**第十次周考题**

**姓名：**

**一：不定项选择：每题3分**

1. 一年有12个月，要求月份的正确格式为: 1,2,….9,10,11,12，以下哪个正则表达式可以符合要求？  (C)

A./^[1-12]$/  B./^[1-9]\d?$/

C./^([1-9]︱1[0-2])$/ D./^\d︱11︱12︱10$/

解析：一般情况下,正则中的一个字符对应字符串中的一个字符，| 竖线两边任意匹配

1. 声明一个对象，给它加上 name 属性和 show 方法,在show方法中打印这个对象的 name 值，以下代码中正确的是（D）

A. var obj = [name:"tom",show:function(){alert(name)}]

B. var obj = {name:”tom”,show:alert(this.name)}

C. var obj = {name:”tom”,show:function(){alert(name)}}

D. var obj = {name:”tom”,show:function(){alert(this.name)}}

解析：声明对象，可以用{}一对花括号表示这是一个对象，对象里的属性和值用’:’分开，不同属性之间用’,’隔开，在对象里，this表示对象本身

1. 观察以下代码：输出结果为（B）

A：undefined B：object C：string D：null

<script>

var a = null;

console.log(typeof a);

</script>

解析：null是一个只有一个值的数据类型，这个值就是null。表示一个空对象，空对象也是一个对象，所以用typeof检测会返回”object”

1. 分析以下代码，输出结果是（C）

<script>

var a;

console.log(a);

console.log(b)

</script>

A:null null B:undefined undefined

C:undefined 报错 D:报错 报错

解析：a：定义未赋值，所以是undefined；b：没有定义

1. 分析以下代码，输出结果是（C）

<script type="text/javascript">

function Person(name,age){

this.name=name;

this.age=age;

}

Person.prototype.showName=function(){

console.log(this.name)

}

Person.prototype.showAge=function(){

console.log(this.age)

}

Person.prototype.LEAGS=222;

var p1=new Person('tom',18);

console.log(p1.LEAGS);

delete Person.prototype.LEAGS;

var p2=new Person('leo',20);

console.log(p2.LEAGS);

</script>

A:undefined undefined B:null null   
C:222 undefined D:222 222

解析：在这里LEAGS属性定义在构造函数的原型上，因此p1和p2都可以访问到，但是p1访问到之后LEAGS属性又被删除掉，那么p2就访问不到了，所以打印出undefined

1. 面向对象的特性有哪些（ABCD）

A：继承 B：多态 C：封装 D：抽象

1. 下列代码的输出结果是（C）

A：peter jack B：peter peter

C：jack peter D：jack jack

<script>

    var fun1 = function(){

        this.name = 'peter';

        return {

            name: 'jack'

        };

    }

    var p1 = new fun1();

    console.log(p1.name)

var fun2 = function(){

        this.name = 'peter';

        return 'jack';

    }

    var p2 = new fun2();

    console.log(p2.name)

</script>

解析：p1是通过fun1 new出来的一个实例，使用new生成实例时，会默认创建一个对象，并把对象返回回来，并且函数中的this就指向这个对象，但是本题中的fun1中，有返回值，并且返回值是一个对象，这个时候，就会覆盖掉new默认返回的对象。不过，如果使用了new，构造函数中也有返回值，但是这个返回值不是一个对象的话，就还会以new这个关键字默认返回的对象为主。

1. 分析下面代码的执行结果（A）

A：100 10 100 B：undefined undefined 100

C：100 undefined 100 D：10 undefined 100

var a=10;

function test(){

a = 100;

console.log(a);

console.log(this.a);

var a;

console.log(a)

}

test()

解析：第一次打印时，因为变量声明的提升，以及上边的a=100;就相当于将test自身的变量a的值赋为100，因此，第一次和第三次打印都是100，第二次打印主要在于this的指向，函数在调用时属于谁，或者说是谁调用的函数，函数内部的this就指向谁，在这里，函数调用时属于window，因此this就指向了window,而window上面的a的值是10,因此最终结果是100 10 100。

1. 在onreadystatechange事件处理函数中根据ajax对象的readyState属性的值做相应的处理,那么请问，在readyState属性的值等于1时表示处于那个步骤（A）

A：已经初始化，并调用了open方法

B：send方法已经调用

C：返回了部分数据

D：数据完全返回

解析：0 没有初始化,还没有调用open方法 1 启动,已调用open方法但是还没有调用send方法 2 发送,已调用send方法但还没有返回数据 3 接收,返回了部分数据 4 完成,数据完全返回

1. 判断下列说法，正确的是（ACD）

A:引用类型主要指保存在内存中的对象,包括对象,数组,函数

B:引用类型变量复制时会新建一个变量并分配空间,将变量的值分配在新的空间上

C:引用类型可以动态的添加属性

D:引用类型访问的是引用,引用是指向内存的一个指针

解析：引用类型复制时不会新建空间,只是复制了一个新的引用,基本类型变量复制时会新建一个变量并分配空间,将变量的值分配在新的空间上

1. 下列代码的执行结果是什么（C）

<script>

var i = 1;

while(i<5){

console.log(i);

i++;

