



Rust 编程语言入门



杨旭，微软 MVP
Rust、Go 开发者

13.4 闭包（4）

- 使用闭包捕获环境

闭包可以捕获他们所在的环境

- 闭包可以访问定义它的作用域内的变量，而普通函数则不能。
- 会产生内存开销。

闭包从所在环境捕获值的方式

- 与函数获得参数的三种方式一样：
 1. 取得所有权：FnOnce
 2. 可变借用：FnMut
 3. 不可变借用：Fn
- 创建闭包时，通过闭包对环境值的使用，Rust 推断出具体使用哪个 trait：
 - 所有的闭包都实现了 FnOnce
 - 没有移动捕获变量的实现了 FnMut
 - 无需可变访问捕获变量的闭包实现了 Fn

move 关键字

- 在参数列表前使用 move 关键字，可以强制闭包取得它所使用的环境值的所有权
 - 当将闭包传递给新线程以移动数据使其归新线程所有时，此技术最为有用。
- （例子）

最佳实践

- 当指定 Fn trait bound 之一时，首先用 Fn，基于闭包体里的情况，如果需要 FnOnce 或 FnMut，编译器会再告诉你。

再见

