**匿名函数**

匿名函数就是指没有名字的函数。

现在我们接触到的函数类型有：字面量的函数，普通函数，构造函数，匿名函数。

function() {

console.log('haha');

}

匿名函数因为没有名字所以不能跟其他函数一样通过函数名调用，匿名函数执行需要通过自己。

匿名函数自执行

(function() {

alert('自执行函数');

})();

// 传参数

(function(a, b) {

console.log(a + b);

})(10, 20);

// 有返回值

var s = (function(a, b) {

return a + b;

})(20, 30);

console.log(s);

**闭包**

闭包就是函数里面嵌套函数，嵌套函数里面使用的变量会一直存储在内存中不会被释放

简单的说，Javascript允许使用内部函数---即函数定义和函数表达式位于另一个函数的函数体内。而且，这些内部函数可以访问它们所在的外部函数中声明的所有局部变量、参数和声明的其他内部函数。当其中一个这样的内部函数在包含它们的外部函数之外被调用时，就会形成闭包。

function fun() {

var a = 10;

return function() {

console.log(++a);

}

}

var s = fun();

s();

s();

s();

循环中的闭包问题

var oLi = document.getElementsByTagName('li');

for (var i = 0; i < oLi.length; i++) {

oLi[i].onclick = function() {

alert(i);

}

}

for (var i = 0; i < oLi.length; i++) {

oLi[i].onclick = (function(i) {

return function() {

alert(i);

}

})(i);

}

for (var i = 0; i < oLi.length; i++) { //4 ()()

(function(i) {

oLi[i].onclick = function() {

alert(i);

}

})(i);

}

面试中的闭包问题

var arr = [];

for (var i = 0; i < 5; i++) {

arr.push(function() {

console.log(i);

})

}

arr[1]();

var arr = [];

for (var i = 0; i < 5; i++) {

arr.push((function(i) {

return function() {

console.log(i);

}

})(i));

}

arr[1]();

**命名空间**