### 介绍JavaScript的基本数据类型?

回答:字符串 String;数字 Number;布尔Boolean;undefined,null,object六种基本数据类型

### JavaScript原型,原型链?有什么特点?

回答:js中创建的对象都会在其内部初始化一个属性，就是prototype，

这prototype就是称之为原型，而属性prototype里面又有自己的prototype

属性，然后里面又可以找到自己的prototype属性，一直找下去的这种链条关系

就称之为原型链呢。

特点：js对象是通过引用来传递的，当我们需要一个对象属性时，程序会先从对

象本身去找，如果没有的话，会再去找它的prototype属性，如果prototype

属性没有的话，会再去找它的原型链，一直找下去，最后找到null元素。

### null 和 undefined 有何区别？

undefined表示变量声明但没赋值的值，

null表示准备用来保存对象，还没有真正保存对象的值，还可以释放内存，等

待程序中的垃圾回收站回收多余的变量。

### 4.观察下面代码,写出输出结果.

**var box = function(){**

**var num = 10;**

**--num;**

**return function(){**

**console.log(num++);**

**}**

**}();**

**box();**

**box();**

**box();**

**输出顺序为:9,10,11.**

### 观察下面代码,写出输出结果.

**function box(a,b){**

**var a=b=c=0;**

**}**

**box(10,100);**

**console.log(a);**

**console.log(b);**

**输出顺序为undefined，undefined**

### 数组对象有哪些原生方法，列举一下？

回答：第一种是Array.pop();// 删除数组最后一个元素，返回被删除的元素

第二种是Array.push();// 在数组尾部插入1-N个元素，返回操作后数组的length

第三种是Array.shift(); // 删除数组第一个元素，返回被删除的元素

第四种是Array.unshift() ;// 在数组头部插入1-N个元素，返回操作后数组的length

第五种是reverse（）;//颠倒数组中元素的顺序，并返回逆序后的数组，

第六种是splice（）;//是针对pop、push、shift、unshift 的一个补充。

返回值是被删除的元素。

1. **编写一个方法 去掉一个数组的重复元素**

**回答：**

1. **截取字符串abcdefg的efg**

**回答：var reg=”abcdefg”;**

**document.write(reg.substring(4));**

**9.以下关于Array数组对象的说法不正确的是（D）  
A. 对数组里数据的排序可以用sort函数，如果排序效果非预期，可以给sort函数加一个排序函数的参数  
B. reverse用于对数组数据的倒序排列  
C. 向数组的最后位置加一个新元素，可以用pop方法  
D. unshift方法用于向数组删除第一个元素**

**10.’= =’和’= = =’的区别**

**回答：’= =’是非严格判断，’= = =’是严格判断。**

**11.以下哪条语句会产生运行错误：（A）   
A.var obj = ();  
B.var obj = [];  
C.var obj = {};  
D.var obj = //;**

**12.请选择结果为真的表达式：（B）   
A.null instanceof Object（if(!(null instanceof Object))是真的）   
B.null === undefined   
C.null == undefined   
D.NaN == NaN**

**13.求y和z的值是多少？  
var x = 1;  
var y = 0;  
var z = 0;  
function add(n){n=n+1;}  
y = add(x);  
function add(n){n=n+3;}  
z = add(x);**

**回答：y值是undefined，z值是undefined。**

1. **写出程序运行结果**

**console.log(box);**

**function box(){}**

**var box = 100;**

**console.log(box);**

**var box = function(a){**

**return a++;**

**}**

**console.log(box);**

**console.log(box(100));**

**输出顺序为:第一个box值为 function box(){}**

**第二个box值为 100**

**第三个box值为 var box = function(a){return a++;}**

**第四个box值为 100**

1. **var box = {}与 var box = new Object()这两种写法的区别.**

**Var box={}字面量创建法，里面写对象属性时要用逗号给开，**

**而var box = new Object()是通过new关键字创建的。**

1. **观察下面代码运行结果.**

**var str = new String(‘aaa’);**

**var str1 = ‘aaa’;**

**console.log(str .substring(0));**

**console.log(str1 .substring(0,5));**

**str.name = ‘小黑’;**

**str1.name = ‘小白’;**

**console.log(str.name);**

**console.log(str1.name);**

**输出顺序为:回答：第一个是str .substring(0)值为aaa**

**第二个是str1 .substring(0,5)值为aaa**

**第三个是str.name值为 小黑**

**第四个是str1.name值为 undefined**

1. **观察下面代码运行结果.**

**function Foo(){}**

**Foo.prototype.X = 10;**

**var obj = new Foo();**

**console.log(obj.X);**

**obj.X = 100;**

**console.log(Foo.prototype.X);**

**console.log(obj.X);**

**输出顺序为:回答：第一个是obj.x值为10**

**第二个是Foo.prototype.x值为10**

**第三个obj.x值为100**

1. **观察下面代码运行结果.**

**function getColor(){**

**console.log(this.color);**

**}**

**var color = '红色';**

**var box = {**

**color:'蓝色'**

**}**

**console.log(getColor.call(this));**

**console.log(getColor.call(window));**

**console.log(getColor.call(box));**

**输出顺序为:**

**第一个是 console.log(this.color)值为 红色**

**第二个是 console.log(getColor.call(this))值为 蓝色**

**第三个是 console.log(getColor.call(window))值为 红色**

**第四个是 console.log(getColor.call(box))值为 蓝色**

1. **观察下面代码运行结果.**

**var num =10;**

**function box1(){**

**console.log(num);**

**function box2(){**

**console.log(arguments[0]);**

**function box3(){**

**var num = 10;**

**console.log(num);**

**}**

**box3();**

**}**

**box2(100);**

**}**

**box1();**

**输出顺序为:**

**第一个是console.log(num) 值为 10**

**第二个是 console.log(arguments[0])值为 100**

**第三个是 console.log(num)值为 10**

**胡树东Js个人总结：就是感觉对js程序的运行，还是不了解，有变量和函数的程序是，对他们的作用域还不是懂他们的运行，特别对函数中的return执行不是明白，return返回是显示一个值和函数时，我分不明白到底是返还一个值还是一个函数，而且返还的值或函数到底起什么作用，希望老师重点讲解一下！**

**严格模式预习小总结：严格模式是一种特殊的执行模式，它修复了 部分语言上的不足，提供更强的错误检查，并增强安全性。**

**严格模式强调的几点：1.**在正常模式中，如果一个变量没有声明就赋值， 默认是全局变量。严格模式禁止这种用法，全局变量必须显式声

明

1. 声明变量前，一定要加var，然后才能使用 **3.**使用构造函数时，如果忘了加new，this不再指向全局对象，而是报错。
2. 严格模式下无法删除变量。只有configurable设置为true的对象属性，才能被删除

5. 正常模式下，如果对象有多个重名属性，最后赋值的那个属性会覆盖前面的值。严格模式下，这属于语法错误。

6.正常模式下，如果函数有多个重名的参数，可以用arguments[i]读取。严格模式下，这属于语法错误。