LESSON 6

main.rs

```rust
#[derive(Debug)]
struct Point { x: i32, y: i32 }
impl Point { fn new() -> Self {..} }
```

Rust

Structs & impl

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUST PROGRAMMING

LESSON 6

Structs and Methods

Defining structs, impl blocks, &self, associated functions and derive

Defining Structs

main.rs

```rust
struct Student {
    name: String,
    city: String,
    score: f64,
}
let name = String::from("Mariam");
let mut s = Student { name, city: "Kabul".into(), score: 95.0 };
s.score += 2.0;
```

- A struct groups named fields. Create an instance by giving every field a value. Field init …
- The whole instance must be mut to change any field.

01

impl and Methods

main.rs

```rust
impl Student {
    fn grade(&self) -> char {
        if self.score >= 90.0 { 'A' } else { 'B' }
    }
    fn add_bonus(&mut self, points: f64) {
        self.score = (self.score + points).min(100.0);
    }
}
s.add_bonus(5.0);
println!("{}", s.grade());
```

- Methods live in an impl block. The first parameter, ampersand self, borrows the instance. …
- Call methods with a dot. Rust automatically adds the ampersand for you.

02

Associated Functions

main.rs

```
impl Student {
    fn new(name: &str, city: &str) -> Self {
        Self { name: name.to_string(), city: city.to_string(), score: 0.0 }
    }
}
let s = Student::new("Ahmad", "Herat");
```

- Functions in impl without self are associated functions, called with two colons. new is the …
- Self with a capital S means the type itself.

03

derive: Debug, Clone, PartialEq

main.rs

```rust
#[derive(Debug, Clone, PartialEq)]
struct Point { x: i32, y: i32 }

let p = Point { x: 3, y: 4 };
println!("{:?}", p);       // Point { x: 3, y: 4 }
println!("{:#?}", p);      // pretty, multi-line
let q = p.clone();
assert_eq!(p, q);
dbg!(&p);
```

- The derive attribute writes common code for you. Debug lets you print with colon question …
- The dbg macro prints a value with its file and line, great for quick debugging.

04

Tuple Structs and the Newtype Pattern

main.rs

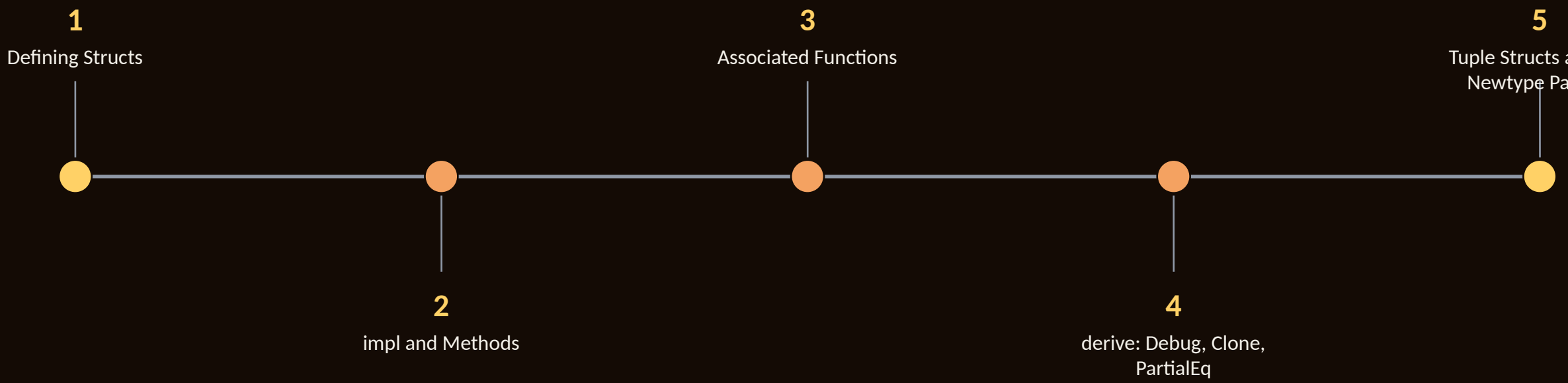
```rust
struct Meters(f64);
struct Seconds(f64);

fn speed(d: Meters, t: Seconds) -> f64 {
    d.0 / t.0
}
speed(Meters(100.0), Seconds(9.6));
// speed(Seconds(9.6), Meters(100.0));   // compile error ✓
```

- Tuple structs have fields without names. A one-field tuple struct, called a newtype, gives …
- Now the compiler stops you from mixing up meters and seconds.

05

Lesson Map


1
Defining Structs
2
impl and Methods
3
Associated Functions
4
derive: Debug, Clone, PartialEq
5
Tuple Structs and the Newtype Pattern

Sources and Further Reading

main.rs

```rust
struct Student {
    name: String,
    city: String,
    score: f64,
}
let name = String::from("Mariam");
let mut s = Student { name, city: "Kabul".into(), score: 95.0 };
s.score += 2.0;
```

RUST PROGRAMMING · 6

main.rs

```rust
impl Student {
    fn new(name: &str, city: &str) -> Self {
        Self { name: name.to_string(), city: city.to_string(), score: 0.0 }
    }
}
let s = Student::new("Ahmad", "Herat");
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari