LESSON 12



Rust

Build a CLI Tool

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUST PROGRAMMING

LESSON 12

Project: A Command-Line Tool

Crates, clap, serde, modules, tests, release builds and next steps

Crates and Cargo.toml

Cargo.toml

```toml
[package]
name = "grades"
version = "0.1.0"
edition = "2021"

[dependencies]
anyhow = "1"
clap = { version = "4", features = ["derive"] }
serde = { version = "1", features = ["derive"] }
serde_json = "1"
```

- Libraries in Rust are called crates, published on crates dot io. cargo add puts one in …
- We will use clap for command-line arguments, serde and serde_json for JSON, and anyhow for …

01

Arguments with clap

src/main.rs

```rust
use clap::Parser;

/// Summarize student grades from a JSON file
#[derive(Parser)]
struct Args {
    /// Path to the JSON file
    file: std::path::PathBuf,
    /// Minimum passing score
    #[arg(long, default_value_t = 50.0)]
    pass: f64,
}
```

- With clap's derive, a struct describes your command-line options. clap parses them, …
- Doc comments become the help descriptions.

02

JSON with serde

src/main.rs

```rust
#[derive(serde::Deserialize)]
struct Student { name: String, city: String, score: f64 }

fn main() -> anyhow::Result<()> {
    let args = Args::parse();
    let text = std::fs::read_to_string(&args.file)?;
    let students: Vec<Student> = serde_json::from_str(&text)?;
    let passed = students.iter().filter(|s| s.score >= args.pass).count();
    println!("{passed}/{} passed", students.len());
    Ok(())
}
```

- serde turns Rust structs into JSON and back. Add derive Deserialize, and serde_json …
- The main function reads the file, parses it, and prints a report, with errors handled by …

03

Tests and Release Builds

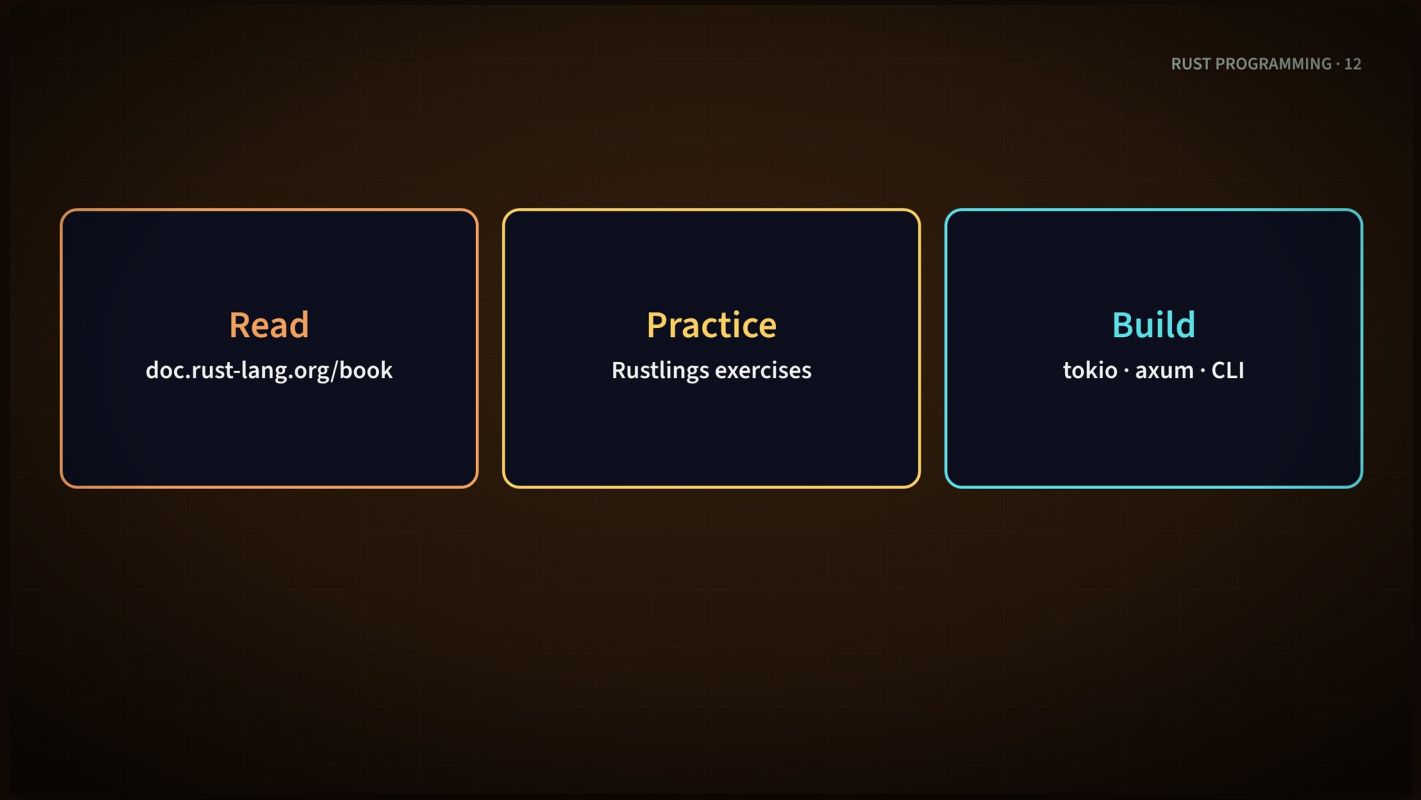
terminal

```
$ cargo test
running 3 tests
test tests::average_works ... ok
test result: ok. 3 passed; 0 failed
$ cargo build --release
$ ./target/release/grades students.json --pass 60
7/9 passed
```

- Write tests in a module marked cfg test, and functions marked with the test attribute. …
- cargo build dash dash release creates a fast, optimized binary in target slash release.

04

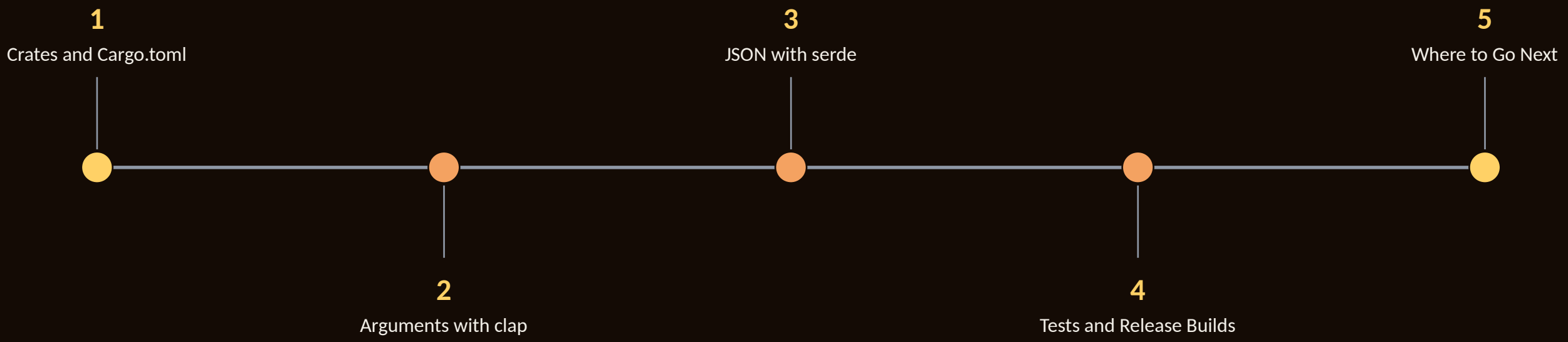
Where to Go Next



- Read — doc.rust-lang.org/book
- Practice — Rustlings exercises
- Build — tokio · axum · CLI

05

Lesson Map


1
Crates and Cargo.toml
2
Arguments with clap
3
JSON with serde
4
Tests and Release Builds
5
Where to Go Next

Sources and Further Reading

Cargo.toml

```
[package]
name = "grades"
version = "0.1.0"
edition = "2021"

[dependencies]
anyhow = "1"
clap = { version = "4", features = ["derive"] }
serde = { version = "1", features = ["derive"] }
serde_json = "1"
```

RUST PROGRAMMING · 12

src/main.rs

```
#[derive(serde::Deserialize)]
struct Student { name: String, city: String, score: f64 }

fn main() -> anyhow::Result<()> {
    let args = Args::parse();
    let text = std::fs::read_to_string(&args.file)?;
    let students: Vec<Student> = serde_json::from_str(&text)?;
    let passed = students.iter().filter(|s| s.score >= args.pass).count();
    println!("{passed}/{} passed", students.len());
    Ok(())
}
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari