1. 基础知识填空
2. 基本数据类型：数类型Number 字符串String 布尔类型Boolean null undefined

引用数据类型：对象 数组 正则 函数 object array regExp function

本质区别：更改基本数据类型的时候是直接更改它的值，更改引用数据类型的时候是更改引用空间里的信息

1. typeof instanceof Number() Array isArray

isNaN

1. 形参实参组合 arguments传参

Number() parseInt() parseFloat

1. 写出DOM中常用的方法和属性

var oDiv = document.getElementById(“div1”);

var oDiv = document.getElementsByTagName(“div”)[0];

var oDiv = document.getElementsByName(“h”)[0];

var oDiv = document.getElementsByClassName(“w”)[0];

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

var oDiv = document.querySelectorAll(“div”)[0];

2、

var oHeight = window.innerHeight;

var oHeight = window.outerHeight;

var oWidth = window.innerWidth;

var oWidth = window.outerWidth;

previousElementSibling

var pre = oDiv.previousSibling;

var ary = [];

while(ary.length < 1){

If( pre.nodeType == 1){

ary[ary.length] = pre;

}else{

pre = pre.previousSibling

}

}

console.log(ary);

3、

var oDiv = document.createElement("div");  
document.body.appendChild(oDiv);  
oDiv.style.height = "80px";  
oDiv.style.width = "80px";  
oDiv.style.background = "pink";  
console.log(oDiv);  
var cloneDiv = oDiv.cloneNode(true);  
cloneDiv.style.height = "50px";  
cloneDiv.style.width = "50px";  
cloneDiv.style.background = "orange";  
document.body.insertBefore(cloneDiv,oDiv);

1. 数组的基础知识和应用

arr.pop() arr.length-- arr.splice(arr.length-1,1) arr.splice(arr.length-1)

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

2、

arr.splice(0) arr.slice(0) [].concat(ary) ary.join()

3、

ary.sort(function(a,b){return a-b;})

arr.slice(n-1,m)

四、写出下面表达式的运算结果

1、 [] == false -> 0 == 0 -> **true**

2、![] == false -> !ture ==false -> false==false->0==0 -> **true**

3、 [] == ![] ->0 == false -> 0 == 0 -> **true**

**4、** Number(“13px”) == NaN -> **false** NaN连自己都不等

5、 5+”3”-2 -> “53”+2-> **”532”**

6、 “3px”\*3 -> NaN -> **false**

7、 “3px”+3 -> “3px3” -> **true**

8、 document.body -> true -> alert(“ok”)

9、 alert([]) -> alert(“ ”) -> undefined

10、 null == undefined -> ture

五、

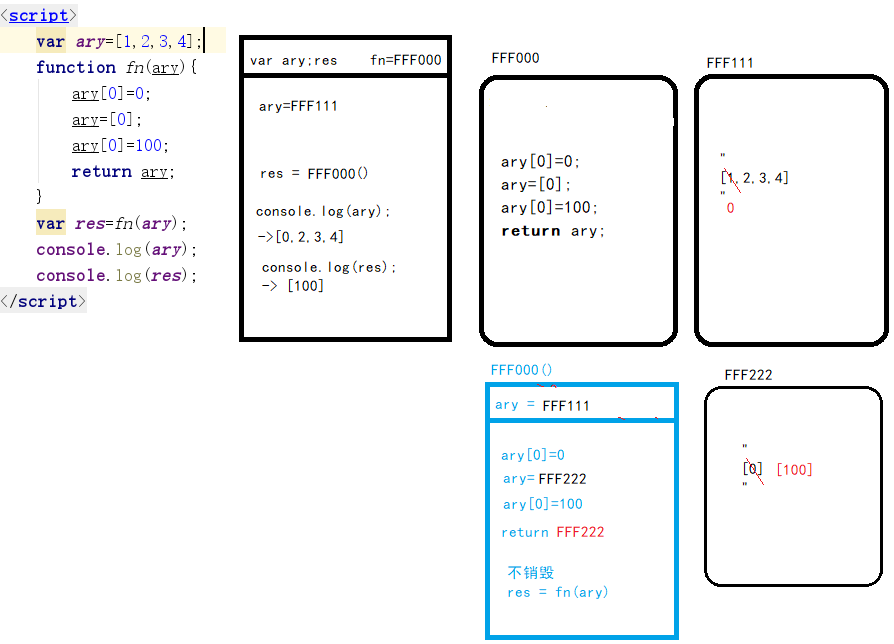
//方案一

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

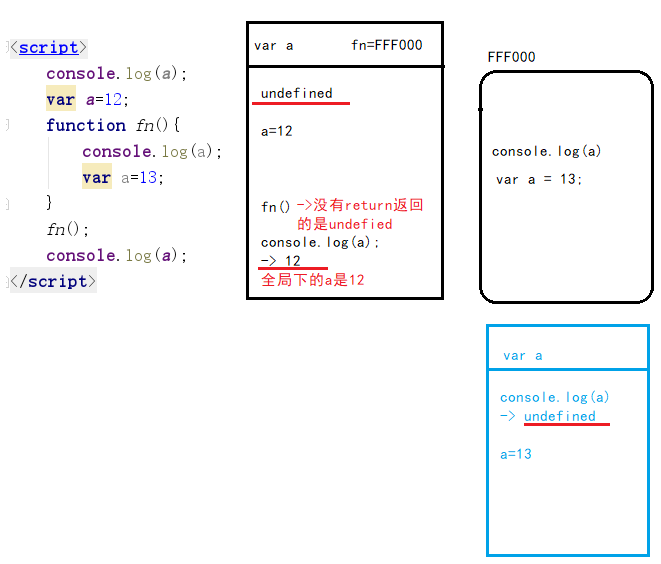
//方案二

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

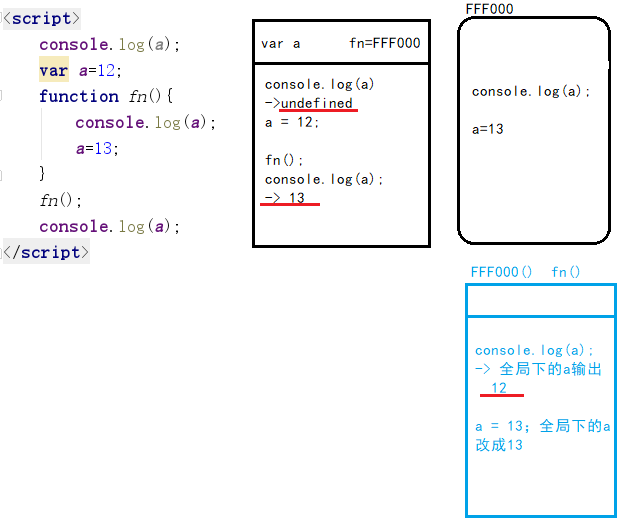
1. 画图分析



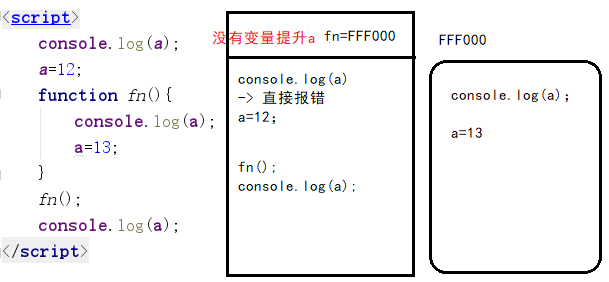
1. 选择题（要求写出分析步骤/画图）
2. 选择B

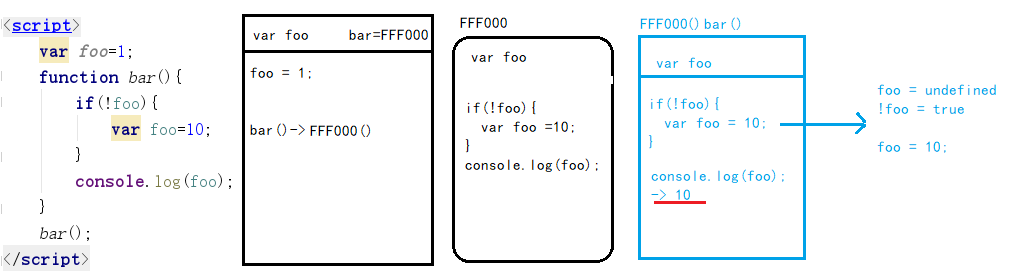


2、选择 A

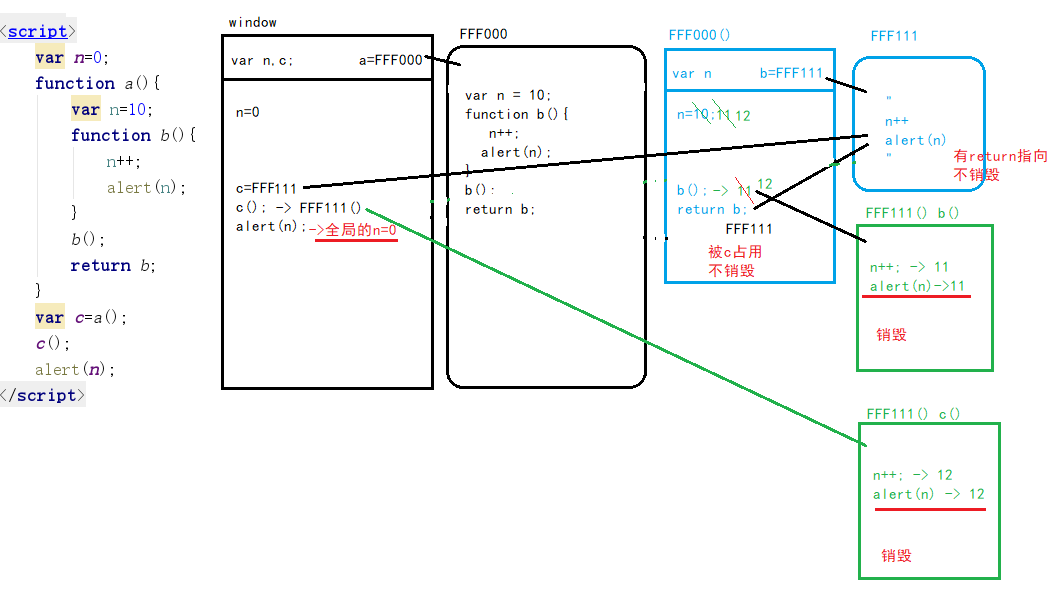


3、选择D

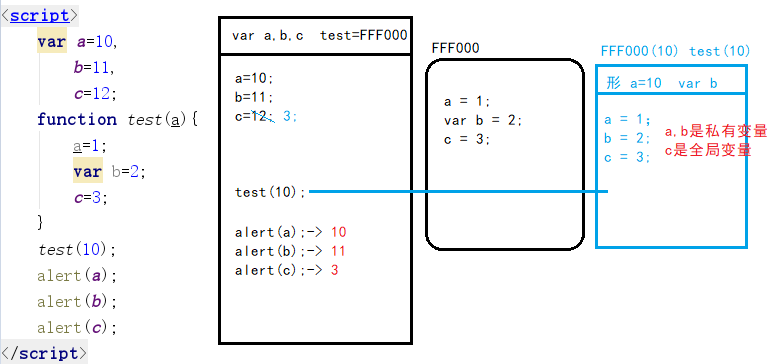
4、选择B

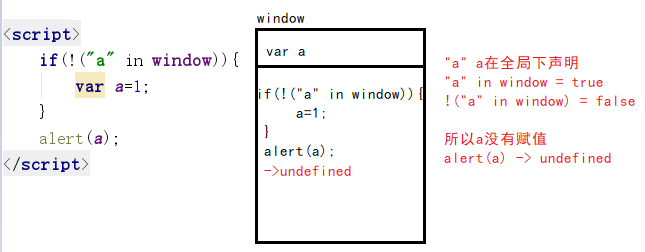


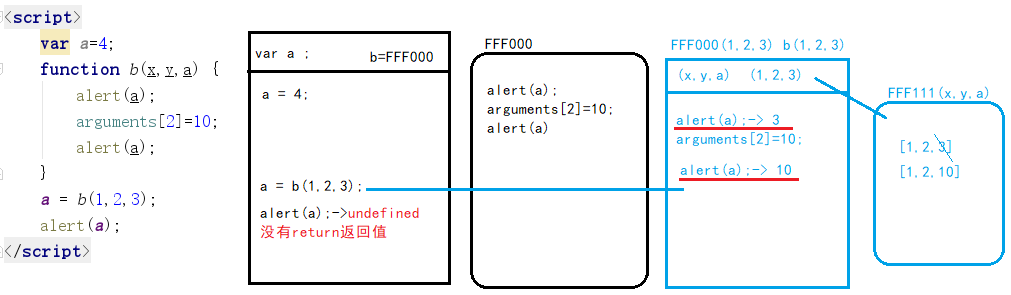
5、选择C

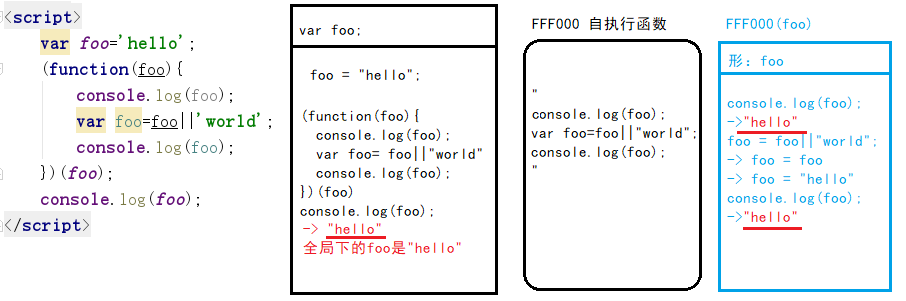


