目录

[1. 记忆函数-判断一个数是否是质数 1](#_Toc503257394)

[2. 数组插入排序 1](#_Toc503257395)

[3. 去掉数组重复的元素 2](#_Toc503257396)

[4. 字母尾部加上单位或数字 2](#_Toc503257397)

[5. 计算一个只有乘法和加法的表达式的值 2](#_Toc503257398)

[6. 查找字符串中出现次数最多的字符，共出现几次? 3](#_Toc503257399)

[7. 数组降维 3](#_Toc503257400)

[8. 无名 4](#_Toc503257401)

[9. 闭包-判断函数的输出结果 4](#_Toc503257402)

[10. 判断函数的输出结果 5](#_Toc503257403)

### 记忆函数-判断一个数是否是质数

**"use strict"**;

**let *isparime***=(**function**(){  
 **let** hash={};  
 **return function**(n){  
 **if**(!*isNaN*(n)&&n>1){  
 **if**(n<=3) **return true**;  
 **else if**(n%2==0) **return false**;  
 **else if**(hash[n]!==**undefined**){  
 ***console***.log(**"不用执行for循环。。"**);  
 **return** hash[n];  
 }**else**{  
 ***console***.log(**"执行for循环"**);  
 **for**(**var** i=3;i<=***Math***.sqrt(n);i+=2){  
 **if**(n%i==0){  
 hash[n]=**false**;  
 **return false**;  
 }  
 }  
 hash[n]=**true**;  
 **return true**;  
 }  
 }**else**{  
 **throw new *Error***(**"必须输入>1的数字"**);  
 }  
 }  
})();  
***console***.log(***isparime***(3));  
***console***.log(***isparime***(499));  
***console***.log(***isparime***(499));

1. 数组插入排序**let *arr***=[4,2,5,3,1,8,9,12,4,77];  
   **function** *insertStort*(arr) {  
    **for**(**let** i=1;i<arr.**length**;i++){  
    **let** t=arr[i];  
    **let** p=i-1;  
    **while**(p>=0&&arr[p]>t){  
    arr[p+1]=arr[p];  
    p--;  
    }  
    arr[p+1]=t  
    }  
   }  
   *insertStort*(***arr***);  
   ***console***.log(***arr***)
2. 去掉数组重复的元素**let *arr***=[3,1,3,1,4,2,3,9,4,5,2,8,2];  
    **let *r***=[];  
    **let *hash***={};  
    **for**(**let *i***=0;***i***<***arr***.length;***i***++){  
    **if**(***hash***[***arr***[***i***]]==**undefined**){  
    ***hash***[***arr***[***i***]]=1;  
    ***r***.push(***arr***[***i***])  
    }  
    }  
    ***console***.log(***r***)

### 字母尾部加上单位或数字

**let *str***=**"ryan is not a good man"**;  
*//ryan5 is6 not7 a8 good9 man10***let *n***=5;  
***str***=***str***.replace(/\b[a-z]+\b/ig,**function**(kw){  
 **return** kw+***n***++;  
});  
***console***.log(***str***);

### 计算一个只有乘法和加法的表达式的值

**function** *calc*(exp){  
 **let** arr=exp.split(**"+"**);  
 **let** sum=0;  
 **for**(**let** i=0;i<arr.length;i++){  
 **if**(arr[i].indexOf(**"\*"**)==-1)  
 sum+=*parseFloat*(arr[i]);  
 **else**{  
 **let** r=1;  
 **let** arr2=arr[i].split(**"\*"**);  
 **for**(**let** j=0;j<arr2.length;j++){  
 r\*=arr2[j]  
 }  
 sum+=r;  
 }  
 }  
 **return** sum;  
}  
***console***.log(*calc*(**"1+23\*2+3\*3\*3"**));

