



Rust 编程语言入门



杨旭，微软 MVP
Rust、Go 开发者

13.6 迭代器（2）

- 消耗/产生迭代器

消耗迭代器的方法

- 在标准库中，Iterator trait 有一些带默认实现的方法
- 其中有一些方法会调用 next 方法
 - 实现 Iterator trait 时必须实现 next 方法的原因之一
- 调用 next 的方法叫做“消耗型适配器”
 - 因为调用它们会把迭代器消耗尽
- 例如：sum 方法（就会耗尽迭代器）
 - 取得迭代器的所有权
 - 通过反复调用 next，遍历所有元素
 - 每次迭代，把当前元素添加到一个总和里，迭代结束，返回总和
- （例子）

产生其它迭代器的方法

- 定义在 Iterator trait 上的另外一些方法叫做“迭代器适配器”
 - 把迭代器转换为不同种类的迭代器
- 可以通过链式调用使用多个迭代器适配器来执行复杂的操作，这种调用可读性较高。
- 例如：map
 - 接收一个闭包，闭包作用于每个元素
 - 产生一个新的迭代器
- （例子）
- collect 方法：消耗型适配器，把结果收集到一个集合类型中。

再见