1. 选D

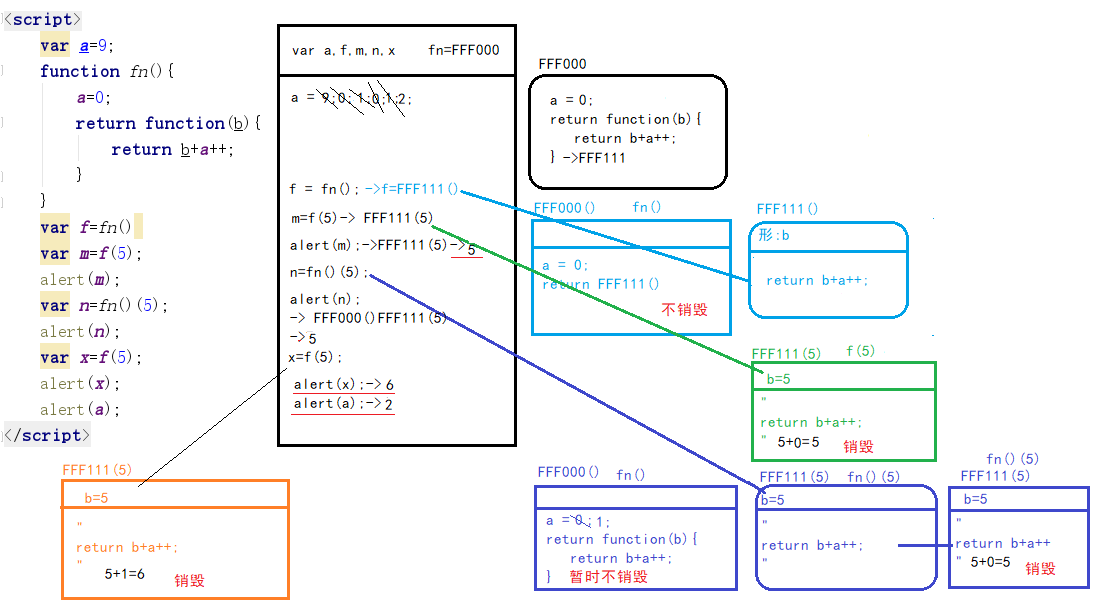


1. 选择 B
2. 选择D



1. 选择A

10、选择D 5 5 6 2



1. 思考题
2. document.parentNode的值是什么，为什么

值是null，document已经是根节点了（祖宗了），再往上就没了

1. document.parentnode的值是什么，为什么

值是undefined，document下没有这个属性，所以是undefined，document.amin也是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？

