一、基础知识填空

JS中的数据类型包含哪些(要求详细)以及基本数据类型和引用数据类型的本质区别：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

基本类型: number string boolean undefined null

引用类型: object function

基本数据类型操作的是值，引用数据类型操作的是内存空间的引用地址

JS中如何的检测数据类型(写出四种方法)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

typeof instanceof constructor Object.prototype.toString.call()

JS中判断某个值是否是一个有效数字的方法是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

isNan()

想要接收函数执行的时候传递的参数，都可以如何的处理：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

形参 arguments

将其他的数据类型转化为number类型的方式有那些：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Number() parseInt() parseFloat()

二、写出DOM中常用的方法和属性

有一个div:<div class=”w”name=”h”id=”div1”></div>我想获取这一个div你有几种解决办法(不考虑兼容)：\_\_\_\_\_\_\_\_\_\_

getElementById(“div1”);

getElementsByName(“h”)[0];

getElementsByClassName(“w”)[0];

getElementsByTagName(“div”)[0];

获取当前浏览器屏幕的宽度和高度(兼容所有的浏览器)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

document.documentElement.clientWidth || document.body.clientWidth;

document.documentElement.clientHeight || document.body.clientHeight;

获取上一个哥哥元素节点（兼容所有的浏览器）: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

function prev(ele){

var pre = ele.previousSibling;

while(pre && pre.nodeType !== 1){

pre = pre.previousSibling;

}

return pre;

}

动态创建一个div标签，并且添加到body的最后面位置：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

var oDiv = document.createElement(“div”);

document.body.appendChild(oDiv);

继续上一题，把刚才创建的克隆一份一模一样的，添加到刚才创建的div前面：\_\_\_\_\_\_\_\_\_\_\_\_\_\_

var oDiv2 = oDiv.cloneNode(true);

document.insertBefore(oDiv2,oDiv);

三、数组的基础知识和应用

我想删除数组中的最后一项，至少写出两种办法：\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

splice(ary.length-1,1) ; pop(); ary.length--;

我想在数组的末尾增加一个元素，至少写出两种办法：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

splice(ary.length,0,x); push(); ary[ary.length] = x;

基于原来的数组复制一份一模一样新的数组，不考虑原有数组是否改变的情况，请写出至少三种方案： \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

slice(0); slice(); concat(); splice(0);

数组从小到大排序(可以处理10以上的数字)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sort(function(a,b){return a-b})

实现找到第n项到第m项的内容，返回一个新的数组(原有数组不变)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_

slice(n-1,m);

四、写出下面表达式的运算结果

[]==false：\_\_\_\_\_true\_\_\_\_\_\_\_ ![]==false：\_\_\_\_\_true\_\_\_\_\_\_\_ []==![]：\_\_\_\_\_\_true\_\_\_\_\_\_\_\_ Number(“13px”)==NaN：\_\_\_\_\_false\_\_\_\_\_\_ 5+”3”-2：\_\_\_\_\_\_\_\_\_51\_\_\_\_\_\_\_\_\_\_

If(“3px”\*3){alert(“true”)}else{alert(“false”)}：\_\_\_\_\_\_”false”\_\_\_\_\_\_

If(“3px”+3){alert(“true”)}else{alert(“false”)}：\_\_\_\_\_\_”true”\_\_\_\_\_

If（document.body）{alert(“ok”);}else{alert(“no”)}：\_\_\_\_\_\_\_”ok”\_\_\_\_\_\_\_\_

alert([])：\_\_\_\_\_\_””（空字符串）\_\_\_\_\_\_\_\_ null==undefined：\_\_\_\_\_\_\_true\_\_\_\_\_\_\_\_

1. <input type="button" value="按钮1"/><input type="button" value="按钮2"/><input type="button" value="按钮3"/><input type="button" value="按钮4"/><input type="button" value="按钮5"/>

var ary=[“red”,”yellow”,”pink”,”green”,”blue”];

以上是页面中的5个按钮，和一个存储五个颜色值的数组，要求实现一个方法bindEvent，循环给5个按钮绑定点击事件，当点击的时候，整个body的背景颜色正好可以和数组中的指定项颜色对应(例如：点击第三个按钮，body的背景颜色”pink”)，请按照你的思路来实现?

方法一：

**var *ary*** = [**"red"**, **"yellow"**, **"pink"**, **"green"**, **"blue"**];  
**var *oBtn*** = **document**.getElementsByTagName(**"input"**);  
**for** (**var *i*** = 0; ***i*** < ***oBtn***.**length**; ***i***++) {  
 ***oBtn***[***i***].**index** = ***i***;  
 ***oBtn***[***i***].onclick = **function** () {  
 **document**.**body**.**style**.**backgroundColor** = ***ary***[**this**.**index**];  
 }  
}

方法二：

**var *ary*** = [**"red"**, **"yellow"**, **"pink"**, **"green"**, **"blue"**];  
**var *oBtn*** = **document**.getElementsByTagName(**"input"**);  
**for** (**var *i*** = 0; ***i*** < ***oBtn***.**length**; ***i***++) {  
 ***oBtn***[***i***].**onclick** = (**function** (i) {  
 **return function**(){  
 **document**.**body**.**style**.**backgroundColor** = ***ary***[i];  
 }  
 })(***i***);  
}

六、画图分析

var ary=[1,2,3,4];

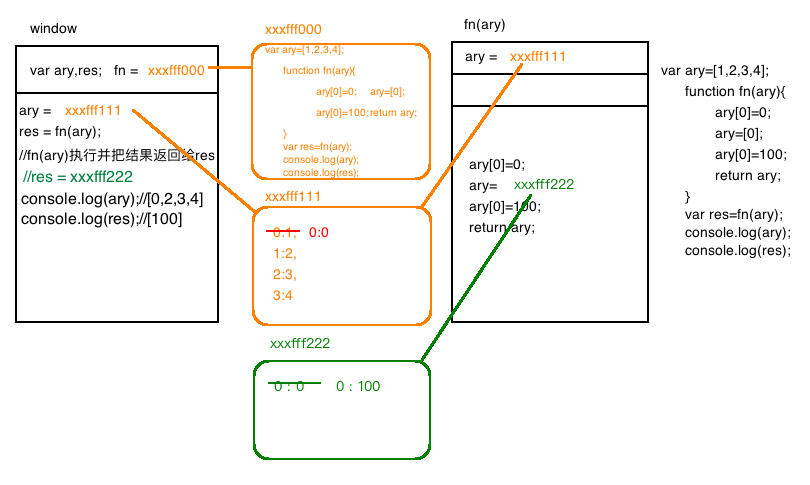
function fn(ary){

ary[0]=0; ary=[0]; ary[0]=100; return ary;

}

var res=fn(ary); console.log(ary); console.log(res); //此题偏难，认真思考(360面试题难度降低版本代码)

[0,2,3,4] [100]



### 二、选择题：（要求写出分析步骤/画图）

**1、** console.log(a);

var a=12;

function fn(){

console.log(a);

var a=13;

}

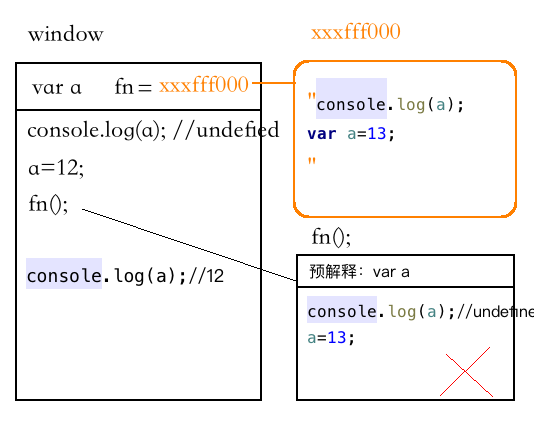
fn();

console.log(a);

