درس ۱۲

Rust

ساختن ابزار CLI

```rust
#[derive(Parser)]
struct Args { file: PathBuf }
let a = Args::parse();
```

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Rust

درس ۱۲

پروژه: یک ابزار خط فرمان



crate ها، clap، serde، module ها، آزمایش‌ها، ساخت release و قدم‌های بعدی

crate ها و Cargo.toml

کتابخانه‌ها در Rust، crate نامیده می‌شوند، که در crates.io منتشر می‌شوند. cargo add یکی را ...

از clap برای آرگومان‌های خط فرمان، از serde و serde_json برای JSON، و از anyhow برای خطاها ...

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"
```

01

آرگومان‌ها با clap

با derive در clap، یک struct گزینه‌های خط فرمان شما را توصیف می‌کند. clap آن‌ها را تحلیل و ...

doc comment ها توضیحات کمک می‌شوند.

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,
}
```

02

JSON با serde

serde، struct های Rust را به JSON و برعکس تبدیل می‌کند. derive Deserialize را اضافه کنید، و ...

تابع main فایل را می‌خواند، تحلیل می‌کند، و یک گزارش چاپ می‌کند، با خطاهایی که علامت سوال ...

src/main.rs

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

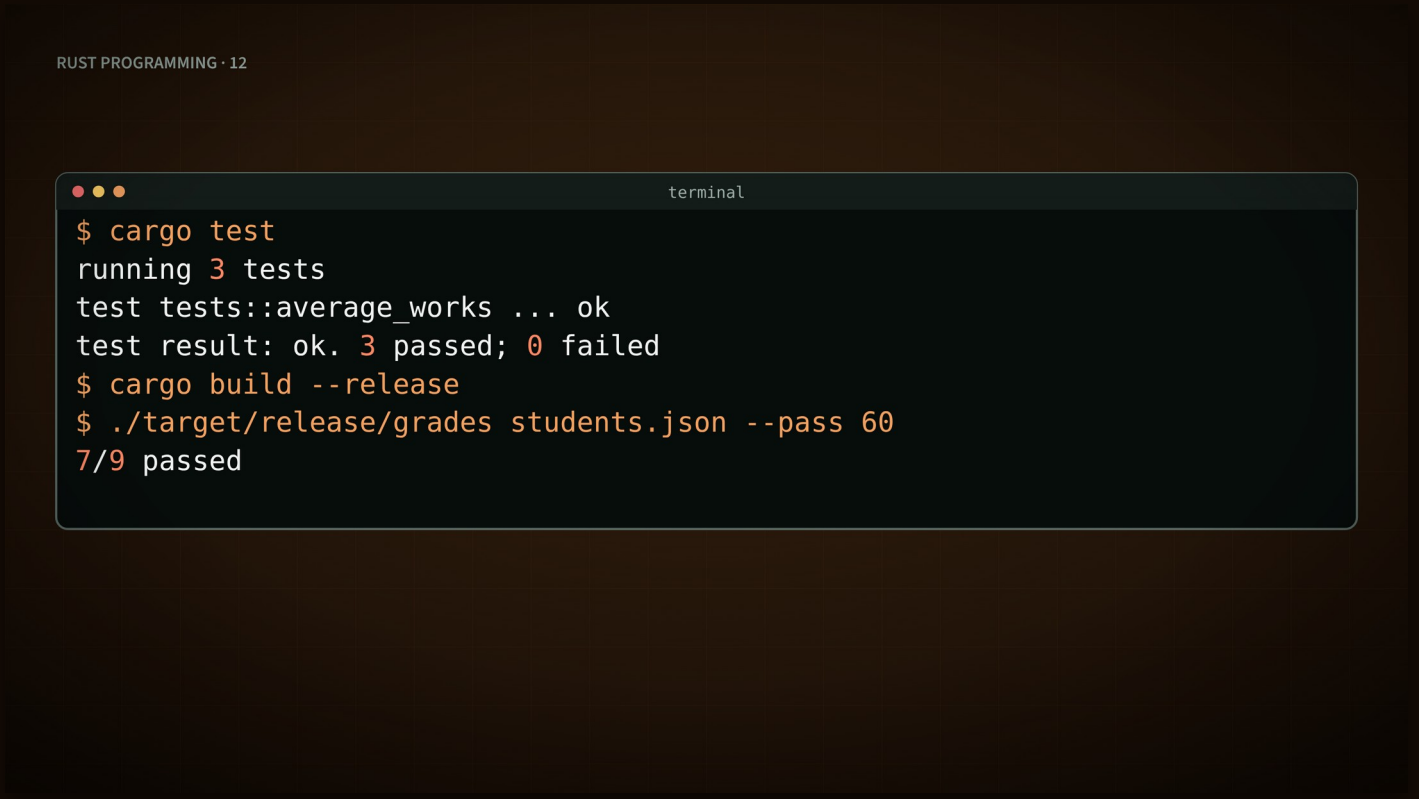
fn main() -> anyhow::Result<()> {
    let args = Args::parse();
    let text = std::fs::read_to_string(&args.file)?;
    let students: Vec<Student> = serde_json::from_str(&text)?;
    let passed = students.iter().filter(|s| s.score >= args.pass).count();
    println!("{passed}/{} passed", students.len());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

