**四、javascript数组**

**一、认识数组**

**1.数据类型：**

(1)基本数据类型

①数字

②字符串

③布尔值

④特殊数据类型 NaN undefined null

(2)复合数据类型：

数组

**2.数组：**

(1)作用：通过1个变量存储一系列的值。

(2)数组的声明

①通过new运算符生成数组 参数是，需要去存储的数据

元素：每一个存储在数组中的数据，我们叫做元素

【注】如果所传参数只有一个，并且这个参数是数字，那么这个参数并不是去存储元素。代表生成数组的长度

\*/

var arr = new Array(10);

alert(arr);

/\*var arr = new Array(10, "hello", true);

alert(arr);

/\*

②省略new运算符

\*/

/\*var arr = Array(10, "hello", true);

alert(arr);\*/

/\*

③使用常量声明数组

\*/

/\*var arr = [10, "hello", true];

alert(arr);\*/

**二、数组的属性**

**1.数组的属性：**

**length** 数组中存储的元素的个数

访问数组中的元素，可以通过下标的方式访问

【格式】

arr[index] index下标 下标是从0开始的

【注】数组和循环更配。

\*/

var arr = ["钢铁侠", "蜘蛛侠", "原子侠", "闪电侠"];

// alert(arr.length);

// alert(arr[0]);

//访问数组中的每一个元素

for(var i = 0; i < arr.length; i++){

alert(arr[i]);

}

//赋值

arr[2] = "阿童木";

alert(arr);

**2.随机数**

//Math.random() 随机 0~1的任意数

//随机10个0~10的任意整数，存到数组中

function random(n){

var arr = [];

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

arr[i] = parseInt(Math.random() \* 10);

}

return arr;

}

alert(random(5));

**三、数组的使用**

**1.使用for循环遍历**

var arr = [1, 2, 3, 4, 5];

for(var i = 0; i < arr.length; i++){

alert(arr[i]);

}

**2.快速遍历**

**【方式】for...in** //i是字符串

for(var i in arr){

alert(arr[i]);

}

**四、数组的函数**

**1.栈结构**

栈 古代木盆

**栈特点：先进后出。**

(1)push: 将元素添加在数组的末尾。

【格式】数组.push(元素...);

【返回值】是添加完元素以后数组的长度。

/\*var arr = ["钢铁侠", "蜘蛛侠", "死侍"];

var tmp = arr.push("绿巨人", "黑寡妇");

// alert(arr);

alert(tmp);\*/

/\*

(2)pop: 在数组的末尾，取元素

【格式】数组.pop();

【返回值】是取下来的元素。

var arr = ["钢铁侠", "蜘蛛侠", "死侍"];

var tmp = arr.pop();

alert(tmp);

// alert(arr);

**2.队列**

**特点：先进先出。**

(1)shift:从数组的头部取元素。

【格式】

数组.shift();

【返回值】取下的元素

var arr = ["钢铁侠", "蜘蛛侠", "死侍"];

var tmp = arr.shift();

alert(tmp);

alert(arr);

(2)unshift: 从数组头部添加元素

【格式】

数组.unshift(元素...)

【返回值】添加完元素以后，数组的长度。

var arr = ["钢铁侠", "蜘蛛侠", "死侍"];

var tmp = arr.unshift("蚁人", "超人");

alert(tmp);

// alert(arr);

**五、数组方法**

**1.concat()**

【作用】合并数组，生成一个新数组，原数组不会改变

【格式】数组1.concat(数组2);

/\*var arr1 = [10, 20];

var arr2 = [40, 50];

var arr = arr1.concat(arr2);

alert(arr);

alert(arr1);

alert(arr2);

**2.join()**

【作用】将数组拼接成字符串。

【参数】拼接符。

【返回值】拼接好的字符串。

var arr = ["钢铁侠", "绿巨人", "蜘蛛侠"];

var str = arr.join("\*\*\*\*");

alert(str); //钢铁侠\*\*\*\*绿巨人\*\*\*\*蜘蛛侠

**3.splice()**

【格式】

数组.splice()

