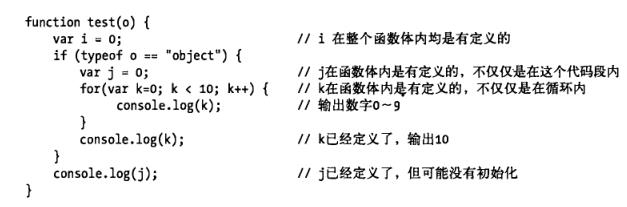
**Javascript总结**

目录

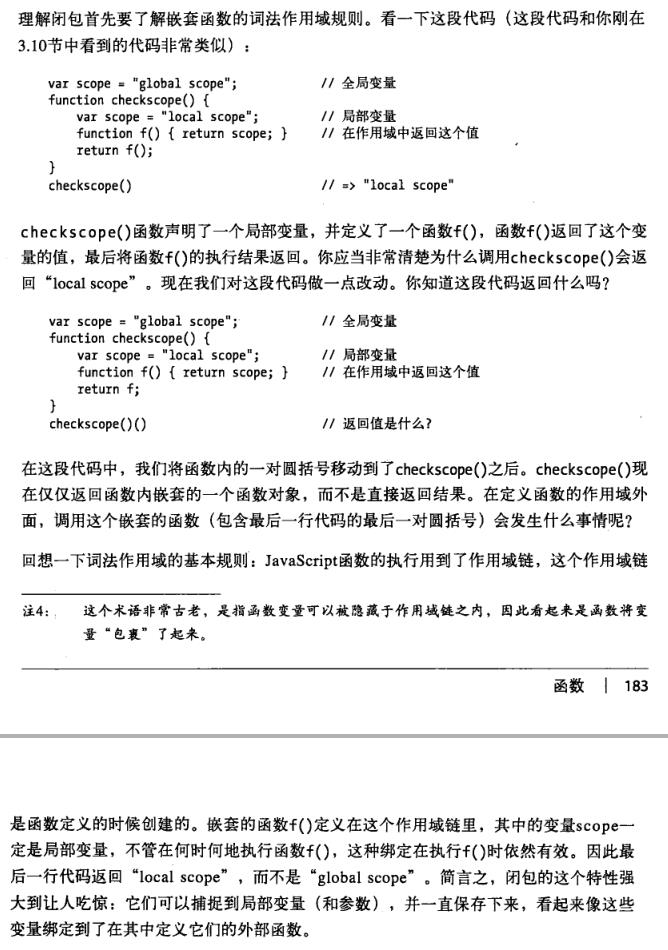
1. 变量、作用域链与闭包问题-----------------------------2
2. 对象与引用类型----------------------------------------4
3. 面向对象的程序设计----------------------------------10
4. 正则表达式----------------------------------------------
5. Window对象-------------------------------------------
6. 脚本化文档（dom）------------------------------------
7. 脚本化css-----------------------------------------------
8. 事件处理------------------------------------------------
9. 变量、作用域链与闭包问题

**作用域：**一个变量的作用域是程序源代码中定义这个变量的区域。全局变量拥有全局作用域；而在函数内声明的变量只在函数体内有定义，它们是局部变量，作用域是局部性的。



**作用域链：**由一个全局变量组成。作用域链的用途，是保证对执行环境有权访问的所有变量和函数的有序访问。作用域链的活动对象只能从所在环境一级一级往外查找，而不能往往内部作用域查找。

**闭包：**闭包是指有权访问另一个函数作用域中的变量的函数。创建闭包的常见方式，就是在一个函数内部创建另一个函数。



1. 对象与引用类型
2. **Object类型**
3. 创建对象的方法：

第一种是使用new操作符后跟Object构造函数，如下：

var Person = new Object( )；

Person.name = “Robin”;

Person.age = 19;

第二种是使用对象字面量表示法，如下：

var person = {

name : “Robin”;

age : 19;

};

1. 访问属性的方法：

一般来说使用的都是点表示法，不过javascript可以使用方括号表示法来访问对象属性，如下：

person.name;

person[“name”];

1. 对象的方法：

所有对象都有的方法是：toLocaleString()、toString()、valueOf()方法。

1. **Array类型**
2. 创建数组的基本方式有两种。第一种是使用Array构造函数，如下：

var colors = new Array( ); //括号里面可以是数组长度

也可以直接向Array构造函数传递数组中包含的项，如下：

var colors = new Array(“red”, “blue”, “green”);