输出的三次分别是多少

A、undefined 12 13 B、undefined undefined 12

C、undefined undefined 13 D、有程序报错



1. console.log(a);

var a=12;

function fn(){

console.log(a);

a=13;

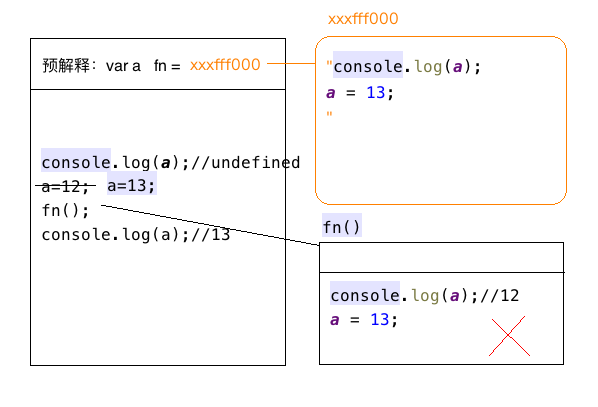
}

fn();

console.log(a);

A、undefined 12 13 B、undefined undefined 12

C、undefined undefined 13 D、有程序报错



1. console.log(a);

a=12;

function fn(){

console.log(a);

a=13;

}

fn();

console.log(a); 输出的三次分别是多少

A、undefined 12 13 B、undefined undefined 12

C、undefined undefined 13 D、有程序报错

预解释时，变量a没有提前用var关键字进行声明，执行到第一行console.log(a);时会出现程序报错

后面的代码不再继续执行

1. var foo=1;   
    function bar(){

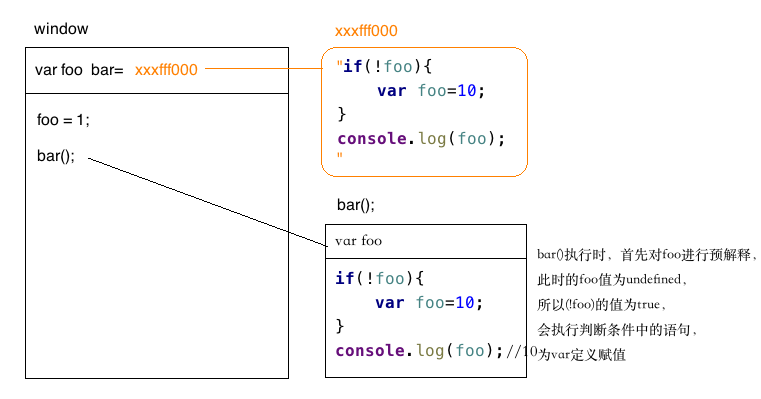
if(!foo){

var foo=10;

}  
 console.log(foo);

}  
 bar(); 输出的结果是多少

A、1 B、10 C、undefined D、报错



1. var n=0;

function a(){

var n=10;

function b(){

n++;

alert(n);

}

b();

return b;

}

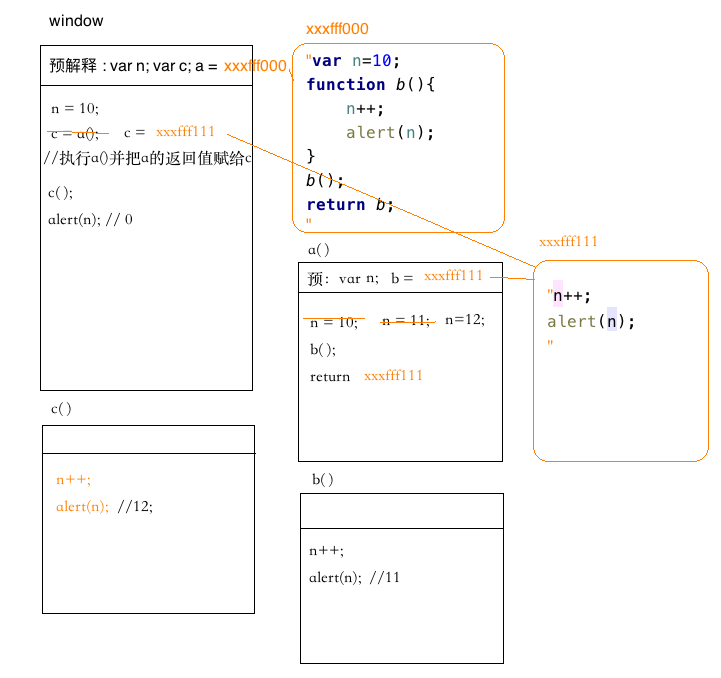
var c=a();

