



# Rust 编程语言入门



Microsoft®  
Most Valuable  
Professional

杨旭，微软MVP  
Rust、Go、C#开发者

# 控制测试如何运行

- 改变 cargo test 的行为：添加命令行参数
- 默认行为：
  - 并行运行
  - 所有测试
  - 捕获（不显示）所有输出，使读取与测试结果相关的输出更容易。
- 命令行参数：
  - 针对 cargo test 的参数：紧跟 cargo test 后
  - 针对 测试可执行程序：放在 -- 之后
- cargo test --help
- cargo test -- --help

## 11.6 并行/连续运行测试

# 并行运行测试

- 运行多个测试：默认使用多个线程并行运行。
  - 运行快
- 确保测试之间：
  - 不会互相依赖
  - 不依赖于某个共享状态（环境、工作目录、环境变量等等）

## --test-threads 参数

- 传递给 二进制文件
- 不想以并行方式运行测试，或想对线程数进行细粒度控制
- 可以使用 --test-threads 参数，后边跟着线程的数量
- 例如：`cargo test -- --test-threads=1`

## 显式函数输出

- 默认，如测试通过，Rust 的 test 库会捕获所有打印到标准输出的内容。
- 例如，如果被测试代码中用到了 `println!`:
  - 如果测试通过：不会在终端看到 `println!` 打印的内容
  - 如果测试失败：会看到 `println!` 打印的内容 和 失败信息
- （例子）
- 如果想在成功的测试中看到打印的内容： `--show-output`
- （例子）

再见

