



Rust 编程语言入门



Microsoft®
Most Valuable
Professional

杨旭，微软MVP
Rust、Go、C#开发者

3.5 控制流

if 表达式

- if 表达式允许您根据条件来执行不同的代码分支
 - 这个条件必须是 bool 类型
- if 表达式中，与条件相关联的代码块就叫做分支（arm）
- 可选的，在后边可以加上一个 else 表达式
- （例子）

使用 else if 处理多重条件

- （例子）
- 但如果使用了多于一个 else if，那么最好使用 match 来重构代码

在 let 语句中使用 if

- 因为 if 是一个表达式，所以可以将它放在 let 语句中等号的右边（例子）

Rust 的循环

- Rust 提供了 3 种循环：loop，while 和 for

loop 循环

- loop 关键字告诉 Rust 反复的执行一块代码，直到你喊停
- （例子）
- 可以在 loop 循环中使用 break 关键字来告诉程序何时停止循环
- （例子）

while 条件循环

- 另外一种常见的循环模式是每次执行循环体之前都判断一次条件。
- while 条件循环为这种模式而生
- （例子）

使用 for 循环遍历集合

- 可以使用 while 或 loop 来遍历集合，但是易错且低效（例子）
- 使用 for 循环更简洁紧凑，它可以针对集合中的每个元素来执行一些代码
- （例子）
- 由于 for 循环的安全、简洁性，所以它在 Rust 里用的最多

Range

- 标准库提供
- 指定一个开始数字和一个结束数字，Range 可以生成它们之间的数字（不含结束）
- rev 方法可以反转 Range
- （例子）

再见

