

# JS中的4种循环

## 普通循环和forEach 循环

JavaScript诞生已经有20多年了，我们一直使用的用来循环一个数组的方法是这样的：

JavaScript 代码：

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

自从JavaScript 5 起，我们开始可以使用内置的 `forEach` 方法：

JavaScript 代码：

```
myArray.forEach(function (value) { console.log(value);});
```

写法简单了许多，但也有短处：你不能中断循环(使用 `break` 语句或使用 `return` 语句)。

## for...in 循环

JavaScript里还有一种循环方法：`for...in`。

`for...in` 循环实际是为循环可枚举(enumerable)对象而设计的：

JavaScript 代码：

```
var obj = {a:1, b:2, c:3}; for (var prop in obj) { console.log("obj." + prop + " = " + obj[prop]);} // 输出:// "obj.a = 1"// "obj.b = 2"// "obj.c = 3"
```

你也可以用它来循环一个数组：

JavaScript 代码：

```
for (var index in myArray) { // 不推荐这样 console.log(myArray[index]);}
```

不推荐用 `for...in` 来循环一个数组，因为，不像对象，数组的 `index` 跟普通的对象属性不一样，是重要的数值序列指标。

总之，`for...in` 是用来循环带有字符串key的对象的方法。

## for...of 循环

JavaScript6里引入了一种新的循环方法，它就是 `for...of` 循环，它既比传统的 `for` 循环简洁，同时弥补了 `forEach` 和 `for-in` 循环的短板。

我们看一下它的 `for...of` 的语法：

JavaScript 代码:

```
for (var value of myArray) { console.log(value);}
```

`for...of` 的语法看起来跟 `for...in` 很相似, 但它的功能却丰富的多, 它能循环很多东西。

`for...of` 循环使用例子:

## 循环一个数组(Array):

JavaScript 代码:

```
let iterable = [10, 20, 30]; for (let value of iterable) { console.log(value);} // 10// 20// 30
```

我们可以使用 `const` 来替代 `let`, 这样它就变成了在循环里的不可修改的静态变量。

JavaScript 代码:

```
let iterable = [10, 20, 30]; for (const value of iterable) { console.log(value);} // 10// 20// 30
```

## 循环一个字符串:

JavaScript 代码:

```
let iterable = "boo"; for (let value of iterable) { console.log(value);} // "b"// "o"// "o"
```

## 循环一个类型化的数组(TypedArray):

JavaScript 代码:

```
let iterable = new Uint8Array([0x00, 0xff]); for (let value of iterable) { console.log(value);} // 0// 255
```

## 循环一个Map:

JavaScript 代码:

```
let iterable = new Map([["a", 1], ["b", 2], ["c", 3]]); for (let [key, value] of iterable) { console.log(value);} // 1// 2// 3 for (let entry of iterable) { console.log(entry);} // [a, 1]// [b, 2]// [c, 3]
```

## 循环一个 Set:

JavaScript 代码:

```
let iterable = new Set([1, 1, 2, 2, 3, 3]); for (let value of iterable) {  
  console.log(value);} // 1 // 2 // 3
```

## 循环一个 DOM 集合

循环一个 DOM 集合，比如 NodeList，之前我们讨论过如何循环一个 NodeList，现在方便了，可以直接使用

`for...of` 循环：

JavaScript 代码：

```
// Note: This will only work in platforms that have// implemented  
NodeList.prototype[Symbol.iterator]let articleParagraphs =  
document.querySelectorAll("article > p"); for (let paragraph of articleParagraphs) {  
  paragraph.classList.add("read");}
```

## 循环一个拥有可枚举属性的对象

`for...of` 循环并不能直接使用在普通的对象上，但如果我们按对象所拥有的属性进行循环，可使用内置的 `Object.keys()` 方法：

JavaScript 代码：

```
for (var key of Object.keys(someObject)) { console.log(key + ": " + someObject[key]);}
```

## 循环一个生成器(generators)

我们可循环一个生成器(generators)：

JavaScript 代码：

```
function* fibonacci() { // a generator function let [prev, curr] = [0, 1]; while  
(true) { [prev, curr] = [curr, prev + curr]; yield curr; }} for (let n of  
fibonacci()) { console.log(n); // truncate the sequence at 1000 if (n >= 1000) {  
break; }}
```

更多关于 `for...of` 循环的示例请查看 [理解 JavaScript 中的 for...of 循环](#)