value: 'static'

}) 里使用属性描述符，还可以在Object.getOwnPropertyDescriptor(obj, prop)和Object.create()中使用

## 属性定义和属性赋值

定义属性是通过内部方法操作的

[[DefineOwnProperty]](P, Desc, Throw)

P: 定义的属性名称

Throw: 定义被拒绝时，是否要抛出异常（是否静默失败）

调用 [[DefineOwnProperty]]时，执行下面步骤

1. 如果this没有名为P的属性

如果this可以扩展，创建P这个自身属性，否则拒绝

1. 如果已有P这个属性

不可配置：

将一个数据属性转换为访问器属性，或相反。

改变[[configurable]] [[Enumerable]]

定义属性的函数

defineProperty 和 defineProperties

definePropery(obj, prop,desc)

defineProperties(obj,{

a:{},

b:{}

})

Object.defineProperty(obj, prop, desc)

----> 内部引擎 obj.[[DefineOwnProperty]]( prop, desc, true)

属性赋值

[[put]](P, Value, Throw)

内部引擎 obj.[[put]](prop, value, isStrictMoDeOn)

isStrictMoDeOn: 是否是严格模式： 只有严格模式下才有可能抛出异常

## 使用属性描述符的姿势

## 注意

循环体中使用复杂对象声明，会重复建立声明对象