# 一 选择题

1、分析下段代码输出结果是（b ）  
    var arr = [2,3,4,5,6];  
    var sum =0;  
    for(var i=1;i < arr.length;i++) {  
        sum +=arr[i]    }  
    console.log(sum);  
A．20     B．18     C．14     D．12  
  
2、以下关于 Array 数组对象的说法不正确的是（ c 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  
  
3、以下代码运行的结果是输出( b )  
    var a = b = 10;  
    (function(){  
        var a=b=20  
    })();  
    console.log(b);  
A．10     B．20     C．报错     D．undefined  
  
4、以下代码运行后的结果是输出(  b  )  
    var a=[1, 2, 3];  
    console.log(a.join());  
A．123     B．1,2,3     C．1  2  3     D．[1,2,3]  
5、在 JS 中，’1555’+3 的运行结果是(  c  )  
A．1558     B．1552     C．15553     D．1553  
  
6、以下代码运行后弹出的结果是( b )  
    var a = 888;  
    ++a;  
    alert(a++);  
A．888     B．889     C．890     D．891  
  
7、关于变量的命名规则，下列说法正确的是（ a,b,c,e ）  
A．首字符必须是大写或小写的字母，下划线（\_）或美元符（$）  
B．除首字母的字符可以是字母，数字，下划线或美元符  
C．变量名称不能是保留字  
D．长度是任意的  
E．区分大小写  
  
8、下列的哪一个表达式将返回值为假（  b  ）  
A．!(3<=1)  
B．(4>=4)&&(5<=2)  
C．(“a”==“a”)&&(“c”!=“d”)  
D．(2<3)||(3<2)  
  
9、下面代码，k的运行结果是（  b  ）  
    var i = 0,j = 0;  
    for(;i<10,j<6;i++,j++){  
        k = i + j;   
    }  
A．16     B．10     C．6     D．12  
  
10、var x = 1; function fn(n){n = n+1}; y = fn(x); y 的值为（  d  ）  
A．2     B．1     C．3     D．undefined  
  
11、[1,2,3,4].join('0').split('') 的执行结果是（  c  ）  
A．'1,2,3,4'                 
B．[1,2,3,4]    
C．[“1”,“0”,“2”,“0”,“3”,“0”,“4”]            
D．'1,0,2,0,3,0,4'  
  
12、下面代码的运行结果是：第一次弹（  a  ）第二次弹（  d  ）  
    function fn1() {  
        alert(1);  
    }  
    alert( fn1() );  
A．1       
B．alert(1);       
C．function fn1() { alert(1); }        
D．undefined  
  
13，以下代码运行后，结果为（  b  ）  
    fn1();  
    var fn1 = function(a){ alert(a); }  
A．1     B．程序报错     C．alert(1);     D．undefined  
  
14，var n = "miao wei ke tang".indexOf("wei",6)；n的值为：（  a  ）  
A．-1     B．5     C．程序报错     D．-10  
  
15，下面对 substring() 方法描述不正确的是（  c  ）  
A．一共有两个参数，省略第二个参数表示从参数开始位置提、截取到字符串结束。  
B．提取之前会比较两个参数的大小，并根据大小调整位置。  
C．可以接收负数参数，负数表示从后往前数字符位置。  
D．如果没有设置参数，直接返回整个字符串。  
  
16，alert( "12">"9" ) 的运行结果正确的是（  b  ）  
A．true     B．false  
  
17，下面的描述中不正确的是（  c  ）  
A．arguments 中保存了实际传入函数内的所有参数。  
B．return 只能在函数内部使用。  
C．setInterval(fn1,1000) 只会调用一次 fn1。  
D．Date 对象的 getMonth() 获取到的值比实际月份小1。  
  
18，下面的等式成立的是（  d  ）  
A．parseInt(12.5) == parseFloat(12.5)  
B．Number('') == parseFloat('')  
C．isNaN('abc') == NaN  
D．typeof NaN === 'number'  
  
19，下面的描述中不正确的是（ a b c ）  
A．'==' 在比较过程中，不但会比较两边的值，还会比较两边的数据类型。          
B．NaN == NaN 的结果是 true。  
C．isNaN，判断传入的参数是否为数字，为数字返回true，否则返回false   
D．字符串的 length 只可以获取，不可以设置。  
  
20，以下代码中，会出现什么结果：第一次弹（  a  ），第二次弹（  a  ），第三次弹（  b  ）  
    function fn1() {  
        var a = 0;  
        function fn2() {  ++a;  alert(a); }  
        return fn2;  
    }  
    fn1()();  
    var newFn = fn1();  
    newFn();  
    newFn();  
A．1     B．2     C．0     D．3  
  
