

js第七天

数组方法

数组的应用

数组的应用非常的广泛、1。未来基本上都是操作数组；2.数组的方法一定要牢记并且能够灵活应用；

- 数组的写法：
 - `let arr = new Array()`,
 - `let arr = []` (性能高，建议用)
- 数组有length，既能读也能写
 - 读 `console.log(arr)`
 - 写 `arr.length = 3`
 - 清空数组，`arr.length = 0`
 - 如果增加数组的长度，就可以给数组内写内容 `arr[4]=5` 把5添加给数组的第五个值（简写方法：`arr.push(5)`;直接在数组最后添加一个5）
 - 数组中获取某个值通过下标去操作它[1,2,3] [number]
- **push () 方法**：给数组的**最后一位**后添加一个或多个值，一个与一个之间用逗号隔开，push也是一个函数，它的返回值是新数组的长度；

```
//arr.push(7,'');
```

- 函数调用里还有函数调用，从里往外执行，里边从左往右执行；
- **pop () 方法**：从数组的**最后一位**删除一个数据，括号内不用传参（传了也没用，就删除最后一个）；**返回值**是删掉的数据

```
//arr.pop()
```

- **unshift()方法**：在数组的首位前添加一个数据。括号内可以传参，添加多个数据。**返回值**是新数组的长度；

```
//arr.unshift(8,8,4,1)
```

- **shift方法**：在数组的首位删除一个数据。括号内也不用传参；返回值是删除的那个数据。

```
// arr.shift()
```

- **splice方法**：能够增删改查数组，根据参数的不同，结果就不同。
 - 语法：()

```
//arr.splice(2,0,9)
```

- 第一个参数是从数组的第几位起（选择数组的第几位起）从0开始；下标值；
- 第二个参数就是操作几个数据，（一般是删除或者替换）
 - 删除的话，返回值是删除的数据。数字代表删除的位数。
 - 替换的话，这个值删除，第三个值再添加即可。
- 第三个（后者以上）参数添加操作。添加多个。
 - 如果第二位是删除，则添加的数是替换删除的数据。如果第二位不删除，添加的数据则插入到该位置中，原数组后移；

注意，返回值主要看第二位，如果第二位为0，name返回值为**空数组**。也就是说，如果第二位不为零，返回值就是你删除的那几个数据（**返回值是数组**）

- **forEach方法**：专门用来循环数组的。只有数组能用forEach;返回值是undefined；
 - 语法：有两个参数arr.forEach()

```
// arr.forEach(function(item,i){  
//     this.item = item + 1;  
//     //console.log(this.item)  
  
// },arr)
```

- 第一个参数：函数function () {}
 - 函数内也有三个参数，（1.数组中的每项2.索引3.整个数组）
 - 第二个参数：改变this指向；写什么this就是什么（null和undefined指向的还是默认值window）
- **map方法**循环数组，它的返回值为新的数组；
 - 有一个参数：函数function () {}
 - 函数里有三个参数，（1.数组中的某项，2，索引，3，整个数组）加return

```
// let arr2 = arr.map(function(item,i){  
//     return item + i  
// })
```

- **join方法**以某个字符串为连接符，连接数组的每一项。最后返回一个字符串
 - **注意：如果不需要连接符，必须使用空字符串表示**

```
// arr2 = arr.join('+')
```

- **filter方法**过滤条件成立的值
 - 有一个参数，也是一个函数function () {}
 - 函数的参数也是item、i、all
 - return 条件成立的某项 (要过滤的条件，item<=28;保留数组中小于等于28的值)

```
// arr2 = arr.filter(function(item,i){  
//     return item < 5;  
// })
```

- **reverse**：翻转数组 (将数组倒序排列)

```
// var arr2 = arr.reverse(function(){  
  
// })
```

- **some方法**：返回一个布尔值；
 - 语法：arr.some(函数 (item ,i) return item===3判断条件)查看数组中某项数据是否满足某个条件，只要有一个符合条件就返回true，如果所有项都不成立，返回false；

```
// var arr2 = arr.some(function(item,i){  
//     return item === 9;  
// })
```

- **every方法**：判断数组中每一项是否都符合某个条件，全都符合返回true。只要有一个不符合就返回false；
 - 语法：arr.every(函数function (item , i) return 判断条件)

```
var arr2 = arr.every(function(item,i){  
    return item ===3  
})
```