



链滴

# JavaScript - for...in、for...of 和 forEach 的不同点

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1594466738196>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2020-07-10

## 描述

在 JavaScript 中，对比 for...in、for...of 和 forEach 之间的不同点。

## for...in

### 说明

for...in 会迭代对象中所有的可枚举属性，当然也包括继承的可枚举属性。他可以对数组、字符串或普通对象进行迭代，但不能对 Map 或 Set 对象进行迭代。

### 代码

```
for (let prop in ['a', 'b', 'c'])
  console.log(prop); // 输出数组的索引: 0, 1, 2

for (let prop in 'str')
  console.log(prop); // 输出字符串的索引: 0, 1, 2

for (let prop in {a: 1, b: 2, c: 3})
  console.log(prop); // 输出对象的属性名称: a, b, c

for (let prop in new Set(['a', 'b', 'a', 'd']))
  console.log(prop); // 没有可枚举的属性, 输出为空
```

## for...of

### 说明

for...of 会对可以迭代的对象进行迭代，但他迭代的是对象的值，而不是对象的属性。他可以对数组、字符串、Map 或 Set 对象进行迭代，但不能对普通对象进行迭代。

### 代码

```
for (let val of ['a', 'b', 'c'])
  console.log(val); // 输出数组的值: a, b, c

for (let val of 'str')
  console.log(val); // 输出字符串中的字符: s, t, r

for (let val of {a: 1, b: 2, c: 3})
  console.log(prop); // Uncaught TypeError: {(intermediate value)(intermediate value)(intermediate value)} is not iterable

for (let val of new Set(['a', 'b', 'a', 'd']))
  console.log(val); // 输出 Set 值: a, b, d
```

# forEach()

## 说明

`forEach()` 是 `Array` 原型上的一个方法，他允许你迭代数组中的每一个元素。`forEach()` 虽然只能迭代组，但他在迭代的时候能访问每一个元素的值和索引。

## 代码

```
['a', 'b', 'c'].forEach(  
  val => console.log(val); // 输出数组的值: a, b, c  
);
```

```
['a', 'b', 'c'].forEach(  
  (val, i) => console.log(i); // 输出数组的索引: 0, 1, 2  
);
```

## 返回总目录

[每天 30 秒系列之 JavaScript 代码](#)