1. 分析下段代码输出结果是（ ）  
       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、以下代码运行后的结果是输出(   )  
       var a=[1, 2, 3];  
       console.log(a.join());   
   A．123     B．1,2,3     C．1  2  3     D．[1,2,3]

注意：a.join(“ “);= 1 2 3; a.join()=1,2,3  
3、在 JS 中，’1555’+3 的运行结果是(   )  
A．1558     B．1552     C．15553     D．1553  
  
4、以下代码运行后弹出的结果是(   )  
    var a = 888;  
    ++a;  
    alert(a++);  
A．888     B．889     C．890     D．891  
  
  
5、下列的哪一个表达式将返回值为假（    ）  
A．!(3<=1)  
B．(4>=4)&&(5<=2)  
C．(“a”==“a”)&&(“c”!=“d”)  
D．(2<3)||(3<2)  
  
6、下面代码，k的运行结果是（    ）  
    var i = 0,j = 0;  
    for(;i<10,j<6;i++,j++){  
        k = i + j;   
    }  
A．16     B．10     C．6     D．12  
  
7、var x = 1; function fn(n){n = n+1}; y = fn(x); y 的值为（   ）  
A．2     B．1     C．3     D．undefined  
  
8、[1,2,3,4].join('0').split("") 的执行结果是（   ）  
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'  
  
9、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  
  
10，以下代码运行后，结果为（    ）  
    fn1();  
    var fn1 = function(a){ alert(a); }  
A．1     B．程序报错     C．alert(1);     D．undefined  
  
11，var n = "miao wei ke tang".indexOf("wei",6)；n的值为：（   ）  
A．-1     B．5     C．程序报错     D．-10  
  
12，下面对 substring() 方法描述不正确的是（    ）  
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。  
  
13，alert( "12">"9" ) 的运行结果正确的是（    ）  
A．true     B．false  
  
14，下面的描述中不正确的是（   ）  
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。  
  
15，下面的描述中不正确的是（   ）  
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。  
  
16，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  
  
17，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  
  
18，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]

19，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。  
  
20，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  
  
21，分析下面的代码，输出的结果是（    ）  
    var arr=new Array(5);   
    arr[1]=1;   
    arr[5]=2;   
    console.log(arr.length);   
A．2     B．5     C．6     D．报错  
  
22，在JavaScript中，下列（    ）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();  
  
23，请选择结果为真的选项（    ）  
A．null == undefined  
B．null === undefined  
C．undefined == false  
D．NaN == NaN  
  
24, 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  
  
25, 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

上机：  
  
1，如何找到数组 [-1,-2,1,10,4,5,8] 中的最大值？

方式1：

let arr=[-1,-2,1,10,4,5,8];

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

console.log(arr[arr.length-1]);

方式2：

let arr=[-1,-2,1,10,4,5,8];

console.log(Math.max.apply(null,arr);

方式3：

2，封装一个函数，将字符串 "a-b-c-d" 从第二个单词开始首字母大写，然后拼成字符aBCD，并返回。（注意：封装成一个函数）

方式1：  
let str="a-b-c-d";

let arry=str.split("-");

let str1="";

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

if(i>0){

str1=str1+arry[i].toUpperCase();

}else{

str1=arry[i];

}

}

console.log(str1);

方式2：

let str="a-b-c-d";

let arry=str.split("-");

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

if(i>0){

arry[i]=arry[i].toUpperCase();

}else{

arry[i]=arry[i];

}

}

str=arry.join();

console.log(str);  
  
3 ,请用 js 代码完成下列需求：body 里生成 100 个 div，每个 div 宽 20px，高 20px，内容是 1-100 自身序号，颜色按红、黄、蓝、绿交替变色，10行10列排列。

css

.box { position:relative; }  
.box div { width: 20px; height: 20px; position: absolute;

html

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

script

var box=document.getElementById("box");

var arr=["red","yellow","blue","green"];

var str = '';

for (var i = 0; i <100; i++) {

    str += '<div style="left:' + i%10\*40 + 'px; top:' + Math.floor(i/10)\*40 + 'px; background:' + arr[i%arr.length] + ';">'+ (i+1)+'</div>';

}

console.log(box);

box.innerHTML += str;