匿名函数：没有函数名的函数

函数的定义三种方式：

1、常规方式

function f(x){

return x\*x;

} //实质是将函数的主体赋值给 f，即方式3

2、使用Function构造函数，把参数列表、函数体都作为字符串

var V = new Function(‘x’,’return x\*x;’);

3、函数字面量（直接量 ）

var V = function(x) {return x\*x;};

后两种方式都是把函数赋值给变量V，而这个函数没有名称，即匿名函数

函数字面量和Function()构造函数的区别：

函数字面量是一个匿名函数，但语法允许为其指定任意一个函数名，当写递归函数时可以调用它自己，使用Function()构造函数则不行。var f = function fact(x) {

if (x < = 1)

return 1;

else

return x \* fact(x - 1);

};

Function()构造函数允许运行时Javascript代码动态的创建和编译。在这个方式上它类似全局函数eval()。

Function()构造函数每次执行时都解析函数主体，并创建一个新的函数对象。所以当在一个循环或者频繁执行的函数中调用Function()构造函数的效率是非常低的。相反，函数字面量却不是每次遇到都重新编译的。

用Function()构造函数创建一个函数时并不遵循典型的作用域，它一直把它当作是顶级函数来执行。

var y = "global";

function constructFunction() {

var y = "local";

return new Function("return y"); // 无法获取局部变量，总是会在顶层函数执行

}

alert(constructFunction()()); // 输出 "global"

调用函数：

f(2); //小括号的作用是将f所指向的函数主体作为函数执行，如果没有小括号，将输出函数主体语句。

匿名函数的创建

1、与上第3种方式相同

2、

(function(x,y){ //红色第一个括号是执行函数

alert(x+y);

})(2,3); //蓝色第二个括号是用来传递参数(也可以放在红色括号内)

等同如下：

function f(x,y){

alert(x+y);

};

f(2,3);

匿名函数没有函数名，也没有指向它的变量，不能像有名称的函数一样可以在任何地方调用，匿名函数必需在定义后立即执行。

小括号的作用就是执行函数，也可以传递参数，所以匿名函数可以使用小括号来调用和执行。

(function(参数){……….})(调用参数);

(function(参数){……….}(调用参数));