### 查找字符串中出现次数最多的字符，共出现几次?

**let *str***=**"abdcbcdadcabce"**;  
**let *hash***={};  
**for**(**let *i***=0;***i***<***str***.length;***i***++){  
 **if**(***hash***[***str***[***i***]]!==**undefined**)  
 ***hash***[***str***[***i***]]+=1;  
 **else  
 *hash***[***str***[***i***]]=1;  
}  
**let *zi***=**""**,***count***=0;  
**for**(**let *key*** in ***hash***){  
 **if**(***hash***[***key***]>***count***){  
 ***zi***=***key***;  
 ***count***=***hash***[***key***];  
 }  
}  
***console***.log(**"出现最多的字符:"**+***zi***+**",共"**+***count***+**"个"**);

### 数组降维

**let *arr***=[  
 [0,0,0,0],  
 [0,0,0,0],  
 [0,0,0,0],  
 [0,0,0,0]  
];  
*//arr=String(arr).split(",");****arr***=***Array***.prototype.concat.apply([],***arr***);  
***console***.log(***arr***);

### 无名

*//let f=function(){ let a= b=1; }  
//f();  
//console.log(b);//1  
//console.log(a);//报错  
  
//let f=function(){ let a=b=1; }  
//setTimeout(f,0);  
//console.log(b);//报错  
////f();  
  
//let a,b=0,fn=function(){let a= b=2};  
//fn();  
//console.log(a);//undefined  
//console.log(b);//2***let *emp***={  
 work:**function**(){  
 **let** sum=0;  
 **for**(**let** i=0;  
 i<***arguments***.length&&***arguments***[0]>0;  
 i++  
 ){  
 sum+=***arguments***[i]  
 +***arguments***.callee(--***arguments***[i])  
 }  
 **return** sum;  
 }  
};  
***console***.log(***emp***.work(3,2,1));

### 闭包-判断函数的输出结果

**let *a***=0,***b***=0;  
**function** *A*(a){*//外层函数  
 //外层函数向外返回内层函数——给全局变量A赋值  
 A*=**function**(b){***console***.log(a+b++);}*//内层函数* ***console***.log(a);  
}  
*A*(1);*//1  
//A:function(b){console.log(a+b++);}(a=1)  
A*(12);*//13?*

### 判断函数的输出结果

**1、**

**function** *fun*(n,o){  
***console***.log(o);*//输出第2个参数***return** {  
fun:**function**(m){*//(保护第1个参数)***return** *fun*(m,n);  
}  
}  
}  
**let *b***=*fun*(0)*//undefined  
/\*b:{fun(){return fun(m,n=0)}}\*/*.fun(1)*//0  
/\*b:{fun(){return fun(m,n=1)}}\*/*.fun(2)*//1  
/\*b:{fun(){return fun(m,n=2)}}\*/*.fun(3);*//2***let *c***=*fun*(0)*//undefined*.fun(1);*//0  
/\*c:{fun(){return fun(m,n=1)}}\*/****c***.fun(2);*//1****c***.fun(3);*//1***let *a***=*fun*(0);*//undefined  
/\*a:{fun(){return fun(m,n=0)}}\*/****a***.fun(1);*//0****a***.fun(2);*//0****a***.fun(3);*//0*

2、

**let *arr***=[];  
**function** *fun*(){  
 **for**(**let** i=0;i<4;i++){  
 **let** x={};  
 *//x.no=i;  
 //x.text=arr[i];* x.fun=**function**(){***console***.log(i)};  
 ***arr***.push(x);  
 }*//i=4  
 /\*  
 arr:[  
 {fun(){console.log(i)}},  
 {fun(){console.log(i)}},  
 {fun(){console.log(i)}},  
 {fun(){console.log(i)}},  
 ]  
 \*/*}  
*fun*();  
***arr***[0].fun();*//4****arr***[1].fun();*//4****arr***[2].fun();*//4****arr***[3].fun();*//4*

3、

**let *funs***=(**function**(){  
 **for**(**let** i=0,arr=[];i<3;i++){  
 arr[i]=**function**(){***console***.log(i)};  
 }*//i=3* **return *arr***;  
})();  
*//funs:[  
// 0: function(){console.log(i)},  
// 1: function(){console.log(i)},  
// 2: function(){console.log(i)}  
// ]****funs***[0]();*//3****funs***[1]();*//3****funs***[2]();*//3*