map

filter&map

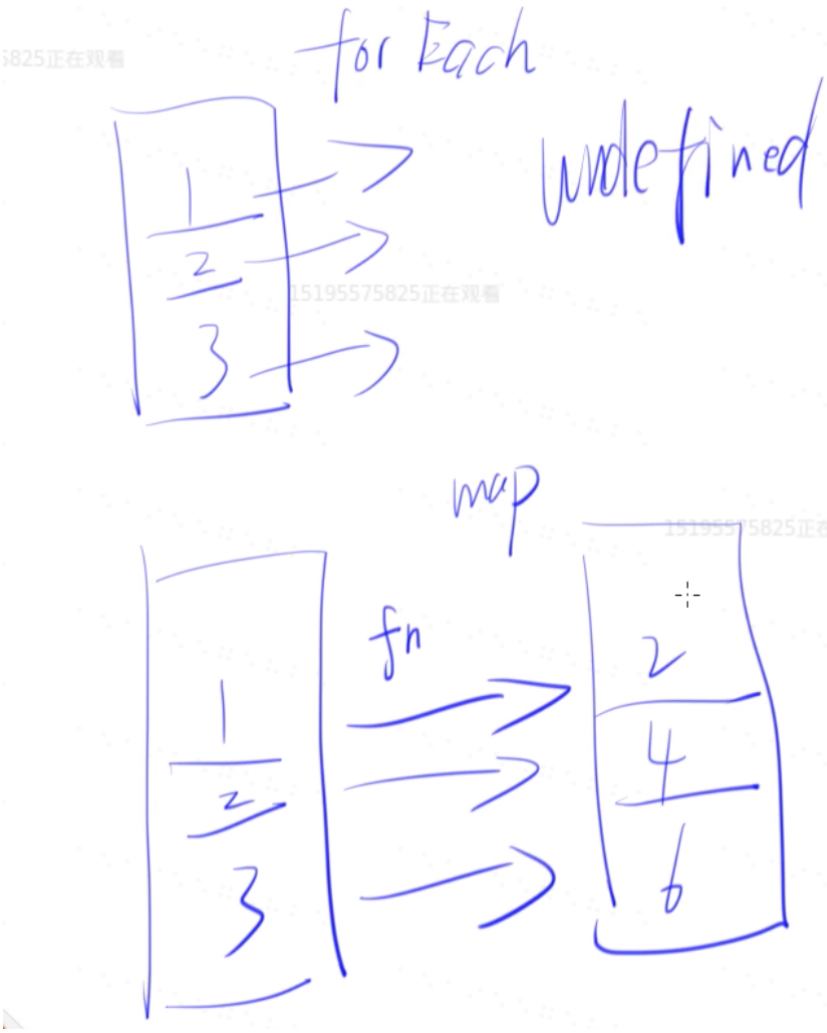
filter

reduce

sort&join&concat

sort

map



map有返回值,但是forEach没有返回值!

```
» a = [1,2,3]
← ▸ Array(3) [ 1, 2, 3 ]
» a.join('方方')
← "1方方2方方3"
» a.join(',')
← "1,2,3"
» a + ''
← "1,2,3"
» a.toString
← ▸ function toString()
» a.toString()
← "1,2,3"
» a.forEach(function(){})
← undefined
» a.map(function(value,key){
return value*2
})
← ▸ Array(3) [ 2, 4, 6 ]
»
```

map接受key和value,但是用不用是你的事;

```
» a =[1,2,3]
← ▸ Array(3) [ 1, 2, 3 ]
» a.map(value=>value*2)
← ▸ Array(3) [ 2, 4, 6 ]
» a.map(value=>value*3)
← ▸ Array(3) [ 3, 6, 9 ]
» a.map(value=>value*10)
← ▸ Array(3) [ 10, 20, 30 ]
» a.map(function(value,key){
return value*2
})
← ▸ Array(3) [ 2, 4, 6 ]
» a.map(function(value,key){
return key*2
})
← ▸ Array(3) [ 0, 2, 4 ]
» a.map(function(value,key){
return key
})
← ▸ Array(3) [ 0, 1, 2 ]
»
```

filter&map

```
>> var a = [1,2,3,4,5,6,7,8,9]
<- undefined
>> a.filter(function(value,key){
     return value%2==0
   })
<- ▸ Array(4) [ 2, 4, 6, 8 ]
>> a.filter(function(value,key){
     return value%2==0
   }).map(v=>v*2)
<- ▸ Array(4) [ 4, 8, 12, 16 ]
>> a.filter(function(value,key){
     return value%2==0
   }).map(v=>v**2)
<- ▸ Array(4) [ 4, 16, 36, 64 ]
>>
```

