

参考网址：

[https://www.w3school.com.cn/jsref/jsref\\_obj\\_array.asp](https://www.w3school.com.cn/jsref/jsref_obj_array.asp)

## 对象的定义

### 创建对象

```
1 var obj = new Object();
```

### 查看对象原型

```
1 console.log(obj.__proto__);  
2  
3 // 通过原生方法Object查看对象原型  
4 console.log(Object.prototype)
```

### 在对象原型下添加方法

```
1 // 对象原型下添加的方法,只有有实例化对象,则该方法就可以使用  
2 Object.prototype.say = function(){  
3   console.log('好想出门! 家里没粮了')  
4 }  
5  
6 var person = new Object()  
7 console.log(person)
```

## 数组

数组：一段有序的数据的集合，多个元素之间采用逗号隔开

数组的定义方式：

方式一：

```
1 var arr = new Array();  
2  
3 // 若只有一个数字作为参数:该数字为创建数组的长度  
4 var arr = new Array(5);  
5 console.log(arr)  
6
```

```
7 //该参数不是数字，则代表数组内实际的值
8 var arr = new Array('宋鑫');
9 console.log(arr)//["宋鑫"]
10
11 // 若参数超过一个,则代表数组具体的值
12 var arr = new Array(1,2,3);
13 console.log(arr)//[1, 2, 3]
```

方式二：

```
1 var arr = [];
```

## length为数组的长度

数组的下标为数字，默认从0开始递增

数组最大的下标+1=数组的长度

访问数组内的值（通过其下标访问）

```
1 arr[i] i代表指定的下标
```

添加指定下标的值

```
1 arr[4] = '马洁';
2 console.log(arr)//["宋鑫", "陈颖", "陈力", "陈易", "马洁"]
```

修改指定下标的值

```
1 // 修改数据 如果下标已存在,则修改指定下标的值
2 arr[2] = '唐元鹏';
3 console.log(arr)//["宋鑫", "陈颖", "唐元鹏", "陈易", "马洁", empty × 5, "瑶瑶姐"]
```

删除指定下标的值 delete()

注意：delete只是将对应下标的值进行删除，保留键

```
1 delete(arr[0])
2 console.log(arr);//[empty, "陈颖", "唐元鹏", "陈易", "马洁", empty × 5, "瑶瑶姐"]
```

二维数组

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

## 数组的遍历（重点）

迭代（遍历）：

### 1、显示迭代:for循环

```
1 for(var i=0;i<arr.length;i++){
2   console.log(arr[i])
3 }
```

### 2、隐式迭代

```
1 arr.forEach(function(val,key,ownObj){
2   //第一个参数为数组的值
3   //第二个参数为数组的下标
4   //第三个参数为当前遍历的数组
5 })
```

## 对象的遍历（重点）

```
1 for...in...
2 语法: for(var 变量名 in 遍历的对象){
3  }
4
5 注意: 该变量名则为对象的属性名和方法名（键）
```

```
1 for(var key in obj){
2   console.log(key)//key为当前对象的属性名和方法名
3   console.log(obj[key])//obj[key]可以取出对应的属性值和方法
4 }
```

拓展作业：

### 去除数组中的undefined的值

封装一个函数，传入一个数组，将数组内值为undefined的去除，返回去除undefined之后的结果

```
1 var arr = [1,2,undefined,5,undefined,undefined,9,null];
2 function fun(){
3   var newArr = [];
```

```
4  for(var i=0;i<arr.length;i++){
5  // 判断值是否为undefined
6  if(arr[i] !== undefined){
7
8  // 数组最大下标 + 1 = 数组的长度
9  newArr[newArr.length] = arr[i];
10 }
11 }
12 return newArr;
13 }
14 var res = fun(arr)
15 console.log(res)//[1,2,5,9]
16 console.log(res.length)//4
```

## 数组函数

### 函数的三要素：功能、参数、返回值

```
1  *concat()连接两个或更多的数组，并返回结果
2  *join(指定的字符)将数组按照指定的字符拼接成字符串（将数组转换为字符串）
3  注意：join默认按照','进行拼接
4
5  *push()向数组的末尾添加一个或更多元素，并返回新的长度（多个元素采用逗号连接）
6  *pop()删除并返回数组的最后一个元素（不需要传参）
7  *unshift()向数组的开头添加一个或更多元素，并返回新的长度。（多个元素采用逗号连接）
8  *shift()删除并返回数组的第一个元素（不需要传参）
9
10 *slice(start,end)数组的截取
11 slice(start)（第一个参数必填）从指定下标（start）开始截取，截取到最后
12 slice(start,end)从指定下标（start）开始截取,截取到指定结束下标（end）之前
13 注意：
14 1、在截取数组时，包含起始下标（start），不包括结束下标（end）
15 2、截取的下标可以为负数，代表倒数第几个
16 3、slice截取为从左往右截取
17 *splice(index,num,item)删除、修改或者添加数组的元素
18 index:代表从指定的索引（下标）开始
19 num:删除的个数
20 splice(index,num)从指定的下标（index）开始，删除num个
21 splice(index,num,item)从指定的下标（index）开始，将指定的num个的内容替换为item
22 注意：splice和delete删除不同点：
```

```
23 splice删除数组的值，不保留键
24 delete删除数组的值，保留键（此时对应的下标的值为undefined）
25
26 splice可以接收返回值，返回值为数组，返回被删除的内容
27 reverse()反转，颠倒数组元素的排序
28 *sort()对数组的元素进行排序（按照ASCII的码值进行排序）
29 sort(sortby)了解
30 sortby必须为一个函数
31
32 *indexOf()查找指定元素在数组中第一次出现的位置
33 如果该元素存在数组中，则返回该元素的下标，若不存在则返回-1
34 *lastIndexOf()查找指定元素在数组中最后一次出现的位置
35
36 注意：默认按照ASCII的码值从小到大排序
37 资料：ASCII的介绍 http://ascii.911cha.com/
38 拓展：0-9 编码为48-57 A-Z 编码为65-90 a-z 编码为97-122
39
40 ES6的数组函数：
41 map
42 filter
```

## 字符串函数

```
1 字符串中的length属性
2 语法：str.length
3
4 字符串查找指定下标的值（es6中使用）
5 str[index] index为指定的下标
6
7 字符串的查找
8 charAt(index)获取指定下标对应的值（index为下标，默认从0开始）
9 注意：若index的值超过0到字符串的长度，则返回的值为空字符串
10 charCodeAt(index)返回在指定的位置的字符的 Unicode 编码（ASCII码值）
11 fromCharCode(num)可接受一个指定的 Unicode 值，然后返回一个字符串
12 语法：String.fromCharCode(num) num为unicode编码
13
14 （indexOf lastIndexOf重点内容）
15 *indexOf()查找指定字符第一次出现的位置
16 *lastIndexOf()查找指定字符最后一次出现的位置
17
18 字符串的连接和拆分（重点）
```

```
19 *concat()用于连接两个或多个字符串
20 *split()将字符串按照指定的字符分割为数组（字符串转换为数组）
21
22 字符串中大小写转换
23 toLowerCase()将字符串转换为小写字母
24 toUpperCase()将字符串转换为大写字母
25
26 字符串的截取（重点）
27 *slice(start,end)截取指定下标的字符串
28 slice(start)（第一个参数必填）从指定下标（start）开始截取，截取到最后
29 slice(start,end)从指定下标（start）开始截取,截取到指定结束下标（end）之前
30 注意：
31 1、在截取字符串时，包含起始下标（start）,不包括结束下标（end）
32 2、截取的下标可以为负数，代表倒数第几个
33 3、slice截取为从左往右截取
34
35 *substr(start,num)从起始索引（下标start）提取字符串中指定数目（num）的字符
36 start:起始下标（索引）
37 下标可以为负数，代表从倒数第几个开始
38 num:截取的字符串个数（截取的个数不能为负数）
39
40 substring(start,stop)提取字符串中两个指定的索引号之间的字符
41 注意：substring()功能基本和slice()一致,只是两个下标可以随意放置
42
43 ES6中的字符串函数使用
44 *trim()去除字符串两侧的空格
45
46 正则中的字符串查找匹配（正则表达式一起讲解）
47 search()
48 match()
49 replace()
```

### 拓展作业：

- 1、封装一个函数，筛选出所有小于50的数值，以数组的方式返回  
如：[20,45,98,33,10,8,43,76,88,34,20,50,66]  
将返回[20,45,33,10,8,43,34,20]

```
1 var arr = [20,45,98,33,10,8,43,76,88,34,20,50,66];
2 function fun(arr){
3   var newArr = [];
```

```

4  for(var i=0;i<arr.length;i++){
5  // 判断数组中的值是否小于50
6  if(arr[i]<50){
7  // newArr[newArr.length] = arr[i];
8  newArr.push(arr[i])
9  }
10 }
11 return newArr;
12 }
13 var res = fun(arr)
14 console.log(res)

```

2、封装一个函数：将1-100中，所有数字中包含7，或者7的倍数的所有值以数组的方式返回

提示：

7的倍数： 7 14 21 28 35 42...       $i\%7==0$

包含7：

个数为7： 17 27 37 47 57...97       $i\%10==7$

十位为7： 70 71 72 73...79       $\text{parseInt}(i/10)==7$

```

1  function fun(){
2  var newArr = [];
3  for(var i=1;i<=100;i++){
4  if(i%7==0 || i%10==7 || parseInt(i/10)==7){
5  newArr.push(i);
6  }
7  }
8  return newArr;
9  }
10 var res = fun()
11 console.log(res)

```

## 数学函数

```

1  数学函数的方法
2  abs()绝对值
3  max()最大值 注意：参数不能直接接收数组
4  min()最小值
5  *floor()向下取整
6  ceil()向上取整（进一法取整）
7  round()四舍五入

```

```
8 *random()随机数
9 注意：默认产生0-1的随机数（可以取到0，但取不到1）
10 pow(x,y)次方根 返回x的y次幂
11 sqrt()开平方根
12
13 数学函数的属性
14 E 算数常量 代表e
15 PI 返回圆周率
```

课间练习：

封装一个函数，传入一个任意的数组，选取一个一等奖和两个二等奖  
(注意：中奖人不能重复)

## 时间函数

### date专门用于处理时间和日期

```
1 var date = new Date();
2 console.log(date)
3 // 查看对象原型内的方法
4 console.log(date.__proto__)
5
6 三种时间表现方式：
7 1、标准时间 toString()
8 2、国际时间 toLocaleString()
9 3、时间戳 valueOf()
10 在js中时间戳为14为的数字（单位：毫秒）
11 时间戳返回的是当前距离1970年1月1日（时间纪元）零时的毫秒数
```

GMT：格林威治时间（格林尼治或者天文学时间）

UTC：通用协调时

北京位于东八区 北京时间=UTC+8小时时差

## 定时器函数（重点）

```
1 设置定时器
2 setTimeout(callback, 时间)一次性定时器（只执行一次则结束）
3 callback代表回调函数 时间单位：毫秒
4 过多久时间执行一次函数体的内容
5 setTimeout(function(){
6 函数体
7 }, 1000)
```



```
8  setInterval(callback,时间)循环定时器（一直循环执行）
9  setInterval(function(){
10     函数体
11     },1000)
12
13  注意（方便理解）：
14  for和setInterval的区别：
15  1、for在不满足判断条件会停止，setInterval若不进行手动清除，则不会停止
16  2、for不能控制每一次循环的时间，setInterval可以去控制每一次循环的时间
17  清除定时器
18  clearTimeout()清除一次性定时器
19  clearInterval()清除循环定时器
```