（1）每一个函数（类），都有一个prototype属性，存储的是给当前类使用的公有属性和方法，prototype是一个对象，浏览器默认给他开辟一个堆内存

（2）每一个prototype都有一个constructor指向类本身

（3）\_\_proto\_\_：每一个实例（对象），都有一个\_\_proto\_\_属性，指向所属类的原型（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 检测属性是不是对象的私有属性，如果是返回true，否则返回false

in是判断对象是否拥有这个属性

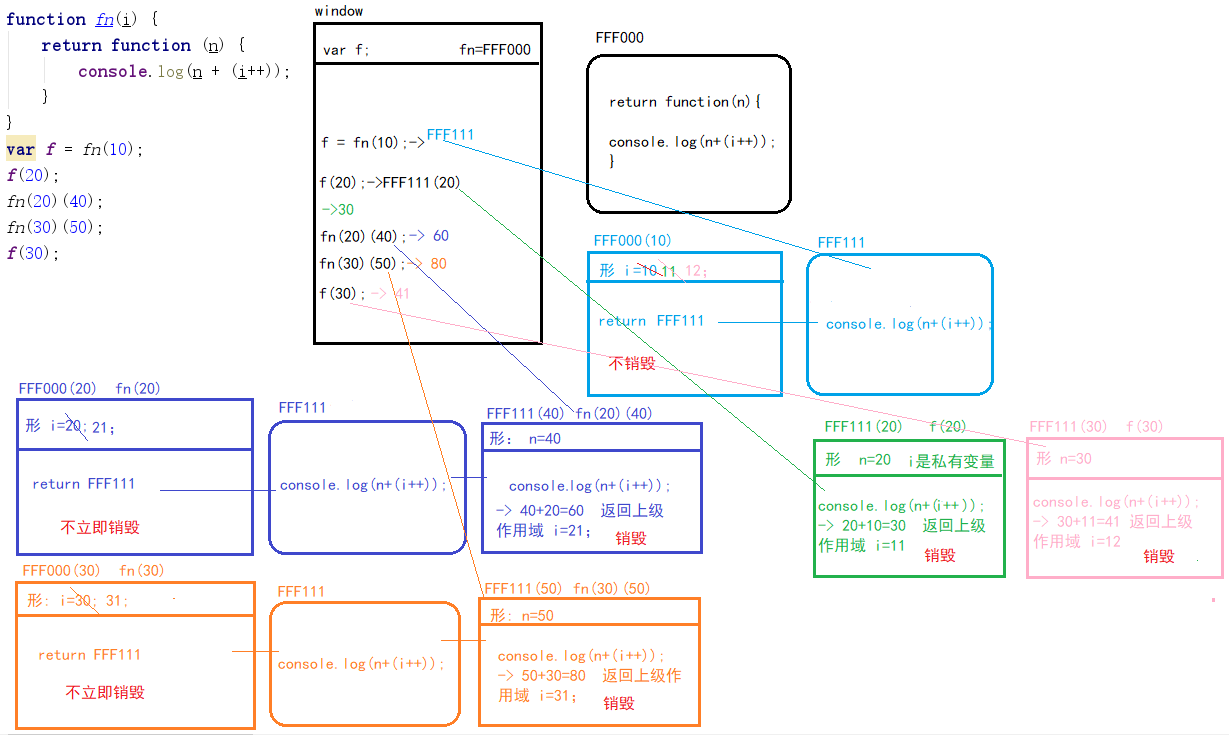
1. 什么是闭包，它有什么作用？

方法在执行的时候产生一个私有作用域，可以保护里面的私有变量不受外界干扰，这种保护机制就叫闭包。

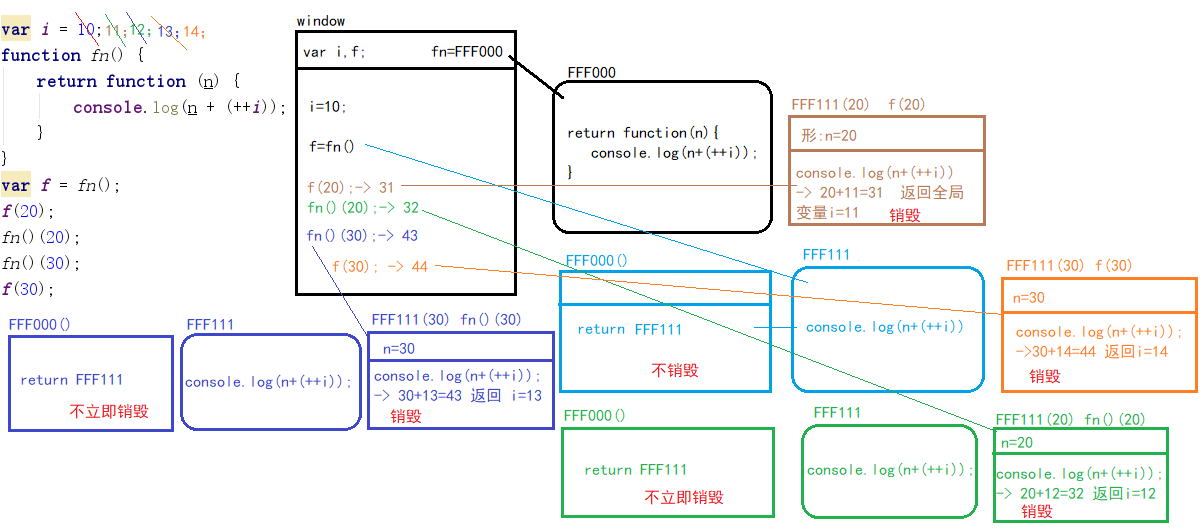
模块化。

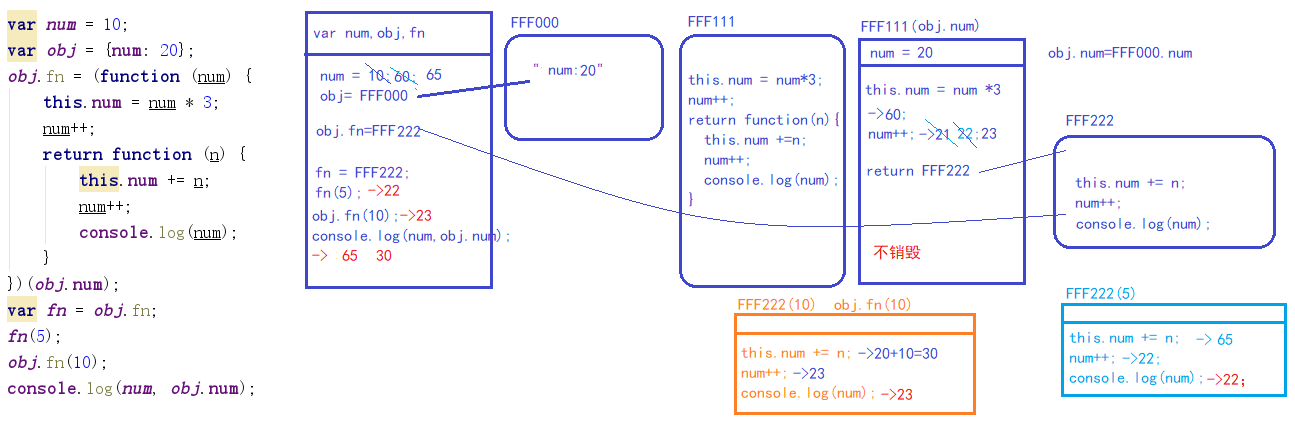
1. 附加题 画图/分析步骤

1、按顺序输出的是 30 60 80 41



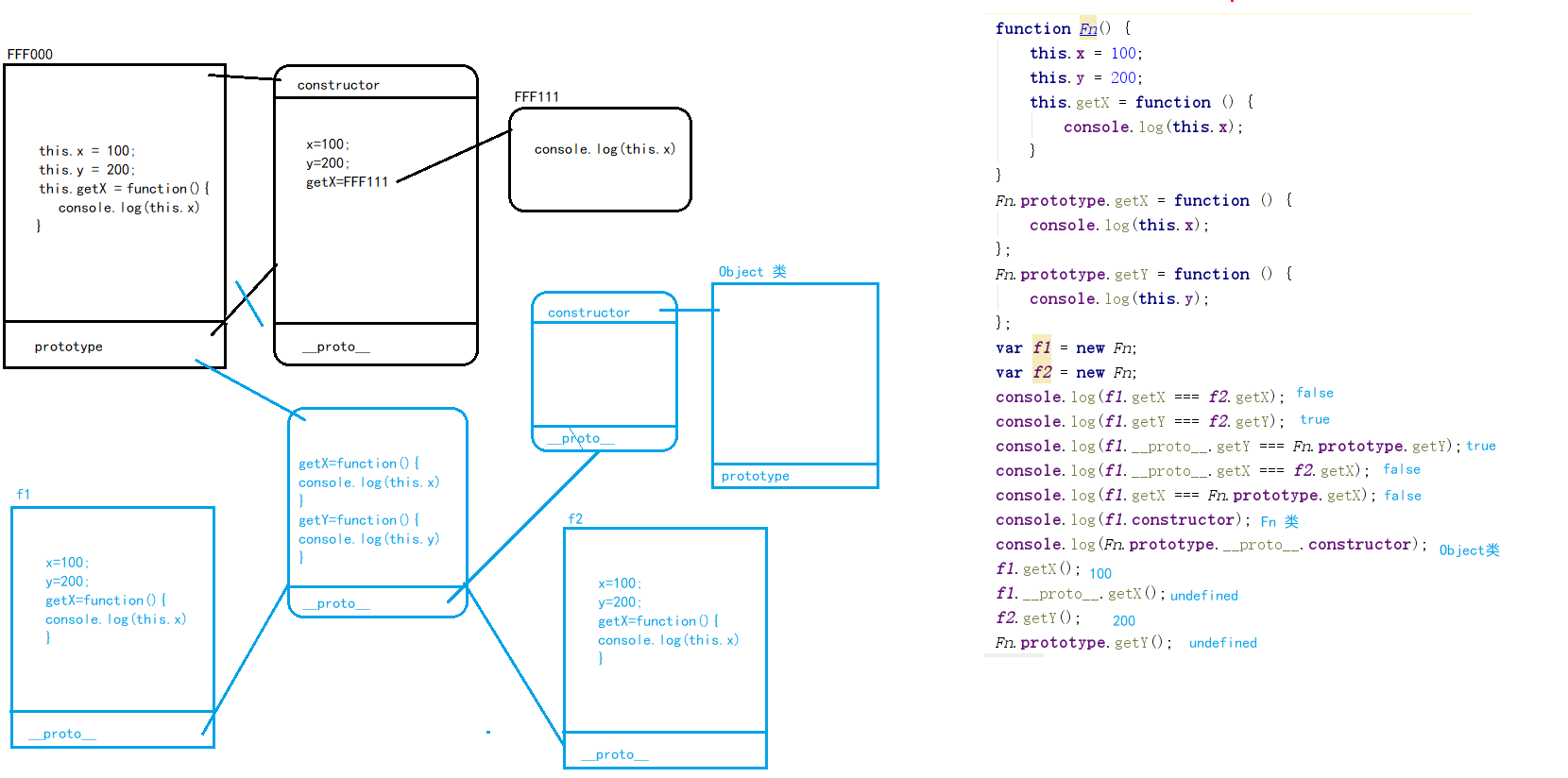