filter

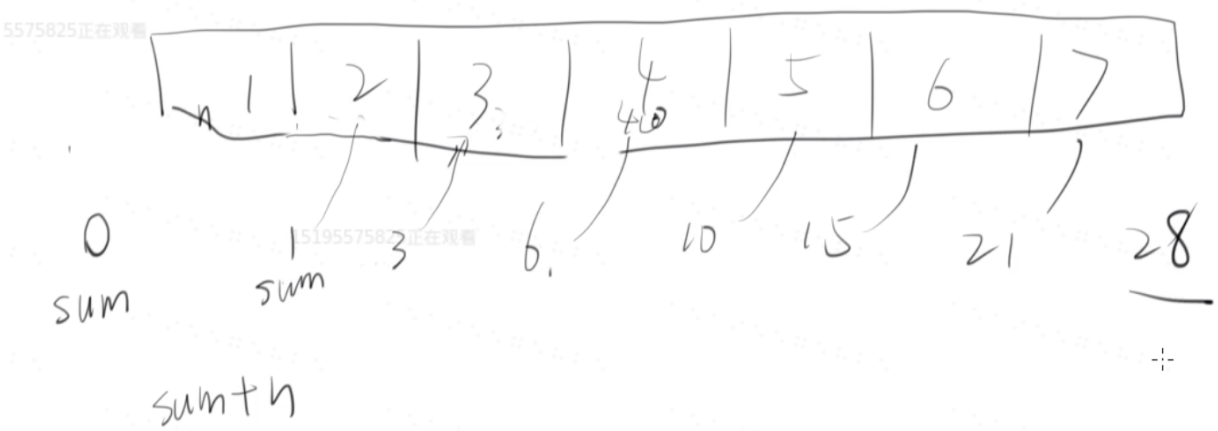
```
>> a=[1,2,3,4,5,6,7,8,9]
<- ▸ Array(9) [ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
>> a.filter(function(value,key){
   return value >= 5
   })
<- ▸ Array(5) [ 5, 6, 7, 8, 9 ]
>> a
<- ▸ Array(9) [ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
>>
```

filter+map:

```
» a=[1,2,3,4,5,6,7,8,9]
← ▸ Array(9) [ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
» a.filter(function(value,key){
  return value >= 5
  })
← ▸ Array(5) [ 5, 6, 7, 8, 9 ]
» a
← ▸ Array(9) [ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
» a.filter(function(value,key){
  return value >= 5
  }).map(function(value)){return value*2}
✖ SyntaxError: missing { before function body
» a.filter(function(value,key){
  return value >= 5
  }).map(function(value){
  return value*2
  })
← ▸ Array(5) [ 10, 12, 14, 16, 18 ]
»
```

reduce

```
» a=[1,2,3]
← ▸ Array(3) [ 1, 2, 3 ]
» var sum = 0
  for (let i =0;i<a.length;i++){
  sum +=a[i]
  }
← 6
» a.reduce((sum,n)=>sum + n,0)//这里的sum是上次结果,n是当前数字 ,这里是用箭头函数表示的, sum+n作为下次的sum
← 6
» a = [1,2,3,4,5,6,7,8,9]
← ▸ Array(9) [ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
» a.reduce((sum,n)=>sum + n,0)//这里的sum是上次结果,n是当前数字 ,这里是用箭头函数表示的, sum+n作为下次的sum
← 45
» a.reduce((sum,n)=>sum + n,0)//这里的sum是上次结果,n是当前数字 ,这里是用箭头函数表示的, sum+n作为下次的sum,0为sum的初值
← 45
»
```



```
» //filter可以用reduce表示
  a = [1,2,3,4,5,6,7,8,9,10]
← ▸ Array(10) [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
» a.reduce(function(arr,n){
  if(n%2===0){
  arr.push(n)
  }
  return arr
  },[])
← ▸ Array(5) [ 2, 4, 6, 8, 10 ]
»
```

```
» a.reduce(function(arr1,n){
  if(n%2===0){
  arr1.push(n)
  }
  return arr1
  },[])
← ▸ Array(5) [ 2, 4, 6, 8, 10 ]
»
```

```
» //map可以用reduce表示
  a=[1,2,3]
← ▸ Array(3) [ 1, 2, 3 ]
» a.reduce(function(arr,n){
  arr.push(n*2)
  return arr
  },[])
← ▸ Array(3) [ 2, 4, 6 ]
» //filter可以用reduce表示
  a = [1,2,3,4,5,6,7,8,9,10]
← ▸ Array(10) [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
» a.reduce(function(arr,n){
  if(n%2===0){
  arr.push(n)
  }
  return arr
  },[])
← ▸ Array(5) [ 2, 4, 6, 8, 10 ]
»
```

push方法:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi")

输出:
Banana,Orange,Apple,Mango,Kiwi


```

sort&join&concat

```
» var a =[1,2,3]
← undefined
» a.join()
← "1,2,3"
» var b =[4,5,6]
← undefined
» a.concat(b)
← ▸ Array(6) [ 1, 2, 3, 4, 5, 6 ]
» c = a.concat(b)
← ▸ Array(6) [ 1, 2, 3, 4, 5, 6 ]
» c
← ▸ Array(6) [ 1, 2, 3, 4, 5, 6 ]
» c.sort(function (x,y){return x-y})
← ▸ Array(6) [ 1, 2, 3, 4, 5, 6 ]
» c.sort(function (x,y){return y-x})
← ▸ Array(6) [ 6, 5, 4, 3, 2, 1 ]
»
```

有hash的一个例子！

```
» a=['马云','马化腾','李彦宏']
← ▸ Array(3) [ "马云", "马化腾", "李彦宏" ]
» hash = {
  '马云':167,
  '马化腾':376,
  '李彦宏':228
  }
← ▸ Object { "马云": 167, "马化腾": 376, "李彦宏": 228 }
» a.sort(function(x,y){
  return  hash[y]-hash[x]
  })
← ▸ Array(3) [ "马化腾", "李彦宏", "马云" ]
»
```

另外一个例子:

```
» var students = ['小明','小红','小花']
← undefined
» var scores = { 小明: 59, 小红: 99, 小花: 80 }
← undefined
» students.sort(function(x,y){
  return scores[y]-scores[x]
  })
← ▸ Array(3) [ "小红", "小花", "小明" ]
»
```

sort

只有sort是改变自己的，其他函数都没有改变自己!

```
» b =[5,6,2]
← ▸ Array(3) [ 5, 6, 2 ]
» b.sort()
← ▸ Array(3) [ 2, 5, 6 ]
»
```