博学谷
www.boxuegu.com

第十一章 Iterator和for...of循环

一样的在线教育，不一样的教学品质

目录 Contents

1.Iterator基本使用

Iterator，被称为遍历器/跌代器，它是一种接口；任何数据结构，只要部署了Iterator接口，即可完成遍历操作。

Iterator遍历的过程：

1）创建一个指针对象，指向当前数据结构的起始位置。也就是说，遍历器对象本质上就是一个指针对象。

2）第一次调用指针对象的next方法，可以将指针指向数据结构的第一个成员。

3）第二次调用指针对象的next方法，指针就指向数据结构的第二个成员。

4）不断调用指针对象的next方法，直到它指向数据结构的结束位置。

说明：

每一次调用next方法，都会返回数据结构的当前成员的信息。具体来说，就是返回一个包含value和done两个属性的对象。其中，value属性是当前成员的值，done属性是一个布尔值，表示遍历是否结束。

2.数据结构的默认Iterator接口

ES6 规定，默认的 Iterator 接口部署在数据结构的Symbol.iterator属性；或者说，一个数据结构只要具有Symbol.iterator属性，就可认为是“可遍历的”（iterable）。

Symbol.iterator属性本身是一个函数，即当前数据结构默认的遍历器生成函数。执行这个函数，返回一个遍历器 。

原生具备Iterator接口的数据结构：**数组、某些类似数组的对象、Set和Map结构**。

如果是其他的数据结构，例如对象，想具有遍历器接口，就需要自己在Symbol.iterator属性部署。

部署Symbol.iterator属性的目的：

可以在对象上使用后面讲到 for...of方式来对对象进行循环遍历。

即想使用for...of循环方式，对应的数据结构必须部署Symbol.iterator属性。

3.调用Iterator接口场合

默认调用Iterator接口（即Symbol.iterator方法）的场合：

- **解构赋值：**

  对数组和Set解构进行解构赋值时，会默认调用Symbol.iterator方法；

- **扩展运算符**

  扩展运算符(...)也会调用默认的Symbol.iterator方法；

目录 Contents

1.for...of循环使用范围

一个数据结构只要部署了Symbol.iterator属性，就被视为具有 iterator 接口；可用for...of循环遍历它的成员。

即**for...of循环内部调用的是数据结构的Symbol.iterator方法**。

for...of循环可以使用的范围：

- **数组**
- **Set 和 Map 结构**
- **某些类似数组的对象（比如arguments对象、DOM NodeList 对象）**
- **Generator 对象**
- **字符串。**

2.与其他遍历语法的比较

- for...of循环与for...in语法一样的简洁；
- 不同于forEach方法， for...of可以与break，return等配合使用；
- 提供遍历所有数据结构的统一操作接口。

一样的在线教育，不一样的教学品质