第二种基本方式是使用数组字面量表示法，如下：

var colors = [“red”,“blue”, “green”];

1. Array类型的方法：
2. 检测数组：

Array.isArray( )方法用于检测括号里面的值是不是数组；

1. 转换方法：

toString()、toLocaleString()、valueOf()方法都是返回用逗号隔开的字符串；

1. 栈方法：

**push()**方法可以接收任意数量的参数，把它们逐个添加到数组末尾，并返回修改后数组的长度。

**pop()**方法从数组末尾移除最后一项，减少数组的length值，并返回移除的项。

1. 队列方法：

**shift()**方法能够移除数组中的第一个项，并返回该项，减少数组的length值。

**unshift()**方法能在数组前端添加任意个项并返回新数组长度。

1. 重排序方法：

reverse()方法会反转数组项的顺序。

**sort()方法**按升序排列数组项。sort()方法会调用每个数组项的toString方法，然后比较字符串。通常需要自己自定义一个函数，将函数名称作为sort()方法的参数进行有效排序。

1. 操作方法：

**concat()方法**可以基于当前数组中的所有项创建一个新数组。在没有传递参数的情况下，它只是简单的复制数组；如果给concat()方法传递了一或多个参数，则将会将参数加到数组副本的末尾。

**slice()方法**能够基于当前数组中的一个或多个项创建一个新数组。它接受一或者两个参数，传递一个参数时，slice()方法返回从该参数指定位置开始到数组末尾的所有项；如果有两个参数，则返回起始位置和结束位置之间的项---但不包括结束位置的项。

**splice()方法**，第一个参数是位置，第二个参数是要删除的项，第三个参数是要插入的项，如果要插入多个项，则传入第四、五以致更多参数。它可以做到删除、插入和替换数组的项。

1. 位置方法：

**indexOf()和lastIndexOf()方法**，两个方法都接收两个参数：要查找的项和表示查找起点位置的索引。

1. 迭代方法：

every():对数组中的每一项运行给定函数。如果函数对每一项都返回true，则返回true。  
filter():对数组中的每一项运行给定函数。返回该函数会返回true的项组成的数组。  
forEach():对数组中每一项运行给定函数。该函数没有返回值。  
map():对数组中每一项运行给定函数。返回每次函数调用的结果组成的函数。  
some():对数组中每一项运行给定函数。如果函数对 任一项返回true，则返回true

1. **Date类型**
2. 创建一个日期对象，使用new操作符和Date构造函数即可：

var now = new Date( );

1. **Number类型**
2. **toString()、toLocaleString()方法**返回字符串形式的数值。其中toString()方法传递一个参数时表示返回几进制的形式。
3. **valueOf()方法**返回对象表示的基本类型的数值。
4. **toFixed()方法**会按照参数指定的小数位返回数值的字符串表示。
5. **toExponential()方法**返回一指数表示法表示的数值的字符串形式。
6. toPrecision()方法返回固定大小格式，或者返回指数格式，看哪一个更适合。
7. **String类型**
8. length属性，返回字符串长度；
9. 字符方法：

**charAt()方法**以单字符字符串的形式返回给定位置的那个字符；

1. 字符串操作方法：

**concat()方法**用于将一个或多个字符串拼接起来，返回拼接得到的新字符串。

**slice()、substr()、substring()方法**都返回被操作字符串的一个子字符串，都接受一或两个参数。第一个参数指定字符的位置，第二个参数表示到哪里结束。 slice()、substring()方法第二个参数指定的是子字符串最后一个字符的后一个位置，substr()的第二个参数是指定返回的字符的个数。

1. 字符串位置方法：

indexOf()、lastIndexOf()方法查找给定字符，返回字符在字符串的位置，如果没找到则返回-1。

1. trim()方法：

这个方法会创建一个字符串的副本，删除前置和后缀的所有空格，然后返回结果。

1. 字符串大小写转换方法：

toLowerCase()、toLocaleLowerCase()、toUpperCase()、

toLocaleUpperCase()。

1. 字符串的模式匹配方法
2. localeCompare()方法：

这个方法比较两个字符串。

1. **Math对象**
2. **max()和min()方法**：

用于确定一组数值中的最大值和最小值。

找到数组中的最大值或最小值，可以像下面使用：

var values = [1,2,3,4,5];

var max = Math.max.apply(Math,values);

