LESSON 7

main.rs

```
match opt {
  Some(v) => v,
  None => 0,
}
```

Rust

enum · Option · match

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUST PROGRAMMING

LESSON 7

Enums, Option and match

Enums with data, Option instead of null, match, if let and let else

Enums with Data

main.rs

```rust
enum Shape {
    Circle { r: f64 },
    Rect { w: f64, h: f64 },
    Triangle(f64, f64, f64),
}

enum Message {
    Quit,
    Move { x: i32, y: i32 },
    Write(String),
}
```

- A Rust enum lists possible variants, and each variant can carry its own data. One type can …
- This makes impossible states impossible to represent.

01

Option: No More null

main.rs

```rust
fn find_student(id: u32) -> Option<String> {
    if id == 1 { Some("Mariam".to_string()) } else { None }
}

let name = find_student(7).unwrap_or("unknown".to_string());
let len = find_student(1).map(|n| n.len());   // Some(6)
```

- Rust has no null. A value that might be missing has the type Option: either Some with a …
- The compiler forces you to handle None, so null pointer crashes simply cannot happen.

02

match

main.rs

```rust
fn area(s: &Shape) -> f64 {
    match s {
        Shape::Circle { r } => 3.14159 * r * r,
        Shape::Rect { w, h } => w * h,
        Shape::Triangle(a, b, c) => heron(*a, *b, *c),
    }
}
let label = match score {
    90..=100 => "A", 75..=89 => "B", _ => "C",
};
```

- match compares a value against patterns and runs the first arm that fits. It must be …
- Patterns can pull data out of enums, and the underscore matches anything else.

03

if let and let else

main.rs

```rust
if let Some(name) = find_student(1) {
    println!("Found {name}");
}

fn parse_age(s: &str) -> u32 {
    let Ok(age) = s.parse::<u32>() else {
        return 0;
    };
    age
}
```

- When you care about only one pattern, if let is shorter than a full match.
- let else binds a value or exits early, which keeps the happy path unindented.

04

Methods on Enums

main.rs

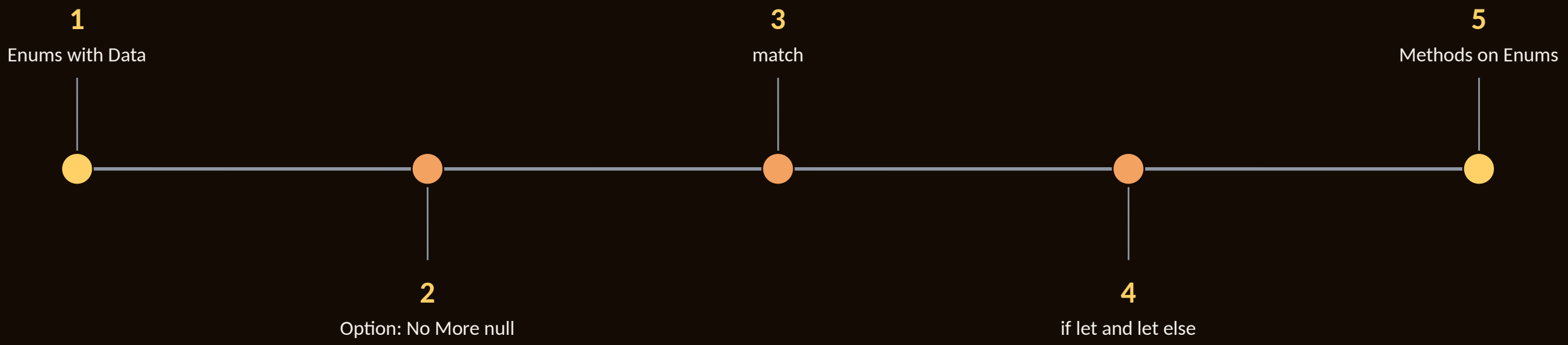
```rust
#[derive(Debug, Clone, Copy)]
enum Light { Red, Yellow, Green }

impl Light {
    fn next(self) -> Light {
        match self {
            Light::Red => Light::Green,
            Light::Green => Light::Yellow,
            Light::Yellow => Light::Red,
        }
    }
}
```

- Enums can have impl blocks with methods, just like structs. Match on self inside them.
- Option and Result themselves are just enums from the standard library, with many helpful …

05

Lesson Map


1
Enums with Data
2
Option: No More null
3
match
4
if let and let else
5
Methods on Enums

Sources and Further Reading

main.rs

```rust
enum Shape {
    Circle { r: f64 },
    Rect { w: f64, h: f64 },
    Triangle(f64, f64, f64),
}

enum Message {
    Quit,
    Move { x: i32, y: i32 },
    Write(String),
}
```

RUST PROGRAMMING · 7

main.rs

```rust
fn area(s: &Shape) -> f64 {
    match s {
        Shape::Circle { r } => 3.14159 * r * r,
        Shape::Rect { w, h } => w * h,
        Shape::Triangle(a, b, c) => heron(*a, *b, *c),
    }
}
let label = match score {
    90..=100 => "A", 75..=89 => "B", _ => "C",
};
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari