

JavaScript 第六章

本章主要内容

1. JS中的内置对象
2. 数组 Array
3. 数学对象 Math
4. 时间对象 Date
5. 正则对象 RegExp

JS中的内置对象

在JS中有一个特殊类型的数据是定义好的对象，直接拿来使用

包含有数组、数学对象、时间对象、正则对象

数组 Array

概念

数组是存储数据的仓库，每个储藏室都有一个数字编号(索引 index)

定义方式

1. 使用 new Array()定义数组

```
// 如果定义时只传入一个数字，表示声明一个长度为该数字的数组，默认值为undefined
var arr = new Array(10);

// 常规定义
var users = new Array('二郎神', '孙悟空', '猪八戒');
```

2. 使用[]定义数组

```
// 定义数组
var arr = ['玉皇大帝', '王母', '宋江'];
```

数组长度

```
// 在数组对象中有属性length  
var len = arr.length;
```

索引 index

1. 每一个数组单元的值都是有一个编号对应的，这个编号就称之为索引(index)
2. 索引的值从0开始
3. 索引的最大值为数组长度-1

数组遍历

1. 使用循环遍历数组

```
for(var i=0;i<arr.length;i++){  
    // i 当前数组的索引  
    // arr[i] 就是当前数组的值  
}
```

2. 使用forEach()函数遍历数组

```
arr.forEach(function(value,index,arrSelf){  
    /*  
        第一个参数 value 获取的是当前数组单元的值  
        第二个参数 index 获取的是当前数组单元的索引  
        第三个参数 arrSelf 获取的是当前数组本身  
    */  
})
```

数组常用方法

1. Array.isArray() 判断是否是数组

```
如果是数组，返回true，  
如果不是数组，返回false
```

2. concat() 连接两个或多个数组，返回结果

```
// 连接值  
arr.concat('美女');  
// 连接数组  
arr.concat([2,3],[4,3]);
```

3. join() 将数组单元的值拼接成字符串，可以指定连接符

```
// 默认返回以逗号(,)分割的字符串  
arr.join(',')
```

4. reverse() 翻转数组的单元的顺序

```
arr.reverse()
```

5. toString() 将数组转为字符串，并返回结果

6. pop() 弹出数组最后一个单元的值，并返回

7. push() 从数组的尾部压入值，返回新的数组长度

8. shift() 删除并返回数组第一个单元的值

9. unshift() 从数组头部压入值，并返回数组的新长度

10. slice() 从数组中截取

11. splice() 删除数组，并向数组中添加元素

12. indexOf() 正序查询数组单元的值

如果存在，返回符合查询条件的第一个结果所在的位置

13. lastIndexOf() 倒序查询数组单元的值

如果存在，返回符合查询条件的第一个结果所在的位置
如果不存在，返回-1

14. map() 映射，原数组单元经回调函数处理后，返回新的结果

```
arr.map(function(value){
    return value*value;
})

// 使用map()获取对象数组中特定的属性值
var users = [
    {
        username : '宋江',
        email : 'songjiang@liangshan.com'
    },
    {
        username : '吴用',
        email : 'zhiduoxing@liangshan.com'
    }
]

// 获取邮箱地址
users.map(function(user){
    return user.email;
})
```

15. filter() 过滤，返回符合条件的结果的值

```
// 获取数组中的偶数
arr.filter(function(i){
    // 返回真，对应的值i，就会返回到新数组中
    return i%2==0;
})
```

16. some() 筛选，只要有一个符合条件就返回true

```
// 只要有一个男的，就返回真
var users = ['男','女','女'];

// 此时res的值为true
var res = users.some(function(value){
    if(value == '男'){
        return true;
    }
});
```

17. every() 筛选，只有当每一个单元都符合条件时才整体返回真

```
// 判断是否全部是男生
var users = ['男','男','女'];

// 此时res的值为false
var res = users.every(function(value){
    if(value == '男'){
        return true;
    }
});
```

18. sort() 数组排序

在使用sort()函数时，需要传入对应的比较函数，作为判断条件

实战案例

1. 计算数组单元的个数
2. 翻转数组
3. 查询出如下格式数组中所有money>1000的人

```
// 数组格式
var uses = [
    {
        username : '张三',
        money : 10000
    },
    ...
];
```

4. 自定义数组排序函数--冒泡排序