

task18

问答

1.数组方法里push、pop、shift、unshift、join、split分别是什么作用。（*）

push() 给数组添加一个新元素并成为最后一个元素

pop() 删掉数组原有最后一个元素

unshift() 给数组添加一个新元素并成为第一个元素

shift() 删掉数组原有第一个元素

join() 把数组中的所有元素放入一个字符串

split() 把一个字符串分割成字符串数组

代码题

数组

1.用 splice 实现 push、pop、shift、unshift方法（*）

```
var arr= [];  
arr[0] = "a"  
arr[1] = "b"  
arr[2] = "c"  
arr[3] = "d"  
console.log(arr)  
arr.splice(0,0,"first one") //unshift()  
console.log(arr)  
arr.splice(0,1) //shift()  
console.log(arr)  
arr.splice(arr.length,0,"last one") //push()  
console.log(arr)  
arr.splice(arr.length-1,1) //pop()  
console.log(arr)  
  
//以下是调试结果  
["a", "b", "c", "d"]  
["first one", "a", "b", "c", "d"]  
["a", "b", "c", "d"]  
["a", "b", "c", "d", "last one"]  
["a", "b", "c", "d"]
```

2.使用数组拼接出如下字符串（*）

```
var prod = {
  name: '女装',
  styles: ['短款', '冬季', '春装']
};
console.log('<dl class="product">');
console.log('<dt>'+prod.name+'</dt>')
var arr =[];
function getTpl(data){
  var stylesArr = prod.styles;
  for(var i=0;i<stylesArr.length;i++){
    arr.push(stylesArr[i])
    console.log('<dd>'+arr[i]+'</dd>')
  }

};
var result = getTpl(prod);
console.log('</dl>');
```

3.写一个find函数，实现下面的功能（*）

```
var arr = [ "test", 2, 1.5, false ]
find(arr, "test") // 0
find(arr, 2) // 1
find(arr, 0) // -1

function find (arr,x){
  return arr.indexOf(x);
}
find();
```

4.写一个函数filterNumeric，把数组 arr 中的数字过滤出来赋值给新数组newarr，原数组arr不变（**）

```
var arr = ["a", "b", 1, 3, 5, "b", 2];
function filterNumeric(){
  var newarr =[];
  for(var i=0;i<arr.length;i++){
    if(typeof(arr[i])=="number"){
      newarr.push(arr[i]);
    }
  }
  return newarr;
}
```

```
newarr = filterNumeric(arr);
```

5.对象obj有个className属性，里面的值为的是空格分割的字符串(和html元素的class特性类似)，写addClass、removeClass函数，有如下功能：(**)

```
var obj = {
  className: 'open menu'
};
//添加的函数
function addClass(obj,x){
  var arr = obj.className.split(' ');
  if(arr.indexOf(x)===-1){
    arr.push(x);
    obj.className=arr.join(' ');
  }
  console.log(obj.className);
}

//删减的函数
function removeClass(obj,x){
  var arr = obj.className.split(' ');
  if(arr.indexOf(x)!==-1){
    arr.splice(arr.indexOf(x),1);
    obj.className=arr.join(' ');
  }
  console.log(obj.className);
}

addClass(obj, 'new') ;
addClass(obj, 'open') ;
addClass(obj, 'me') ;
console.log(obj.className) ;
removeClass(obj, 'open') ;
removeClass(obj, 'blabla') ;
```

6.写一个camelize函数，把my-short-string形式的字符串转化成myShortString形式的字符串，如 (*)

```
function camelize(str){
  var arr = str.split('-');
  for(var i=1;i<arr.length;i++){
    arr[i]=arr[i].replace(arr[i][0],arr[i][0].toUpperCase());
  }
  return arr.join('');
}
```

```

}
return arr.join('');
}

camelize("background-color") == 'backgroundColor'
camelize("list-style-image") == 'listStyleImage'

```

7.如下代码输出什么？为什么？(*)

```

arr = ["a", "b"];
arr.push( function() { alert(console.log('hello hunger valley')); } );
arr[arr.length-1]() ; // ?

