

下面这些方法会改变数组自身的值：

1. **Array.prototype.pop()** 删除数组的最后一个元素，并返回这个元素

```
let arr = [1, 2, 3];
let value = arr.pop();
console.log(value); // 3
```

2. **Array.prototype.push()** 在数组的末尾增加一个或多个元素，并返回数组长度

```
let arr = [1, 2, 3];
arr.push(4);
console.log(arr); // [ 1, 2, 3, 4 ]
```

3. **Array.prototype.shift()** 删除数组的第一个元素，并返回这个元素

```
let arr = [1, 2, 3];
let value = arr.shift();
console.log(value); // 1
```

4. **Array.prototype.unshift()** 在数组的开头增加一个或多个元素，并返回数组长度

```
let arr = [1, 2, 3];
arr.unshift(0);
console.log(arr); // [ 0, 1, 2, 3 ]
```

5. **Array.prototype.sort()** 对数组进行排序，并返回当前数组。

```
let arr = [3, 1, 2];
arr.sort();
console.log(arr); // [ 1, 2, 3 ]
arr.sort(function (a, b) {
  if (a < b) return 1;
  if (a > b) return -1;
  if (a === b) return 0;
})
console.log(arr); // [ 3, 2, 1 ]
```

6. **Array.prototype.reverse()** 颠倒数组中元素的排列顺序。

```
let arr = [3, 2, 1];
arr.reverse();
console.log(arr); // [ 1, 2, 3 ]
```

7. **Array.prototype.splice()** 在任意位置给数组添加或删除任意个元素。

下面这些方法不会改变数组自身的值，只会返回一个新的数组或者其他值：

1. **Array.prototype.concat()** 返回当前数组和其他若干数组或者若干个费数组值组合而成的新数组

2. **Array.prototype.includes()** 判断当前数组是否包含某指定的值。

3. **Array.prototype.join()** 连接所有数组元素组成一个字符串

4. **Array.prototype.slice()** 抽取当前数组中的一段元素组合成一个新数组

5. **Array.prototype.toString()** 返回由数组元素组成的字符串

6. **Array.prototype.indexOf()** 返回数组中第一个与指定值相等的元素的索引，如果找不到返回-1

7. **Array.prototype.lastIndexOf()** 返回数组中最后一个与指定值相等的元素的索引，如果找不到返回-1

遍历方法：

1. **Array.prototype.forEach()** 为数组中每一个元素执行一次回调。

2. **Array.prototype.every()** 如果数组中每个元素都满足测试函数，则返回true，否则返回false；

3. **Array.prototype.some()** 如果数组中至少有一个元素满足测试函数，则返回true，否则返回false

4. **Array.prototype.filter()** 将所有在过滤函数中返回true的数组元素放假一个新数组中并返回。

5. **Array.prototype.find()** 找到第一个满足测试函数的元素并返回那个元素的值，如果找不到，返回undefined

6. **Array.prototype.map()** 返回一个由回调函数的返回值组成的新数组

7. **Array.prototype.reduce()** 从左到右为每个元素执行一次回调函数，并把上次回调函数的返回值放在一个暂存器中传给下次回调函数，并返回最后一次回调函数的返回值。