21，下面代码的运行结果是：第一次弹（  d  ），第二次弹（  d  ）  
    var a = 100;  
    function fn1() {   
        alert(a);   
        var a = 10;  
    }    
    alert( fn1() );  
A．100  
B．10  
C．function fn1() { alert(1); }  
D．undefined  
  
22，以下代码运行后，arr的结果为（  a  ），arr2的结果为（  c  ）  
    var arr = [1,2];  
    var arr2 = arr.concat();      
    arr2.push( arr.splice(1,0) );  
A．[1,2]     B．[1,2,[2]]     C．[1,2,[]]     D．[1,2,3]  
23，下面关于数组的描述正确的是（  a  ）  
A．数组的 length 既可以获取，也可以修改。  
B．调用 pop() 方法，不会修改原数组中的值。  
C．shift() 方法的返回值是新数组的长度。  
D．调用 concat() 方法，会修改原数组的值。  
  
24，下列程序中alert按顺序分别弹出：（  b  ），（  a  ），（  b  ）  
    var a = 10;    
    function test() {    
        a = 100;    
        alert(a);    
        alert(this.a);    
        var a;    
        alert(a);   
    }  
    test();    
A．10     B．100     c．undefined     D．程序报错  
  
25，分析下面的代码，输出的结果是（  c  ）  
    var arr=new Array(5);   
    arr[1]=1;   
    arr[5]=2;   
    console.log(arr.length);   
A．2     B．5     C．6     D．报错  
  
26，在JavaScript中，下列（  d  ）语句能正确获取系统当前时间的小时值。   
A．var date=new Date();  var hour=date.getHour();   
B．var date=new Date();  var hour=date.gethours();   
C．var date=new date();  var hour=date.getHours();   
D．var date=new Date();  var hour=date.getHours();  
  
27，请选择结果为真的选项（  a  ）  
A．null == undefined  
B．null === undefined  
C．undefined == false  
D．NaN == NaN   
  
28, Math.ceil(-3.14)的结果是（  b  ），Math.floor(-3.14) 的结果是（  c  ）  
A．-3.14     B．-3     C．-4     D．3.14  
  
29, 阅读以下代码，在页面中结果是（  b  ）  
    var  s="abcdefg";  
    alert(s.substring(1,2));  
A．a     B．b     C．bc     D．ab  
  
30,  以下 ECMAScript 变量命名格式正确的是（  a  ）  
A．\_125dollor     B．1207A     C．-dollor     D．this

# 二 填空题

1 js中强制类型转换为number类型的方法有:\_\_\_\_number()\_\_\_\_\_\_\_\_. \_\_\_\_parseFloat()\_\_\_\_\_\_\_\_.\_\_\_\_parseInt()\_\_\_\_\_\_\_\_.

2 获取所有子元素节点: \_\_\_\_\_children\_\_\_\_\_\_\_.获取父节点\_\_\_\_\_\_parentNode\_\_\_\_\_\_.

3 把字符串按照指定字符分割并返回数组的方法为: \_\_\_\_split()\_\_\_\_\_\_\_\_.查询字符a在字符串‘fsdfsadf’中位置索引为：\_\_\_\_\_indexOf\_\_\_\_\_\_\_.

4 有x = 10,y=49，写出生成x到y直接随机数的方法：\_\_\_\_\_\_Math.random()\*39+10\_\_\_\_\_\_­­\_\_\_\_\_\_\_\_.

5 js中的多次执行的定时器为：\_\_\_\_\_\_setInterval\_\_\_\_\_\_.

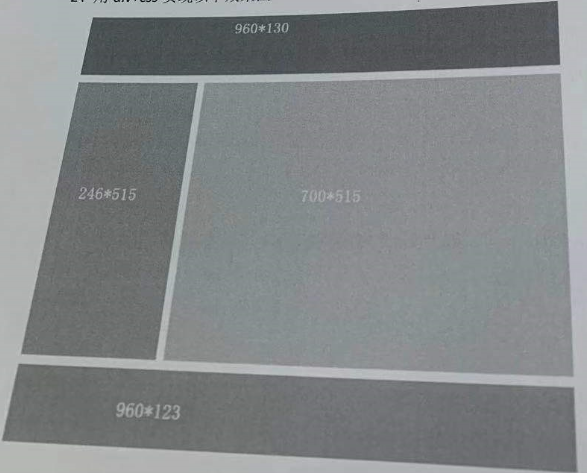
只执行一次的定时器为：\_\_\_setTimeout\_\_\_\_\_\_\_\_\_.

