1. **连字符串转化为驼峰**

vars1 ="foo-style-css";

s1 = s1.replace(//-(/w)/g, function(all, letter){

returnletter.toUpperCase();

});

1. **自运行函数**

(function(funArg){

Var num=30;

funArg()

})(foo)

1. **循环**

var i=0;

for (i;i<50;i++) {

document.write(i+"</br>");

}

1. **call和apply**

function add(a,b)

{

alert(a+b);

}

function sub(a,b)

{

alert(a-b);

}

add.call(sub,3,1); 用add来替换sub方法

apply()方法：接受两个参数，一个是在其中运行函数的作用域，另一个是参数数组（可以是Array的实例，也可以是arguments对象）。

语法：apply(thisObj，[argArray])----传入的数组

定义：应用某一对象的一个方法，用另一个对象替换当前对象。

call()方法：他与apply()方法的区别仅在与接收参数的方式不同。对于call()方法而言，第一个参数是this值没有变化变化的是其余参数都直接传递个函数。换句话说就是在使用call()方法时，传递给函数的参数必须逐个例举出来。

语法：call(thisObj，Object)

定义：调用一个对象的一个方法，以另一个对象替换当前对象。

1. **css左侧宽度固定，右侧自适应**

.left1{

float:left;width:200px;height:auto;border:1px solid red;min-height: 200px;

}

.right1{

height:auto;width:auto;border:1px solid blue;margin-left:200px;min-height: 200px;

}

10233213213213213