

Rust | Course Introduction

■ Why Rust?

- ◆ High-level language features without performance penalties
- ◆ Program behaviors can be enforced at compile time
 - Enhanced program reliability
- ◆ Built-in dependency management, similar to *npm*
- ◆ Quickly growing ecosystem of libraries
- ◆ Friendly & welcoming developer community

■ Technical Rust Goodies

- ◆ First-class multithreading
 - Compiler error to improperly access shared data
- ◆ Type system:
 - Can uncover bugs at compile time
 - Makes refactoring simple
 - Reduces the number of tests needed
- ◆ Module system makes code separation simple
- ◆ Adding a dependency is 1 line in a config file
- ◆ Tooling:
 - Generate docs, lint code, auto format

■ About Course

- ◆ Rust is different by design
- ◆ “Clean slate” approach
 - Important because of subtle language differences between Rust and others
 - ▶ Learn Rust versus learn to fight the compiler
 - Helps facilitate learning Rust
- ◆ Plenty of coding exercises for becoming familiar with Rust code

Enjoy!