清除定时器的方法：\_\_\_\_\_clearInterval/clearTimeout\_\_\_\_\_\_\_.

6 js获取焦点事件为：\_\_\_\_onfocus\_\_\_\_\_\_\_\_.失去焦点事件为：\_\_\_\_onblur\_\_\_\_\_\_\_\_.

# 三 代码题

1 用div + CSS 实现以下效果



2 现有数组 var arr = [0,2,3,6,6,0,2],请编写一个方法，去掉其中的重复元素。

3 请我JS代码把数组 arr = [1,23,12,7,3,28,2,14]按正序的方式排序。

4 请使用原生DOM对象的方法，实现如下操作，创建一个内容为“姓名5”的p标签并插入到“姓名6”的p标签上面，同时删除最后一个p标签（注：不确定有多少个p标签）

<div id="a">

    <p>姓名 1</p >

    <p>姓名 2</p >

    <p>姓名 3</p >

    <p>姓名 4</p >

    <p>姓名 6</p >

<p>姓名 7</p >

……………

  </div>

# 四 简答题

1 .call()和.apply()的含义和区别?

都能改变函数的this指向，而且都会调用函数；

call 和 apply传递的参数不一样,call传递参数使用逗号隔开,apply使用数组传递

2 谈谈对原型和闭包的理解？

每个构造函数都有原型对象（prototype），原型对象下的constructor指向构造函数本身，构造函数实例化出来的实例对象有对象原型（\_\_proto\_\_），对象原型指向构造函数的原型对象。

每个对象都有对象原型\_\_proto\_\_，而他们通过\_\_proto\_\_连接起来形成的链子就叫原型链。

Object的原型对象（prototype）的\_\_proto\_\_指向null.

闭包：一个函数访问另一个函数作用域内的变量的函数就叫做闭包

闭包可以延长函数的生命周期

3 简单介绍一下promise,他解决了什么问题？

Promise解决了地狱回调的问题，但是它并没有简化代码，现在的话一般都推荐使用async await去代替promise。

4 普通函数和箭头函数this指向说明？

普通函数的this指向window对象，定时器中this指向window，事件函数中this指向触发事件的对象，构造函数中的this指向实例化对象，箭头函数中的this根据上下文环境决定。

5 let var const的区别是什么？

Var 可重复声明 没有块级作用域 顶级对象Window可以直接调用

Let 不可以重复声明 有块级作用域 有暂存性死区不能再声明之前调用

Const 定义的是常量 值不可以改变 有块级作用域 也有暂存性死区

6 说一下es6的新特性和es6新增的数组方法不少于6个？

新增自定义变量的方法 let const

箭头函数

展开运算符

类 class

对象解构

Set 方法 有效的解决了数组的值重复

7 ==和===有什么区别？JavaScript基本数据类型有哪些？

== 只判断两边的值是否相等，不判断数据类型

=== 不仅判断两边的值是否相等，还要判断两边的数据类型是否相等

JavaScript的基本数据类型都有 string number null Boolean undefined

8 什么是事件委托？请写出一个事件委托？

事件委托就是通过给父元素注册监听事件，利用事件冒泡的原理，让新增的元素也能触发监听事件

9 null 和 undefined的区别？

Null 表示没有这个值，undefined则表示定义的但是没有赋值，js在预解析的时候会提升变量，在赋值之前去调用这个声明的变量就会返回undefined

10 HTML5有哪些新特性？

新增了很多语义化标签 如：header Nav footer等

11 跨越问题如何解决？

1.通过 jsonp 跨域

2. CORS 核心思想：在服务器端通过检查请求头部的 origin，从而决定请

求应该成功还是失败。具体的方法 是在服务端设置 Response Header 响应头

中的 Access-Control-Allow-Origin 为对应的域名，实现了 CORS（跨域资源

共享），这里出于在安全性方面的考虑就是尽量不要用 \*，但对于一些不重要的

数据则随意。

12 ajax工作流程以及原理？

Ajax的原理简单来说通过XmlHttpRequest对象来向服务器发异步请求，

从服务器获得数据，然后用javascript来操作DOM而更新页面。这其中最

关键的一步就是从服务器获得请求数据

工作流程：

（1）实例化XMLHttpRequest对象

const xhr = new XMLHttpRequest()

(2) 调用open（）函数

xhr.open(‘get’,URL地址)

(3) 发送请求

xhr.send()

(4)监听readystatechange事件

xhr.onreadystatechange = function(){

if(xhr.readyState===4&&xhr.status===200){

console.log(xhr.responseText)

}

}