c();

alert(n);

弹出三次的结果分别是什么？

A、1 1 1 B、11 11 0 C、11 12 0 D、11 12 12



1. var a=10,b=11,c=12;

function test(a){

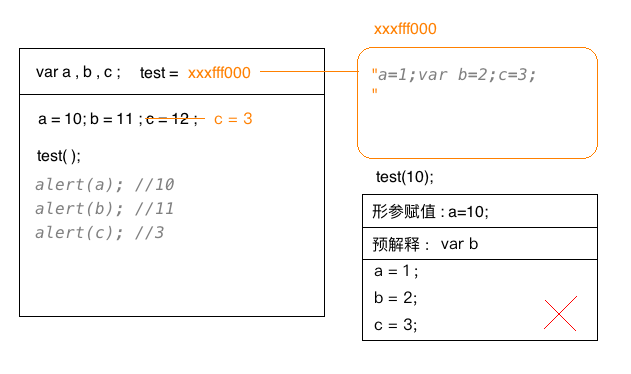
a=1;var b=2;c=3;

}

test(10);

alert(a); alert(b); alert(c); 弹出的三个值分别是多少？

A、1 11 3 B、10 11 12 C、1 2 3 D、10 11 3



7、if(!("a" in window)){

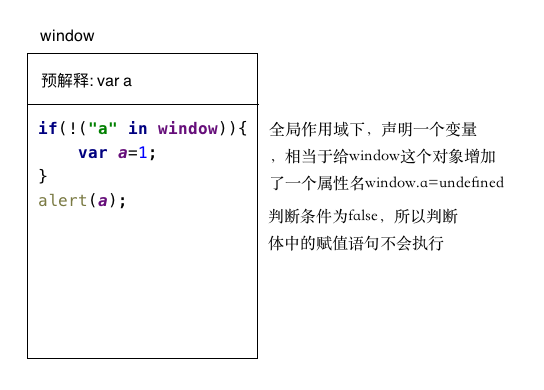
var a=1;

}

alert(a);

弹出的a的值是？

A、1 B、undefined C、报错 D、以上答案都不对



8、var a=4;

function b(x,y,a) {

alert(a);

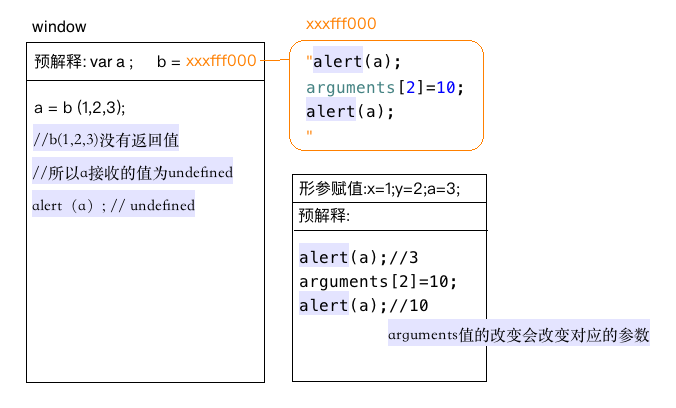
arguments[2]=10;

alert(a);

}

a=b(1,2,3); alert(a); 三次弹出的结果是

A、3 3 4 B、3 10 4 C、3 10 10 D、3 10 undefined



9、var foo='hello';

(function(foo){

console.log(foo);

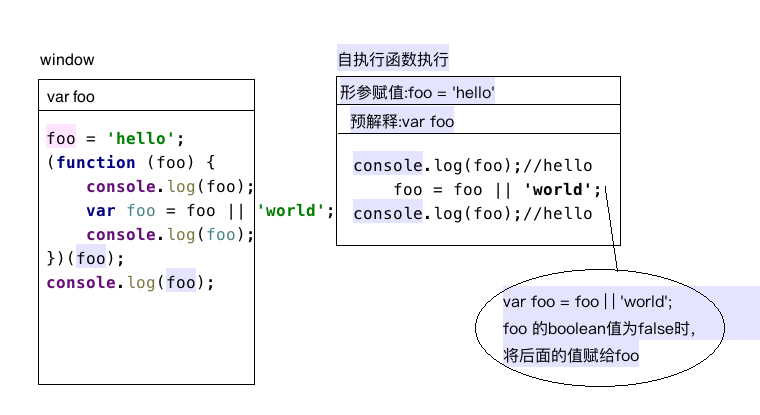
var foo=foo||'world';

console.log(foo);

})(foo);

console.log(foo); 三次分别输出什么？

A、hello hello hello B、undefined world hello C、hello world world D、以上答案都不正确



10、var a=9;

function fn(){

a=0;

return function(b){ return b+a++; }

}

var f=fn()

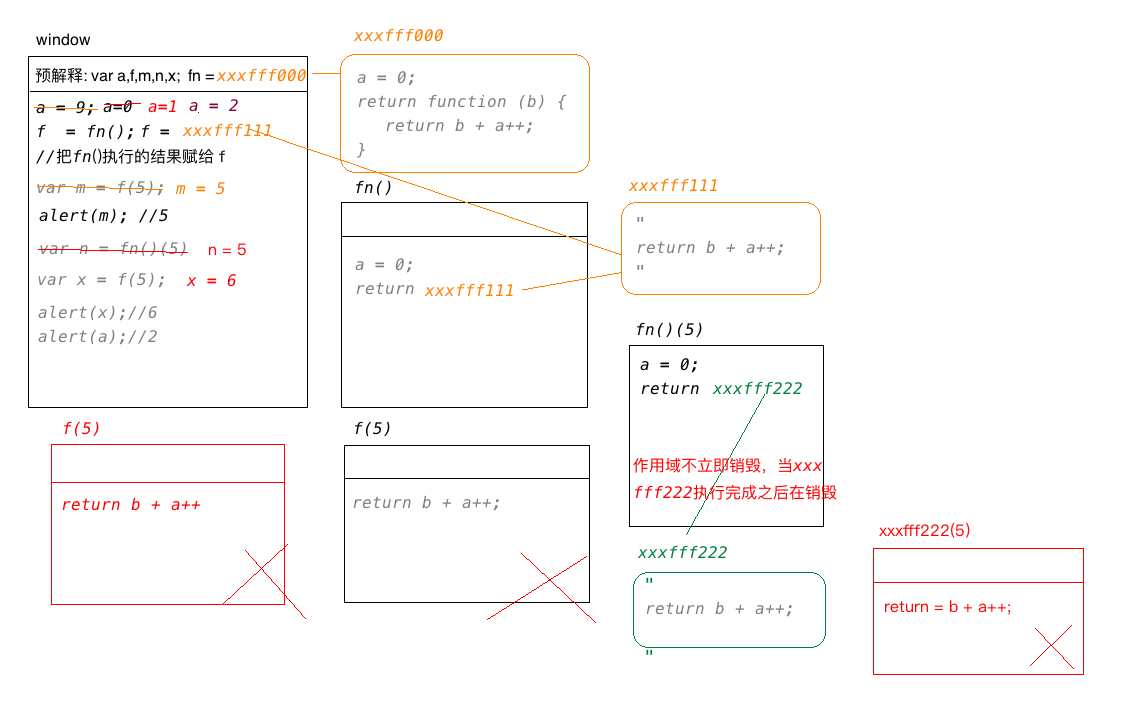
var m=f(5);alert(m);

