Object类型

var o = new Object();

仅仅创建 Object 的实例并没有什么用处，但关键是要理解一个重要的思想：即在 ECMAScript 中，

（就像 Java 中的 java.lang.Object 对象一样） Object 类型是所有它的实例的基础。换句话说，

Object 类型所具有的任何属性和方法也同样存在于更具体的对象中。

Object 的每个实例都具有下列属性和方法。

□ constructor ：保存着用于创建当前对象的函数。对于前面的例子而言，构造函数（constructor）

就是 Object() 。

□ hasOwnProperty(propertyName) ：用于检查给定的属性在当前对象实例中（而不是在实例

的原型中）是否存在。其中，作为参数的属性名（ propertyName ）必须以字符串形式指定（例

如： o.hasOwnProperty("name") ）。

□ isPrototypeOf(object) ：用于检查传入的对象是否是传入对象的原型（第 5 章将讨论原

型）。

□ propertyIsEnumerable(propertyName) ：用于检查给定的属性是否能够使用 for-in 语句

（本章后面将会讨论）来枚举。与 hasOwnProperty() 方法一样，作为参数的属性名必须以字符

串形式指定。

□ toLocaleString() ：返回对象的字符串表示，该字符串与执行环境的地区对应。

□ toString() ：返回对象的字符串表示。

□ valueOf() ：返回对象的字符串、数值或布尔值表示。通常与 toString() 方法的返回值

相同。

由于在 ECMAScript 中 Object 是所有对象的基础，因此所有对象都具有这些基本的属性和方法。

从技术角度讲，ECMA-262 中对象的行为不一定适用于 JavaScript 中的其他对象。浏览器环境中的对象，比如 BOM 和 DOM 中的对象，都属于宿主对象，因为它们是由宿主实现提供和定义的。ECMA-262 不负责定义宿主对象，因此宿主对象可能会也可能不会继承 `Object`。