1. 舍入方法：

**Math.ceil()**执行向上舍入；

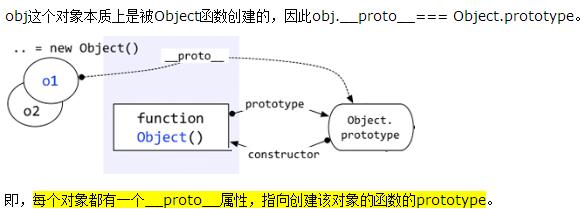
**Math.floor()**执行向下舍入；

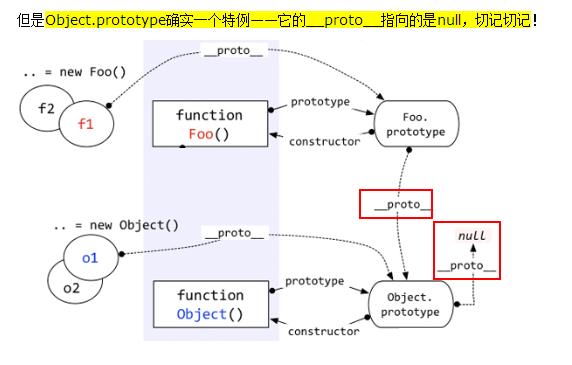
**Math.round()**执行标准舍入，即四舍五入；

1. random()方法：

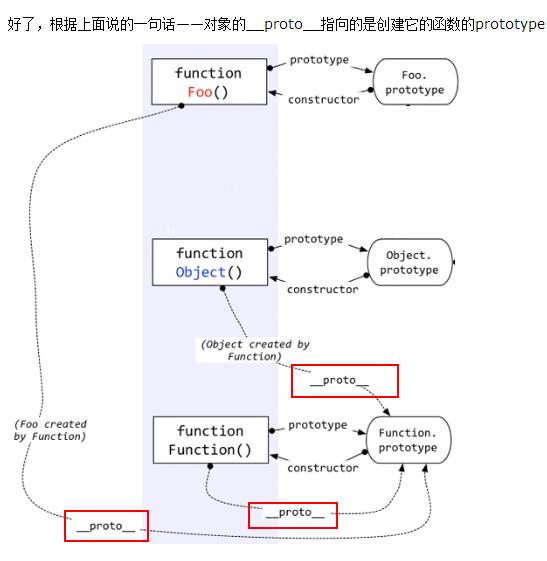
返回一个大于等于0小于1的随机数。

1. 面向对象的程序设计
2. **原型与原型链问题**（重点理解）
3. 函数与对象之间的关系：**对象是通过函数创建的，而函数又是一种对象。**
4. **每个函数**都有一个默认属性：**prototype**。 prototype是一个对象，默认只有constructor属性，这个属性指向函数本身。
5. **每个对象**都有一个 **\_proto\_** 属性，称为隐式原型，是一个隐藏的属性。该属性指向创建该对象的函数的prototype。 如图：





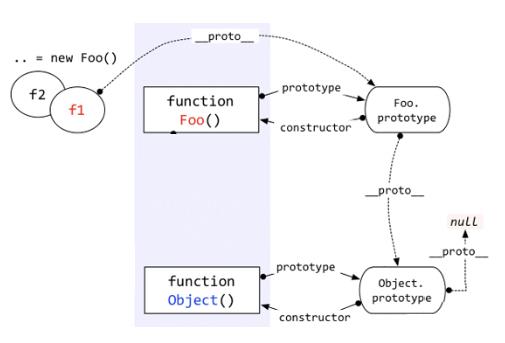
**函数也是一种对象，函数也有\_\_proto\_\_。**如图：



自定义函数Foo.\_\_proto\_\_指向Function.prototype，Object.\_\_proto\_\_指向Function.prototype，唉，怎么还有一个……Function.\_\_proto\_\_指向Function.prototype？这不成了循环引用了？

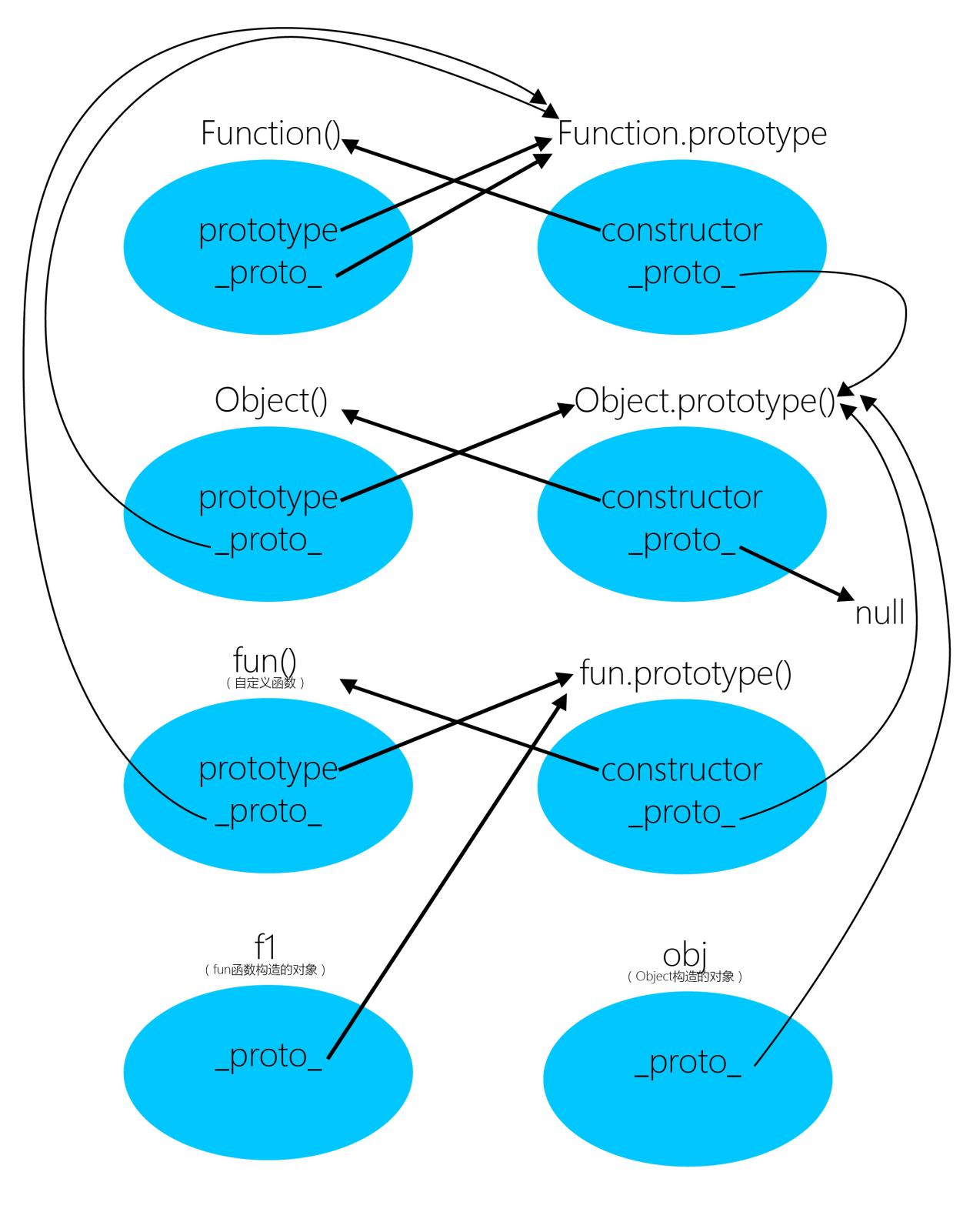
对！是一个环形结构。

（4）instanceof



Instanceof的判断队则是：沿着A的\_\_proto\_\_这条线来找，同时沿着B的prototype这条线来找，如果两条线能找到同一个引用，即同一个对象，那么就返回true。如果找到终点还未重合，则返回false。

原型与原型链图：



1. window对象
2. **window对象**
3. 窗口位置：screenLeft和screenTop以及screenX和screenY属性。
4. 窗口大小：innerWidth、innerHeight、outerWidth和outerHeight属性。
5. 打开窗口：window.open()。
6. 间歇调用和超时调用：setTimeout()和setInterval()， 清除为：clearTimeout()和cleatInterval()。
7. 系统对话框：alert()、confirm()、prompt()。
8. **location对象**
9. **navigator对象**
10. **history对象**
11. go():在用户的历史记录中任意跳转。
12. forward()和back().
13. 脚本化文档（dom）

一、