

[←](#) 2-11 数组 - Array

数组 - Array

Array

在 ES6 中新增了很多实用的原生 API，方便开发者对 Array 的操控性更强，如 for...of、from、of、fill、find、findIndex 等。

for...of

看这个名字就知道这个语法是用来遍历数组的，我们简单回忆下原始的数组遍历方法是怎样的？

```
for (var i = 0; i < array.length; i++) {  
  console.log(array[i]);  
}
```

后来语法有所升级，到 ES5 遍历数组的 API 多了起来，其中有 forEach、every、filter 等，同样的功能可以用 forEach 或者 every 来实现。

```
array.forEach(function(i){  
  console.log(i);  
})
```

这个语法看起来要简洁很多，不需要通过索引去访问数组项，然而它的缺点也是很明显，不支持 break、continue 等。

```
[1,2,3,4,5].forEach(function(i){  
  if(i===2){  
    return;  
  }else{  
    console.log(i)  
  }  
})
```

这段代码的“本意”是从第一个元素开始遍历，遇到数组项 2 之后就结束遍历，不然打印出所遍历过的数值项。可是，事实让你大跌眼镜，因为它输出是 1,3,4,5。

[!DANGER]

forEach 的代码块中不能使用 break、continue，它会抛出异常。

```
[1,2,3,4,5].every(function(i){  
  if(i===2){  
    return false;  
  }else{  
    console.log(i)  
    return true  
  }  
})
```

同样完成刚才的目标，使用 every 遍历就可以做到 break 那样的效果，简单的说 return false 等同于 break，return true 等同于 continue。如果不是默认是 return false。

[!DANGER]

every 的代码块中不能使用 break、continue，它会抛出异常。

有的同学会说，还有 for...in 可以遍历数组。

```
for (var index in array) {  
  console.log(array[index]);  
}
```