```

//中间那一行，加进去的东西，先看最里边的括号（因为括号的优先级）是一个console，输出'hello hunger valley'，然后返回值是一个undefined给外层括号的alert，于是alert就弹出一个undefined。

8.写一个函数isPalindrome,判断一个字符串是不是回文字符串（正读和反读一样，比如 abcdcba 是回文字符串， abcdefg不是）

```

function isPalindrome(str){
  var arr = str.split('');
  var arrTwo = [];
  for(var i=0;i<arr.length;i++){
    arrTwo.unshift(arr[i]);
  }

  var x = arr.join("");
  var y = arrTwo.join("");

  if(x===y){
    return true;
  }else{
    return false;
  }
}

```

9.写一个ageSort函数实现数组中对象按age从小到大排序（*）

```

var john = { name: "John Smith", age: 23 };
var mary = { name: "Mary Key", age: 18 };

```

```

var bob = { name: "Bob-small", age: 6 };
var people = [ john, mary, bob ];
function ageSort(people){
people.sort(function(a,b){return a.age-b.age;});
for(var i=0;i<people.length;i++){
    console.log(people[i].name);
}
}
ageSort(people);

```

10.写一个filter(arr, func) 函数用于过滤数组，接受两个参数，第一个是要处理的数组，第二个参数是回调函数(回调函数遍历接受每一个数组元素，当函数返回true时保留该元素，否则删除该元素)。实现如下功能：（**）

```

function isNumeric (el){
    return typeof el === 'number';
}
var arr = ["a",3,4,true, -1, 2, "b"];

function filter(arr, func){
    for(var i=0;i<arr.length;i++){
        if(!func(arr[i])){
            arr.splice(i,1);
        }
    }
    return arr;
}

arr = filter(arr, isNumeric) ; // arr = [3,4,-1, 2]
arr = filter(arr, function(val) { return  typeof val === "number" && val > 0; }
); // arr = [3,4,2]

```

字符串

1.写一个 ucFirst函数，返回第一个字母为大写的字符（*）

```

function ucFirst(str){
    var arr = str.split('');
    arr[0] =arr[0].replace(arr[0],arr[0].toUpperCase());
    arr=arr.join('');
    console.log(arr);
}

```

```
}  
  
ucFirst("hunger");
```

2.写一个函数truncate(str, maxlength), 如果str的长度大于maxlength, 会把str截断到maxlength长, 并加上..., 如
(**)

```
function truncate(str,maxlength){  
  if(str.length>maxlength){  
    return str.substr(0,maxlength)+"...";  
  }  
  return str  
}  
  
truncate("hello, this is hunger valley,", 10)) == "hello, thi...";  
truncate("hello world", 20)) == "hello world"
```

数学函数

1.写一个函数, 获取从min到max之间的随机整数, 包括min不包括max (*)

```
function calculate(min,max){  
  var a = Math.floor(Math.random() * (max-min)) + min;  
  console.log(a);  
}
```

2.写一个函数, 获取从min都max之间的随机整数, 包括min包括max (*)

```
function calculate(min,max){  
  var a = Math.floor(Math.random() * (max-min+1)) + min;  
  console.log(a);  
}
```

3.写一个函数, 获取一个随机数组, 数组中元素为长度为len, 最小值为min, 最大值为max(包括)的随机整数 (*)

```
function randonArr(len,min,max){
```

```
var r;  
var arr = [];  
for(var i=0;i<len;i++){  
    r = Math.floor(Math.random() * (max-min+1)) + min;  
    arr.push(r);  
}  
console.log(arr);  
}  
  
randonArr(5,1,10);
```

4.写一个函数，生成一个长度为 n 的随机字符串，字符串字符的取值范围包括0到9，a到z，A到Z。

```
function randonArr(n){  
    var x = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';  
    var arr=x.split('');  
    var arrTwo=[];  
  
    for(var i=0;i<n;i++){  
        var y=Math.floor(Math.random() * (arr.length)) + 1;  
        arrTwo.push(arr[y]);  
    }console.log(arrTwo.join(''));  
  
}  
  
randonArr(777);
```



Leanote
Upgrade Account