2、按顺序输出 31 32 43 44

3、按顺序输出 22 23 65 30



1. false true true false false

Fn类 Object类 100 undefined 200 undefined



5、不能实现，下面的tabChange（i）传递的i是for循环里遍历完之后的i = oLis.length-1

可以使用自定义属性将每个i保存下来

(1)

for（var i=0; i <oLis,length; i++）{

oLis[i].zhufeng = i;

oLis[i].onclick = function(){

tabChange(this.zhufeng);

}

}

(2)

**for** (**var *i*** = 0; ***i*** < ***oLis***.**length**; ***i***++) {  
 (**function** change(n){  
 ***oLis***[n].onclick = **function**(){  
 **for** (**var** i = 0; i < ***oLis***.**length**; i++){  
 ***oLis***[i].**className** = **""**;  
 ***oDiv***[i].**className** = **""**;  
 }  
 ***oLis***[n].**className** = **"selectedLi"**;  
 ***oDiv***[n].**className** = **"selectedDiv"**;  
 }  
 })(***i***)  
}

(3)

**for** (**let *i*** = 0; ***i*** < ***oLis***.**length**; ***i***++) {  
 ***oLis***[***i***].onclick = **function**(){  
 **for** (**var** j = 0; j < ***oLis***.**length**; j++){  
 ***oLis***[j].**className** = **""**;  
 ***oDiv***[j].**className** = **""**;  
 }  
 ***oLis***[***i***].**className** = **"selectedLi"**;  
 ***oDiv***[***i***].**className** = **"selectedDiv"**;  
 }  
}