**数组：存储一组相关数据的容器**

**特性：**

**var *arr*=[90,85,87,65,60];**

*1.访问数组第0个数：*console.log(*arr*[0]);  
*2.数组个数：*console.log(*arr*.length);*//5  
length属性 长度可变  
 arr*[10]=100;  
console.log(*arr*.length);*//数组个数为11*

**赋值方式**  
1.  
var *arr*=[90,85,87,65,60];  
*2.*var *arr*=[];  
*arr*[0]=90;  
*arr*[1]=87;  
*arr*[2]=65;  
*arr*[3]=58;  
*arr*[4]=78;  
console.log(*arr*);

*实例化：*

*var arr2=new Array(5);  
 //一个参数表示数组的长度  
 //多个参数表示数组的内容  
//console.log(arr2);*

**遍历：**

*//求平均值*var *arr*=[];  
*arr*[0]=90;  
*arr*[1]=87;  
*arr*[2]=65;  
*arr*[3]=58;  
*arr*[4]=78;  
var *sum*=0;  
 for(var i=0;i<*arr*.length;i++){  
 *sum*+=*arr*[i];  
 ave=*sum*/*arr*.length;  
 }  
 console.log(ave);

数组当中可以放置任意类型的数据

*//选出数组中的数值：*var *arr4*=["123",5678,1,2,3,"as","asdfv",123];  
 var *newarr*=[];  
var *num*=0;  
 for(var *i*=0;*i*<*arr4*.length;*i*++){  
 if(typeof *arr4*[*i*]=="number"){  
 *newarr*[*num*]=*arr4*[*i*];  
 *num*++;  
 }  
 }  
console.log(*newarr*);  
 *//不及格的个数：* var *arr5*=[90,85,45,65,89,56,98,100];  
var *num*=0;  
 for(let *i*=0;*i*<*arr5*.length;*i*++){  
 if(*arr5*[*i*]<60){  
 *num*++;  
 }  
 }  
console.log(*num*);  
*//输出数组中的最大值*var *arr*=[90,85,45,65,89,56,98,100];  
var *max*=*arr*[0];  
 for(let *i*=0;*i*<*arr*.length;*i*++){  
 if(*arr*[*i*]>*max*){  
 *max*=*arr*[*i*];  
 }  
 }  
console.log(*max*);

同：

var *arr*=[90,85,45,65,89,56,98,100];  
var *max*=*arr*[0];  
for(let *i*=0;*i*<*arr*.length;*i*++){  
 *max*=*max*<*arr*[*i*]?*arr*[*i*]:*max*;  
}  
console.log(*max*);

二维数组：

如：var *arr*=[[1,2,3,4],[4,5,4,3],[1,3,5,7]];

怎么遍历二维数组中的每一个数；

var *arr6*=[[1,2,3,4],[4,5,4,3],[1,3,5,7]];  
 for(var *i*=0;*i*<*arr6*.length;*i*++) {  
 for(var *j*=0;*j*<*arr6*[*i*].length;*j*++ )  
 *console*.log(*arr6*[*i*][*j*]);  
}

怎么遍历混合型数组；

var *arr7*=[1,2,3,[3,56,7],[6,8,9]];  
for(var *i*=0;*i*<*arr7*.length;*i*++){  
 if(typeof *arr7*[*i*]=="number"){  
 *console*.log(*arr7*[*i*]);  
 }else{  
 for (var *j*=0;*j*<*arr7*[*i*].length;*j*++)  
 *console*.log(*arr7*[*i*][*j*]);  
 }  
}

怎么倒写。

var *arr*=[1,2,3,4,5,6];  
 var *newarr*=[];  
 var *num*=0;  
 for(var *i*=*arr*.length-1;*i*>=0;*i*--){  
 *newarr*[*num*]=*arr*[*i*];  
 *num*++;  
}  
console.log(*newarr*);

Return 当满足后面都不执行

利用递归函数遍历：  
function *bianli*(arr) {  
 for (var i=0;i<arr.length;i++) {  
 if (typeof arr[i] == "number") {  
 console.log(arr[i]);  
 } else {  
 *bianli*(arr[i]);  
 }  
 }  
}  
*bianli*([1,2,3,[4,5,6,[7,7,8],9],10]);  
  
 递归函数的应用：  
var *arr1*=[1,2,2,[4,5,6,[7,7,8],9],10];  
 function *bianli*(arr,num){  
 for(var i=0;i<arr.length;i++){  
 if(typeof arr[i]=="number"){  
 if(arr[i]==num){  
 return true;  
 }  
 }else{  
 var r=*bianli*(arr[i],num);  
 if(r){  
 return true;  
 }  
 }  
 }  
 return false;  
 }  
console.log(*bianli*(*arr1*,3));  
 选出不重复的数  
var *arr0*=[1,1,2,3,5,6,3,4,5];  
var *newarr*=[];  
 var *num*=0;  
 for(var *i*=0;*i*<*arr0*.length;*i*++){  
 if(!*bianli*(*newarr*,*arr0*[*i*])){  
 *newarr*[*num*]=*arr0*[*i*];  
 *num*++;  
 }  
 }  
console.log(*newarr*);  
  
排序：var *arr2*=[9,5,6,8,7,3,43,2,1,4,76,89,54];  
for(var *i*=0;*i*<*arr2*.length;*i*++){  
 for(var *j*=*i*;*j*<*arr2*.length;*j*++){  
 if(*arr2*[*i*]>*arr2*[*j*]){  
 var *aa*=*arr2*[*i*];  
 *arr2*[*i*]=*arr2*[*j*];  
 *arr2*[*j*]=*aa*;  
 }  
 }  
}  
 console.log(*arr2*);

Es6 新语法：

扩展运算符：将数组转化为一个参数序列 （...）

1.

var *arr*=[1,2,3,4,5];  
 *console.log(...arr);//1,2,3,4,5*

2.var *arr*=[1,2,3,4,5];  
 function *sum*(){  
 var he=0;  
for (i = 0; i < *arr*.length; i++) {  
 he += *arr*[i];  
}  
return he;  
}  
 console.log(*sum*(...arr));

解构赋值：

*//解构赋值：  
 // 1.*var [*a*,*b*,*c*]=[1,2,3];  
console.log(*a*);  
console.log(*b*);  
console.log(*c*);  
*// 1.*var [*a*,*b*,*c*]=[1,2];  
console.log(*a*);  
console.log(*b*);  
console.log(*c*);  
*// 1.*var [*a*,*b*,*c*=3]=[1,2];  
console.log(*a*);  
console.log(*b*);  
console.log(*c*);  
*//1*var [*a*=1,*b*,*c*]=[undefined,1,2];  
console.log(*a*);  
console.log(*b*);  
console.log(*c*);  
*//1*var [*a*,...b]=[1,2,3,4,5];  
console.log(*a*);  
console.log(*b*);