参数1：开始截取的下标

参数2：截取的元素的长度

参数3：我要插入的元素

【返回值】截取掉的元素组成的数组

//(1)删除功能

var arr = ["钢铁侠", "绿巨人", "蜘蛛侠", "死侍"];

var tmp = arr.splice(1, 2);

alert(arr);//钢铁侠,死侍

alert(tmp);//绿巨人,蜘蛛侠

//(2)插入

arr.splice(2, 0, "蚁人", "xxx");

alert(arr);//钢铁侠,绿巨人,蚁人,xxx,蜘蛛侠,死侍

//(3)替换

arr.splice(2, 1, "奇异博士");

alert(arr);

**4.slice(start, end);**

start: 开始提取元素的下标位置

end: 结束提取元素的下标位置，不包括这个位置

【功能】在指定范围内，提取数组的元素，并且生成新数组，但是对原数组不造成影响。

【格式】数组.slice(start, end);

var arr = ["钢铁侠", "绿巨人", "蜘蛛侠", "死侍"];

var newArr = arr.slice(1, 3);

alert(newArr);

alert(arr);

**五、补充知识-引用（指针）**

数组

复合数据类型/【引用】数据类型/都是对象

【注】函数内部的变量个形参的作用域和声明周期是一样的。

【注】函数的形参相当于是函数内部声明的变量。

【注】基本数据类型和函数运行的空间在栈段。(一旦分配就无法修改)

【注】复合数据类型运行的控件在堆段。（可以任意增大缩小）

【注】基本数据类型存储的数据，复合数据类型，存储的房间号。

/\*var a = 10;

function show(a){ //a = 10

a = 5;

alert(a);

}

alert(a); //10

show(a); //5

alert(a); //10\*/

//========================================

var arr = ["钢铁侠", "蜘蛛侠"];

function show(arr){ //arr = 房间号

arr.push("蚁人");

alert(arr);

}

alert(arr); //钢铁侠,蜘蛛侠

show(arr); //钢铁侠,蜘蛛侠,蚁人

alert(arr); //钢铁侠,蜘蛛侠,蚁人

**六、系统的排序方法**

**1.reverse()**

【作用】逆序

【格式】数组.reverse()

\*/

/\*var arr = [10, 2, 3, 4];

arr.reverse();

alert(arr);\*/

/\*

**2.sort()**

【注】按照字符串从小到大排序。

【注】从小到大排序

【格式】数组.sort();

\*/

//var arr = [5, 2, 3, 4];

var arr = [15, 1, 5, 10];

arr.sort();

//1,10,15,5

alert(arr);

**3.冒泡排序**

冒泡排序

【规律】从头到尾依次比较相邻两个数字大小，若大数在前、小数在后，则交换两数位置，依次比较，使全部数据按从小到大排列

【特点】每一轮排序都会选出一个对的数，放在正确的位置。

第一轮：

9, 8, 7, 6, 5, 4

8, 9, 7, 6, 5, 4

8, 7, 9, 6, 5, 4

8, 7, 6, 9, 5, 4

8, 7, 6, 5, 9, 4

8, 7, 6, 5, 4, 9

第二轮：

8, 7, 6, 5, 4

7, 8, 6, 5, 4

7, 6, 8, 5, 4

7, 6, 5, 8, 4

7, 6, 5, 4, 8

第三轮：

7, 6, 5, 4

6, 7, 5, 4

6, 5, 7, 4

6, 5, 4, 7

第四轮：

6, 5, 4

5, 6, 4

5, 4, 6

第五轮：

5, 4

4, 5

var arr = [9, 8, 7, 6, 5, 4];

function bubble(arr){

for(var i = 0; i < arr.length - 1; i++){ //比较的轮数

for(var j = 0; j < arr.length - i - 1; j++){

if(arr[j] > arr[j + 1]){

var tmp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = tmp;

}

}

}

}

bubble(arr);

alert(arr);

**4.选择排序**

通过比较首先选出最小的数放在第一个位置上，然后在其余的数中选出次小数放在第二个位置上,依此类推,直到所有的数成为有序序列。

第一轮：

9, 8, 7, 6, 5, 4

8, 9, 7, 6, 5, 4

7, 9, 8, 6, 5, 4

6, 9, 8, 7, 5, 4

5, 9, 8, 7, 6, 4

4, 9, 8, 7, 6, 5

第二轮：

9, 8, 7, 6, 5

8, 9, 7, 6, 5

7, 9, 8, 6, 5

6, 9, 8, 7, 5

5, 9, 8, 7, 6

var arr = [9, 8, 7, 6, 5, 4];

function select(arr){

for(var i = 0; i < arr.length - 1; i++){

for(var j = i + 1; j < arr.length; j++){

if(arr[i] > arr[j]){

var tmp = arr[i];

arr[i] = arr[j];

arr[j] = tmp

}

}

}

}

alert(arr);

select(arr);

alert(arr);

**5.数组去重**

编写函数norepeat(arr) 将数组的重复元素去掉，并返回新的数组

【注】不能正序删除数组。

【注】数组必须倒序删除。

function norepeat(arr){

for(var i = 0; i < arr.length - 1; i++){

for(j = arr.length - 1; j > i; j--){

if(arr[i] == arr[j]){

//删除j

arr.splice(j, 1);

}

}

}

}

var arr = [20, 30, 40, 30, 20, 20, 20];

norepeat(arr);

alert(arr);



**6.打印矩阵**

1、通过循环按行顺序为一个5×5的二维数组a赋1到25的自然数，然后输出该数组的左下半三角。试编程。

var arr = [];

var count = 0;

for(var i = 0; i < 5; i++){

var rows = [];

for(var j = 0; j < 5; j++){

rows.push(++count);

}

document.write(rows + "<br />");

arr.push(rows);

}

//alert(arr);

for(var i = 0; i < 5; i++){

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

document.write(arr[i][j] + ",");

}

document.write("<br />");

}

/\*var arr = [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]];

var newArr = arr[0];

arr[0][1];\*/

**7.简易计算器**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style>

div{

width:500px;

height:600px;

margin:100px auto 0;

text-align:center;

padding-top:50px;

}

#result{

width:400px;

height:70px;

margin:20px 0;

font-size:35px;

}

button{

width:100px;

height:50px;

font-size:30px;

font-weight:600;

}

</style>

<script>

function operation(num1, str, num2){

switch(str){

case "+":

return num1 + num2;

break;

case "-":

return num1 - num2;

break;

case "\*":

return num1 \* num2;

break;

case "/":

return num1 / num2;

break;

default:

return "error";

break;

}

}

function $(id){

return document.getElementById(id);

}

var arr = [];

function btnClick(str){

arr.push(str);

$("result").value = arr.join("");

}

function equal(){

var index = 0;

for(var i = 0; i < arr.length; i++){

if(!(arr[i] >=0 && arr[i] <= 9)){

var index = i;

break;

}

}

var arr1 = arr.slice(0, index);

var arr2 = arr.slice(index + 1, arr.length);

var num1 = Number(arr1.join(""));

var num2 = Number(arr2.join(""));

$("result").value = operation(num1, arr[index], num2);

arr = [];

}

function clearBtn(){

$("result").value = "";

arr = [];

}

</script>

</head>

<body>

<div>

<input id = "result" type="text" />

<br />

<button onclick = "btnClick(7);">7</button>

<button onclick = "btnClick(8);">8</button>

<button onclick = "btnClick(9);">9</button>

<button onclick = "btnClick('+');">+</button>

<br />

<button onclick = "btnClick(4);">4</button>

<button onclick = "btnClick(5);">5</button>

<button onclick = "btnClick(6);">6</button>

<button onclick = "btnClick('-');">-</button>

<br />

<button onclick = "btnClick(1);">1</button>

<button onclick = "btnClick(2);">2</button>

<button onclick = "btnClick(3);">3</button>

<button onclick = "btnClick('\*');">\*</button>

<br />

<button onclick = "btnClick(0);">0</button>

<button onclick = "clearBtn();">C</button>

<button onclick = "equal();">=</button>

<button onclick = "btnClick('/');">/</button>

</div>

</body>

</html>