03

آزمایش‌ها و ساخت release

آزمایش‌ها را در یک module با نشانی cfg(test)، و تابع‌هایی با attribute test بنویسید. cargo ...

cargo build --release یک فایل اجرایی سریع و بهینه در target/release می‌سازد.



04

قدم‌های بعدی

بخوانید — doc.rust-lang.org/book

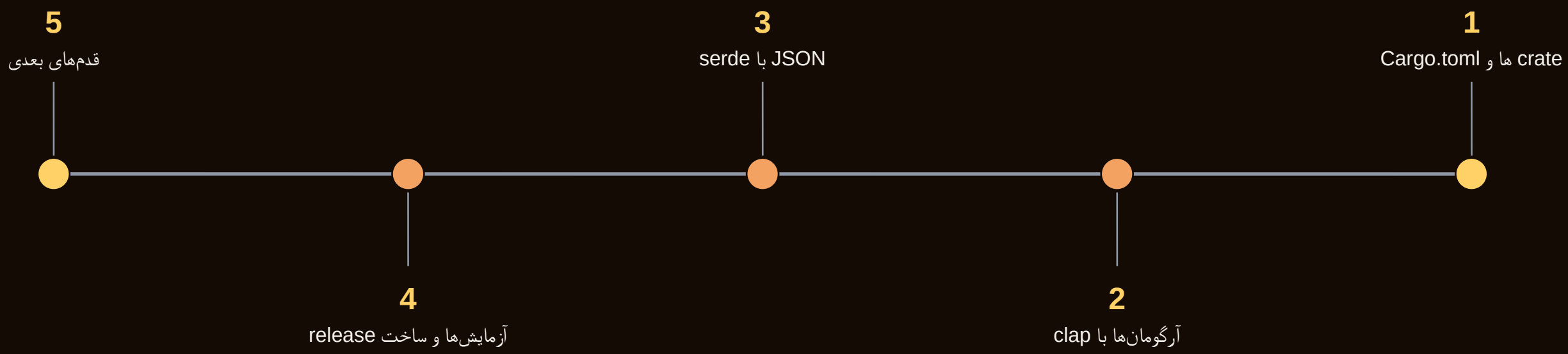
تمرین — تمرین‌های Rustlings

بسازید — tokio · axum · CLI



05

نقشهٔ درس


1
crate ها و Cargo.toml
2
آرگومان‌ها با clap
3
JSON با serde
4
آزمایش‌ها و ساخت release
5
قدم‌های بعدی

منابع و مطالعهٔ بیشتر

RUST PROGRAMMING · 12

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"
```

RUST PROGRAMMING · 12

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(())
}
```

تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity