

[←](#) 2-63 Iterator

Iterator



Iterable



处理集合中的每个项是很常见的操作。JavaScript 提供了许多迭代集合的方法，从简单的for循环到map()和filter()。迭代器和生成器将迭代的概念直接带入核心语言，并提供了一种机制来自定义for...of循环的行为。



如果对 MDN 这个描述理解不是很到位的话，可以看下接下来这个小示例：

```
let authors = {
  allAuthors: {
    fiction: [
      'Agatha Christie',
      'J. K. Rowling',
      'Dr. Seuss'
    ],
    scienceFiction: [
      'Neal Stephenson',
      'Arthur Clarke',
      'Isaac Asimov',
      'Robert Heinlein'
    ],
    fantasy: [
      'J. R. R. Tolkien',
      'J. K. Rowling',
      'Terry Pratchett'
    ]
  }
}
```

这个数据结构是汇总了所有作者，每个作者按创作性质进行了分类。如果我们想获取所有作者的名单，该怎么做呢？

```
for (let author of authors) {
  console.log(author)
}
```

[!DANGER]

你发现这个遍历遇到了报错：Uncaught TypeError: authors is not iterable

有的同学会说可以这样做：

```
for (let key in authors) {
  let r = []
  for (let k in authors[key]) {
    r = r.concat(authors[key][k])
  }
  console.log(r)
  // ["Agatha Christie", "J. K. Rowling", "Dr. Seuss", "Neal Stephenson", "Arthur Clarke", "Isaac Asimov", "Robert Heinlein"]
}
```

这个做法确实可以，实际上也是手动实现的遍历加数据合并，这其实不算是遍历。这个小节就是讲述如何给这种自定义的数据结构进行遍历。

Iterator

Iterator 就是 ES6 中用来实现自定义遍历的接口，按照上述的示例，我们来实现下这个接口：