var n=fn()(5);alert(n);

var x=f(5);alert(x);

alert(a); 弹出的四次结果？

A、6 6 7 2 B、5 6 7 3 C、5 5 6 2 D、以上答案都不正确



### 三、思考题：

1. document.parentNode的值是什么，为什么

返回null，parentNode方法会返回元素的父节点，document没有父节点，其它情况下会返回一个元素节点，因为只有元素节点才拥有子节点

1. document.parentnode的值是什么，为什么 。

undefined js严格区分大小写，所以parentnode是一个属性名，不是一个方法，获取未定义过的属性名会返回undefined

1. prototype是谁的属性，它起什么作用？\_\_proto\_\_是谁的属性，它有什么作用？prototype和\_\_proto\_\_有什么联系？

每个函数都会默认自带一个prototype属性，它的值是对象数据类型，浏览器会默认为它开辟一个堆内存，这个堆内存中会有一个constructor属性，指向当前类本身（只有浏览器默认为类创建的堆内存对象才有constructor这个属性）

每个对象都有一个\_\_proto\_\_属性，这个属性指向当前对象所属类的原型（prototype）

1. hasOwnProperty和in都是用来判断对象的属性的，这两个是怎么用的，有什么区别？

hasOwnProperty和in都用来判断一个对象是否存在某个属性

用法：例如检测fn对象中是否存在name属性

fn.hasOwnProperty(“name”) 如果name是fn对象的私有属性返回true，不管公有还是没有，只要私有的不存在就返回false

“name”in fn 如果fn中拥有name属性，不管是公有的还是私有的都会返回true，不存在 则返回false

1. 什么是闭包，它有什么作用？

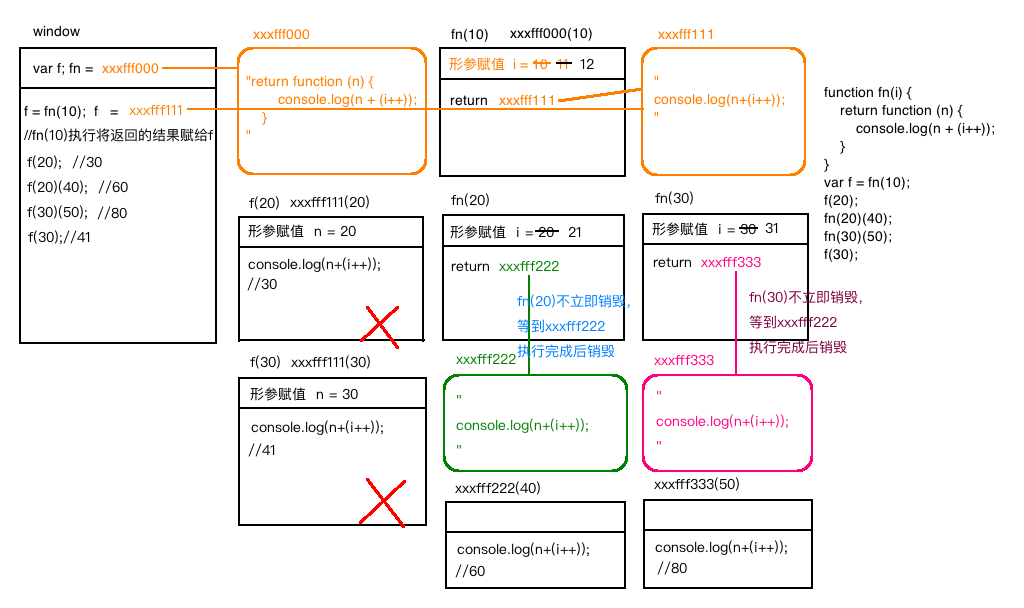
函数执行的时候会形成一个私有作用域，来保护里面的私有变量不受外界的影响，这种机制叫做闭包