}

</script>

A：0 1 2 3 4 B： 1 2 3 4 5      C：1 2 3 4    D：0 1 2 3

解析：这是一个while循环语句，语法为：

初始值;

while(循环退出条件){

循环体(语句)

循环退出条件的改变

}

1. function test(obj,name,age){console.log(arguments[1])},var obj=null;test(obj,”tom”,18)输出结果是（D）

A：null B：obj

C：18 D：tom

解析：每一个函数都有一个参数数组,数组的名字是arguments，参数数组中的值由调用时传递,按照传递参数的顺序存储在数组中，数组的下标从0开始，所以下标为1其实是数组中的第二个数

1. 以下设置盒子边框正确的是（CD）

A:border:1 dashed black;

B:border:1px black;

C:border:1px solid black;

D:border: black solid;

解析：围绕在元素四周的线条,边框有三要素,即宽度,样式,颜色，复合样式1(设置所有):border:宽度 样式 颜色。颜色可以省略，默认的是黑色、常见样式有dotted(点状),dashed(虚线),solid(实线)、宽度可以省略，默认是3px

1. 当我们给元素设置vertical-align属性时，该元素必须为（C）

A:行内元素

B:块状元素

C:行内块状元素

解析：vertical-align是用来设置元素的垂直对齐方式的，只有行内块状元素设置vertical-align属性才有效,图片默认就是行内块状元素

1. 分析以下代码的执行结果（C）

A：undefined B：null C:：2 D：1

<script>

var a=2;

function test(){

console.log(a);

}

function use(){

var a=1;

test();

}

use()

</script>

解析：1.函数中变量的值是在定义时决定的，不是在调用时决定的

2.子函数可以使用父函数中的变量

**二：判断 每题3分**

1. 判断：定义属性时，想要该属性是可以删除的，则可以在定义是添加configurable:true；属性和值（√）

解析：enumerable:true,//是否可以枚举,默认是false

writable:true,//是否可写(赋值),默认是false

configurable:true//是否可删除,是否可以修改属性的权限，默认是false

1. 判断：script标签中有两个封闭空间，通常的写法是()();()();，在这里“;”是可以省略的（×）

解析：这里的“;”不能省略，会报错

1. 判断：delete操作符用于删除程序中的所有变量或对象（×）

解析：delete不能删除使用VAR关键字声明的变量。

1. 判断：JS是一门解释性语言,运行在浏览器中,由浏览器的JS引擎负责解释运行JS代码。（√）
2. 判断：可以用js修改元素的任何属性（√）

**三：简答 每题8分**

1. 01-test.html内容如下：

<head>

<meta charset="UTF-8">

<title>01-test.html</title>

<script src="01-test.js"></script>

</head>

<body>

<div id="box"></div>

</body>

01-test.js内容如下：

function test(){

var oBox=document.getElementById(‘box’)

oBox.style.background = ‘red’;

}

var oBox=document.getElementById(‘box’)

oBox.onclick = test;

01-test.html和01-test.js在相同目录下；观察以上代码能否执行成功，如有不同意见，请写出方案？

解析：方法一：在script标签中添加defer属性

方法二：将script标签写到body标签的下边

方法三：将js中的代码写到window.onload里边

1. 请封装一个函数，用于计算0-100之间的偶数和，（包括0和100）？

解析：<script>

function sum(min,max){

var result=0;

for(var i=min;i<=max;i++){

if(i%2 == 0){

result+=i;

}

}

return result;

}

console.log(sum(1,100));

</script>

1. 现有数组arr = [1,0,3,4,0,6,0,8,9,7,0];要求如下：

将数组中的非0数字拿出来，并放入到一个新的数组中去，将新数组在控制台打印出来？

解析：方法一： var newArr = [];

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

if(arr[i] == 0) continue;

newArr.push(arr[i]);

}

console.log(newArr);

方法二：var newArr=arr.filter(function(value,index){

return value>0;

})

console.log(newArr);

1. 简答题：下列代码的输出结果是什么？

var a = 10,

b = 20,

c = 10;

console.log(a=b);

console.log(a==b);

console.log(a==c);

20，true，false

解析： 第一次打印其实是打印a这个变量，但是在这里将b的值赋给了a，于是打印20。

第二次打印时判断a等不等于b，因为之前的赋值，所以现在a等于 b，于是打印true。

第三次打印时，同样因为之前的赋值，a不等于c所以返回false。

1. 简答：什么是同源策略？

解析：不是一个源的文件操作另外一个源的文件就会形成跨域,即请求端的协议,域名,端口和服务器的协议,域名,端口有一个不一致就会发生跨域