LESSON 3

main.rs

```rust
fn square(x: i32) -> i32 {
  x * x
}
```

Rust

fn · if · loop

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

RUST PROGRAMMING

LESSON 3

Functions and Control Flow

fn, expressions vs statements, if as an expression, loop, while, for and ranges

Functions

main.rs

```rust
fn square(x: i32) -> i32 {
    x * x                // no semicolon: returned
}

fn average(scores: &[f64]) -> f64 {
    let sum: f64 = scores.iter().sum();
    sum / scores.len() as f64
}
```

- Define functions with fn. Parameter types are required, and the return type follows an arrow.
- The last expression, without a semicolon, is the return value. Adding a semicolon turns it …

01

if Is an Expression

main.rs

```rust
let grade = if score >= 90 {
    'A'
} else if score >= 75 {
    'B'
} else {
    'C'
};
let label = if passed { "pass" } else { "fail" };
```

- if needs no parentheses, and its condition must be a real bool: Rust will not treat a …
- Because if is an expression, you can assign its result directly, like a ternary operator in …

02

loop, while and break Values

main.rs

```rust
let mut n = 1;
let first_big = loop {
    n *= 2;
    if n > 100 { break n; }       // 128
};

'outer: for i in 0..5 {
    for j in 0..5 {
        if i * j == 6 { break 'outer; }
    }
}
```

- loop repeats forever until you break. break can carry a value out of the loop, so loop is …
- while loops while a condition holds. Labels with an apostrophe let you break out of an …

03

for and Ranges

main.rs

```rust
for i in 1..=5 { print!("{i} "); }         // 1 2 3 4 5
for i in (0..10).step_by(3) { print!("{i} "); }   // 0 3 6 9
for i in (1..=3).rev() { print!("{i} "); }  // 3 2 1

let cities = ["Kabul", "Herat"];
for (i, c) in cities.iter().enumerate() {
    println!("{}. {c}", i + 1);
}
```

- for walks over anything iterable: ranges, arrays and collections. Two dots exclude the end, …
- rev reverses, step_by skips, and enumerate gives the index too.

04

Blocks Are Expressions

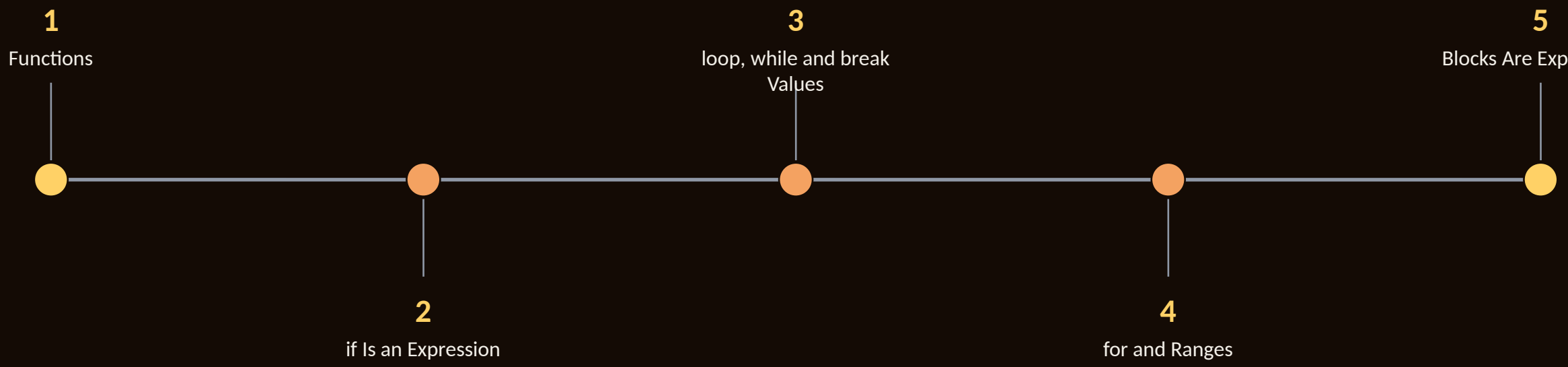
main.rs

```
let area = {
    let w = 3.0;
    let h = 4.5;
    w * h                // value of the block
};
// w and h no longer exist here
println!("{area}");     // 13.5
```

- Any block in curly braces is an expression whose value is its last line. Use this to …
- Variables created inside the block disappear when it ends.

05

Lesson Map


1
Functions
2
if Is an Expression
3
loop, while and break Values
4
for and Ranges
5
Blocks Are Expressions

Sources and Further Reading

main.rs

```rust
fn square(x: i32) -> i32 {
    x * x                 // no semicolon: returned
}

fn average(scores: &[f64]) -> f64 {
    let sum: f64 = scores.iter().sum();
    sum / scores.len() as f64
}
```

RUST PROGRAMMING · 3

main.rs

```rust
let mut n = 1;
let first_big = loop {
    n *= 2;
    if n > 100 { break n; }       // 128
};

'outer: for i in 0..5 {
    for j in 0..5 {
        if i * j == 6 { break 'outer; }
    }
}
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari