

Control-Flow in Rust

Spoken Tutorial Project

<https://spoken-tutorial.org>

National Mission on Education through ICT

<https://sakshat.ac.in>

Jayesh Katta Ramalingaiah
Full Stack Developer

23 November 2020



Learning Objectives

In this tutorial, we will learn about:



Learning Objectives

In this tutorial, we will learn about:

- ▶ `if/else` control flow statements



Learning Objectives

In this tutorial, we will learn about:

- ▶ if/else control flow statements
- ▶ **Different types of Loops supported in Rust**



Learning Objectives

In this tutorial, we will learn about:

- ▶ `if/else` control flow statements
- ▶ **Different types of Loops supported in Rust**
- ▶ **Syntax of all the control flow statements and their usage**



System Specifications

This tutorial is recorded using:



System Specifications

This tutorial is recorded using:

▶ **Ubuntu Linux OS version 18.04**



System Specifications

This tutorial is recorded using:

- ▶ **Ubuntu Linux OS version 18.04**
- ▶ **Rust version 1.47.0**



System Specifications

This tutorial is recorded using:

- ▶ **Ubuntu Linux OS version 18.04**
- ▶ **Rust version 1.47.0**
- ▶ **Visual Studio Code version 1.45.0**
(code editor)



Prerequisites



Prerequisites

- ▶ **You should be familiar with** `compiling and running Rust files`



Prerequisites

- ▶ **You should be familiar with** `compiling and running Rust files`
- ▶ **If not, please go through the prerequisite Rust tutorials on** <https://spoken-tutorial.org>



Code Files

- ▶ The file used in this tutorial is available in the Code files link on this tutorial page



Code Files

- ▶ **The file used in this tutorial is available in the Code files link on this tutorial page**
- ▶ **Pls download and extract the file**



Code Files

- ▶ The file used in this tutorial is available in the Code files link on this tutorial page
- ▶ Pls download and extract the file
- ▶ Make a copy and then use it for practising



Control Flow

- ▶ Control flow means deciding whether to run a piece of code, based on a condition



Control Flow

- ▶ Control flow means deciding whether to run a piece of code, based on a condition
- ▶ This is an integral part of any programming language



Control Flow

- ▶ Control flow means deciding whether to run a piece of code, based on a condition
- ▶ This is an integral part of any programming language
- ▶ This can be achieved using `if` and `else` statements



Control Flow

▶ **if expressions**



Control Flow

- ▶ **if expressions**
- ▶ **else if expressions**



Control Flow

- ▶ We can use `if` without `else`, however `else` cannot be used without `if`



Control Flow

- ▶ We can use `if` without `else`, however `else` cannot be used without `if`
- ▶ `else` is always preceded with an `if` statement



Control Flow

- ▶ We can use one or more `else if` along with the `if` statement



Control Flow

- ▶ We can use one or more else if along with the if statement
- ▶ When you use else if, always follow it up with an else condition too



Control Flow

- ▶ The code works fine even when we don't end with an else statement



Control Flow

- ▶ **The code works fine even when we don't end with an else statement**
- ▶ **To maintain coding standards it is advised to use else condition when we are using else if**



Control Flow - Loops



Control Flow - Loops

- ▶ Loops **are** control structures **that are repeatedly executed until a particular condition fails**



Control Flow - Loops

- ▶ Loops **are** control structures **that are repeatedly executed until a particular condition fails**
- ▶ Loops **repeat a block of code n number of times until a particular condition fails**



Control Flow - Loops



Control Flow - Loops

► for Loop



Control Flow - Loops

- ▶ `for` Loop
- ▶ `while` Loop



Control Flow - Loops

- ▶ for Loop
- ▶ while Loop
- ▶ **Note: There's no do while loop in Rust**



Summary

In this tutorial, we have learnt:

- ▶ if/else control flow statements
- ▶ **Different types of loops**
- ▶ **Their syntax and usage**



Assignment

- ▶ **Go to the project folder** `rust-assignment`
- ▶ **In the `main.rs` file**
 - ▶ **Create a array named `n` and assign 10 random numbers to it**
 - ▶ **Loop over the array**
 - ▶ **Count the number of even values and odd values**
 - ▶ **Print the count of odd and even values**



Assignment

- ▶ Compile **and** execute **the project**
- ▶ **Observe the output in the Terminal**



About Spoken Tutorial project

- ▶ Watch the video available at https://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- ▶ It summarises the Spoken Tutorial project
- ▶ If you do not have good bandwidth, you can download and watch it



Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- ▶ Conducts workshops using spoken tutorials
- ▶ Gives certificates to those who pass an online test
- ▶ For more details, please write to contact@spoken-tutorial.org



Answers for THIS Spoken Tutorial

- ▶ Questions in THIS Spoken Tutorial?
- ▶ Visit <https://forums.spoken-tutorial.org>
- ▶ Choose the minute and second where you have the question
- ▶ Explain your question briefly
- ▶ The Spoken Tutorial project will ensure an answer
- ▶ You will have to register to ask questions



Acknowledgements

- ▶ **Spoken Tutorial project is funded by Ministry of Education (MoE), Govt. of India**



Thank you

Jayesh Katta Ramalingaiah
Full Stack Developer

<https://www.linkedin.com/in/jayeshkattar>

