#### ES2019提案预览

##### 之迭代器

摘要

JavaScript语言特性——迭代器

@link <https://github.com/tc39/proposal-iterator-helpers>

JavaScript新特性——Object与Array??相互转换

@link <https://github.com/tc39/proposal-object-map>

正文

什么是迭代器？

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。

在 JavaScript 中，迭代器是一个对象，它定义一个序列，并在终止时可能返回一个返回值。 更具体地说，迭代器是通过使用 next() 方法实现 Iterator protocol 的任何一个对象，该方法返回具有两个属性的对象： value，这是序列中的 next 值；和 done ，如果已经迭代到序列中的最后一个值，则它为 true 。如果 value 和 done 一起存在，则它是迭代器的返回值。

JS中有哪些iterator helper？

forEach迭代器

every迭代器

some迭代器

reduce迭代器

map迭代器

fiter迭代器

怎样使用iterator helper？

示例如下：

binding: {

data: this.kpis.map(kpi=>{

return {

kpi: kpi.name,

location: this.selectedPo.shortName

}

}),

}

这个提案讲了什么？

这个提案建议将Python、Rust中的迭代器引进JS中，如flatMap、findMap、filterMap等。

Making mapping over Objects more concise

该特性建议JavaScript支持Object与Array类型的相互转换。这样做的优势在于：Object对象可以充分使用丰富的iterator helper函数，示例如下：

// Utility method to get an iterator on an Object

Iterator.from(obj)

// Standard `map` function that operates on iterables

.map(([key, value]) => [transform(key), transform(value)])

// Collect the iterable data back into an Object, specifying key and value selectors

.toObject(([key]) => key, ([, value]) => value);

**Note** 这里改为JSON类型可能更为合适.

□