前端面试（20分针完成）

一、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：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JS中如何的检测数据类型(写出四种方法)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ JS中判断某个值是否是一个有效数字的方法是：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

想要接收函数执行的时候传递的参数，都可以如何的处理：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 将其他的数据类型转化为number类型的方式有那些：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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)：\_\_\_\_\_\_\_\_\_\_

获取当前浏览器屏幕的宽度和高度(兼容所有的浏览器)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 获取上一个哥哥元素节点（兼容所有的浏览器）: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

三、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：\_arr.pop()\_\_\_\_\_\_ \_\_\_\_\_arr.splice(arr.length-1,1)\_\_\_\_\_\_\_\_\_ 我想在数组的末尾增加一个元素，至少写出两种办法：\_\_\_\_arr.push(val)\_\_\_\_\_\_arr.splice(arr.length,val)\_\_\_\_\_\_\_\_

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： \_\_arr.slice()/arr.concat()/\_\_\_\_\_\_\_\_\_\_\_\_\_

var arr = [1,2,3,4,5]

var arr2 = copyArr(arr)

function copyArr(arr) {

let res = []

for (let i = 0; i < arr.length; i++) {

res.push(arr[i])

}

return res

}

数组从小到大排序(可以处理10以上的数字)：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 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)：\_\_\_arr.slice(n-1,m-n+2)\_\_\_\_\_\_\_\_\_\_\_

//冒泡

var arr=[1, 0,-3, 9, 10, 33, 5, 21, 3],len=arr.length-1

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

for(var j=0;j<len-i;j++){

if(arr[j]>arr[j+1]){

var temp = arr[j+1]

arr[j+1] = arr[j]

arr[j] = temp

}

}

}

四、写出下面表达式的运算结果

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

![]为false， Number('13px')结果为NaN，但NaN不等于NaN

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

If（document.body）{alert(“ok”);}else{alert(“no”)}：\_\_\_\_\_alert(“ok”)\_\_\_\_\_\_\_\_\_\_ alert([])：\_\_\_\_\_空\_\_\_\_\_\_\_\_\_ alert({})：\_\_\_\_\_[Object Object]\_\_\_\_\_\_\_\_\_ 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?

void function bindEvent(){

var aBtn = document.getElementsByTagName("input")

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

for(let i=0;i<aBtn.length;i++){

//方法一：提前设置属性变量

// aBtn[i].index=i;

// aBtn[i].onclick=function(){

// document.body.style.background=ary[this.index]

// }

//方法二：闭包

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

return function(){

document.body.style.background=ary[i]

}

})(i)

}

}()

六、画图分析

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

1. 声明  
    var ary=undefined;  
    function fn(ary){...}  
    var res=undefined;
2. 赋值  
    ary=[1,2,3,4]  
    res=function fn(ary){...}返回的ary  
    即res=[100]
3. 打印  
    [1,2,3,4]  
    [100]

function fn(ary){

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

}

var res=fn(ary); console.log(ary); console.log(res);

### 二、选择题：

**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

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=undefined

"a" in window-->true

!("a" in window)-->false

alert(a); a--->undefined

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 3 D、以上答案都不正确 5562

### 三、思考题：

1. document.parentNode的值是什么，为什么 null，document没有父节点
2. document.parentnode的值是什么，为什么 。 undefined，document没有parentnode属性
3. 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？

prototype是函数数据类型的属性，作用：这个属性是一个对象数据类型，存放公有属性和方法。

\_\_proto\_\_是对象数据类型的属性，作用：原型链，值：当前实例所属类的原型。

prototype和\_\_proto\_\_联系：当前实例通过\_\_proto\_\_找到prototype这个原型。

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？  
in判断的是对象的所有属性，包括对象实例及其原型的属性；

而hasOwnProperty则是判断对象实例的是否具有某个属性。

1. 什么是闭包，它有什么作用？

当函数执行的时候，把原有存储的代码字符串拿出来，变成真正的js代码，然后从上到下执行，并形成一个新的代码运行环境（作用域 ），在自己的作用域中定义的变量不受外界的干扰（自己家里的变量和函数外面的没有关系），我们把这种机制叫做闭包。

作用是让作用域中的变量不受外界影响

5、你理解的DOM是用来干什么的？（DOM的意义和用途）

DOM是文档对象模型 document object model

DOM中提供了很多获取元素的方法、获取元素之间关系的属性，获取元素本身本身特性的方法（DOM盒子模型）、以及对这些元素的相关操作。

1. **附加题**

1、

function fn(i) {  
 return function (n) {  
  **console.log(n + (i++));**  
 }  
}  
var f = fn(10);  
f(20);//30  
fn(20)(40);//60  
fn(30)(50);//80  
f(30);//41

f=function(n){  
 console.log(n+(i++))  
}

先打印n+i

后i++

2、

var i = 10;  
function fn() {  
 return function (n) {  
 console.log(n + (++i));  
 }  
}  
var f = fn();  
f(20);//31  
fn()(20);//32  
fn()(30);//43  
f(30);//44

3、

var num = 10;  
var obj = {num: 20};  
obj.fn = (function (num) {  
 this.num = num \* 3;//window.num=60  
 num++;//obj.fn函数局部变量num=21  
 return function (n) {  
 this.num += n;  
 num++;  
 console.log(num);  
 }  
})(obj.num);  
var fn = obj.fn;  
fn(5);//22  
obj.fn(10);//23

fn(5)

--->window.num+= 5-->65

--->obj.fn函数局部变量num++ -->22

obj.fn(10)

--> obj.num += 10--->30

--->obj.fn函数局部变量num++ -->23

obj.fn = function (n) {

this.num += n;

num++;

console.log(num);

}

console.log(num, obj.num);//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);   
console.log(Fn.prototype.\_\_proto\_\_.constructor);//function Object() { [native code] }  
f1.getX();//100  
f1.\_\_proto\_\_.getX();//undefined

f1.constructor=function Fn() {

this.x = 100;

this.y = 200;

this.getX = function () {

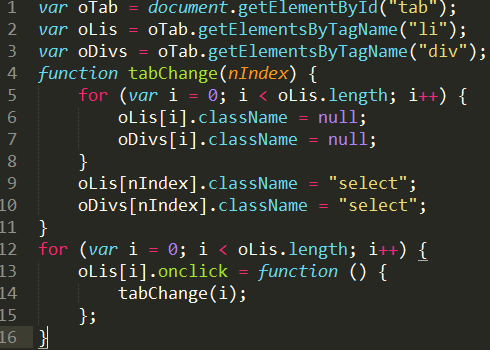
console.log(this.x);

}

}

f2.getY();//200  
Fn.prototype.getY();//undefined

1. 我想通过下面的代码实现选项卡，请问是否可以实现？不能的话，说明原因，那么你有几种办法可以解决?



修改：

方法2：

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

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

return function(){

tabChange(this.index)

}

})(i)

}

方法1：

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

oLis[i].index=i;

oLis[i].onclick = function(){

tabChange(this.index)

}

}