1. **附加题** 画图/分析步骤

1、

function fn(i) {  
 return function (n) {  
 console.log(n + (i++));  
 }  
}  
var f = fn(10);  
f(20);  
fn(20)(40);  
fn(30)(50);  
f(30);

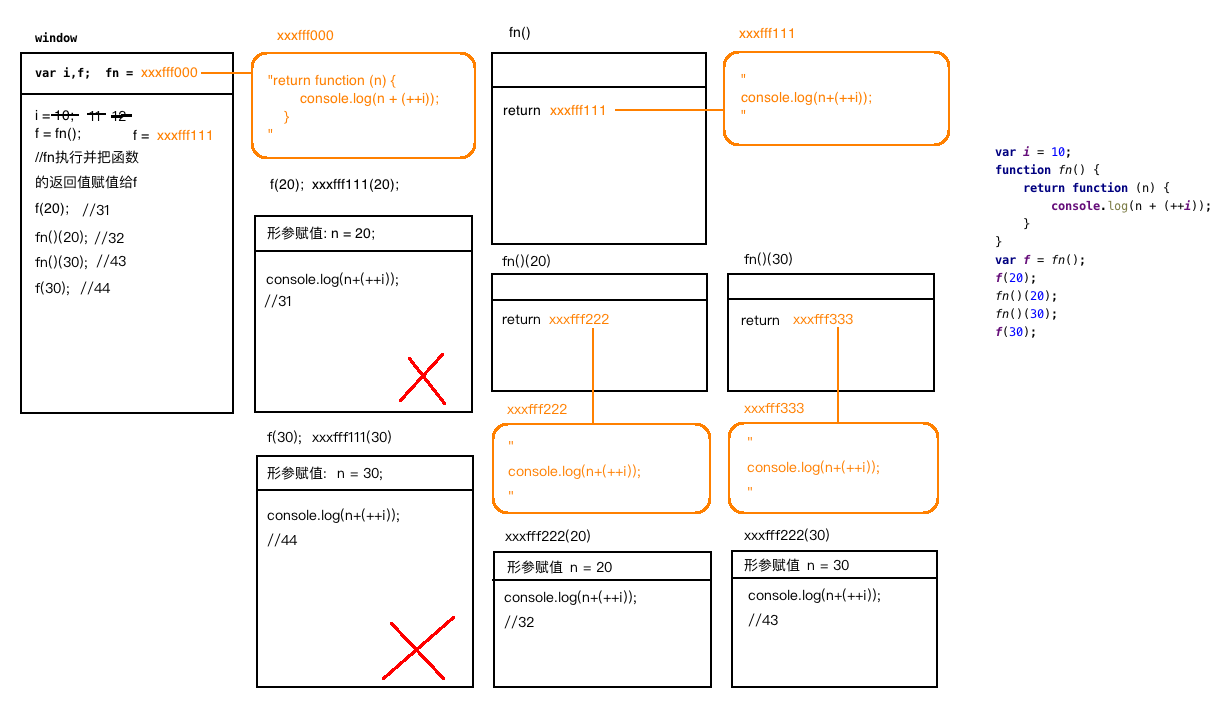
结果：30，60，80，41



2、

var i = 10;  
function fn() {  
 return function (n) {  
 console.log(n + (++i));  
 }  
}  
var f = fn();  
f(20);  
fn()(20);  
fn()(30);  
f(30);

结果：31，32，43，44

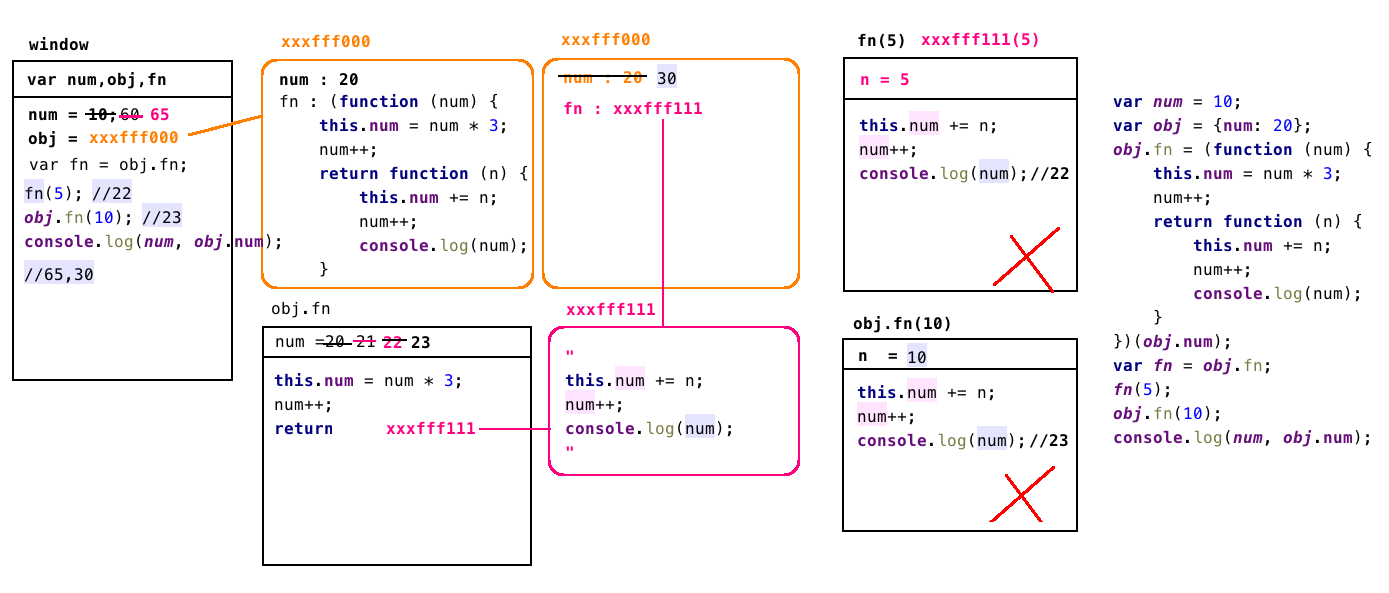


3、

var num = 10;  
var obj = {num: 20};  
obj.fn = (function (num) {  
 this.num = num \* 3;  
 num++;  
 return function (n) {  
 this.num += n;  
 num++;  
 console.log(num);  
 }  
})(obj.num);  
var fn = obj.fn;  
fn(5);  
obj.fn(10);  
console.log(num, obj.num);

结果：

22，23，65，30



4、

function Fn() {  
 this.x = 100;  
 this.y = 200;  
 this.getX = function () {  
 console.log(this.x);  
 }  
}  
Fn.prototype.getX = function () {  
 console.log(this.x);  
};  
Fn.prototype.getY = function () {  
 console.log(this.y);  
};  
var f1 = new Fn;  
var f2 = new Fn;

console.log(f1.getX === f2.getX); //false  
console.log(f1.getY === f2.getY); //true  
console.log(f1.\_\_proto\_\_.getY === Fn.prototype.getY); //true  
console.log(f1.\_\_proto\_\_.getX === f2.getX); //false  
console.log(f1.getX === Fn.prototype.getX); //false  
console.log(f1.constructor); //Fn  
console.log(Fn.prototype.\_\_proto\_\_.constructor); //Object  
f1.getX(); //100  
f1.\_\_proto\_\_.getX(); //undefined  
f2.getY(); //200  
Fn.prototype.getY(); //undefined

