1. 对象
2. 数据类型

基本类型：string number boolean null undefined symbol(es6)

复杂数据类型：object

* 1. typeof null == “object”
  2. 函数和数组都是对象

1. 内置对象

String Number Boolean Object Function Array Date RegExp Error

var strPrimitive="I am a string";

console.log(typeof strPrimitive);//string

console.log(strPrimitive instanceof String);//false

var strObject=new String("I am a string");

console.log(typeof strObject);//object

console.log(strObject instanceof String);//true

console.log(Object.prototype.toString.call(strObject));//[object String]

如果要在字面量上执行一些操作，比如获取长度，需要将其转换为对象，自动(隐式)转换

1. 属性

存储在对象容器内部的是这些属性的名称，它们就像(引用)指针一样，指向这些值真正的存储位置。

在对象中，属性名永远都是字符串，不是字符串的会自动转换为字符串

es6中新增了可计算属性名，最常见的应用是symbol

var prefix="name"

var myObject={

[name+"1"]:"hello",

[name+2]:"world"

}

console.log("name1");//name2

console.log("name2");//name2

1. 属性描述符

var myObj={

a:"2"

}

console.log(Object.getOwnPropertyDescriptor(myObj,"a"));

//{value: "2", writable: true, enumerable: true, configurable: true}

Object.defineProperty(myObj,"a",{

value:3,

writable:true

})

console.log(myObj.a);//3

Writable是否可修改属性的值

Configurable是否可配置 设置为false单向操作不能撤销 还禁止删除这个属性

Enumerable是否可枚举

1. 不变性

Ⅰ对象常量

writable:false configurable:false

Ⅱ禁止扩展

Object.preventExtensions(myObj);//不能添加新属性

Object.isExtensible(myObj);//是否可扩展

Ⅲ密封

Object.seal(myObj)) //不能添加、配置、删除现有属性，可以修改

Object.isSealed(myObj);//是否被密封

Ⅳ冻结

Object.freeze(myObj))

Object.isFrozen(myObj);//是否被冻结

1. getter setter

如果有访问描述符，会忽略value和writable属性

var myObj={

get a(){return 2;}

}

myObj.a=3;

console.log(myObj.a);//2

1. 存在性

var myObj={a:"2"}

console.log("a" in myObj);//true

console.log(myObj.hasOwnProperty("a"));//true

in会检查属性是否存在对象及其原型链上，hasOwnProperty()只会检查属性是否存在其对象上

propertyIsEnumerable()会检查属性是否存在其对象(不包括原型链)上，而且可枚举

Object.keys()返回一个数组，包含所有的可枚举属性，只在对象本身上

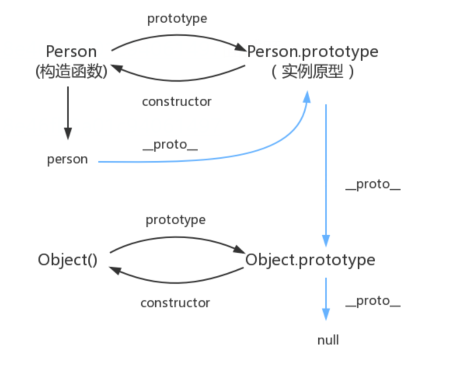
Object.getOwnPropertyNames(obj)只查找对象本身

1. 遍历

forEach every some

<https://www.cnblogs.com/shapeY/archive/2017/09/27/7600460.html>

1. 博客
2. 原型和原型链



* 1. person.constructor===Person

person中并没有constructor属性，当不能读取到constrctor属性时，会沿原型链到Person.prototype中查找

* 1. 当使用obj.\_\_proto\_\_时，可以理解为返回了Object.getPrototypeOf()

1. 词法作用域

函数的作用域在函数定义时确定

var value=1;

function foo(){

console.log(value);

}

function bar(){

var value=2;

foo();

}

bar();//1