



Vorlesung

Rust

**Programmieren
in Rust**

Organisation

- 1. Voraussetzungen**
- 2. Ablauf**
- 3. Benotung**
- 4. Termine**
- 5. Ressourcen**



Voraussetzungen

- Alle Fächer aus dem Grundstudium
- Möglichst alle Fächer aus dem 3. Semester
 - Algorithmen und Theoretische Informatik
 - Softwaretechnik
 - Betriebssysteme
- Tools
 - git !

Ablauf

- Lernen von Rust Grundlagen
- Vorbereiten von Rust Grundlagenthemen
 - Vortrag über das Themengebiet
 - Beispielcode vorführen (Livecoding)
 - Aufgaben fürs Labor entwerfen und leiten
- Erstellung einer eigenen Rust Bibliothek (Crate) als 1er oder 2er Team

Benotung

- Benotung
 - ca. 50% Vortrag / Beispielcode aus Rust Grundlagenthema
 - ca. 50% Eigene Rust Crate
- Die genaue Gewichtung ergibt sich im Laufe der Vorlesung, wenn die Verteilung der Themen auf die teilnehmenden Gruppen bekannt ist.

Termine

- Freitag
 - 11:30 Uhr bis 13:00
 - Themenvortrag + Livecoding
 - 14:00 Uhr bis 15.30
 - Laboraufgaben bearbeiten

Zentrale Rust Seite

■ Rust Language



Rust

Empowering everyone to build
reliable and efficient software.

[Install](#)[Learn](#)[Tools](#)[Governance](#)[Community](#)[Blog](#)

GET STARTED

[Version 1.33.0](#)

The Rust 2018 Edition is here!

Why Rust?

Performance

Rust is blazingly fast and memory-efficient: with no runtime or garbage collector, it can power performance-critical services, run on embedded devices, and easily integrate with other languages.

Reliability

Rust's rich type system and ownership model guarantee memory-safety and thread-safety — and enable you to eliminate many classes of bugs at compile-time.

Productivity

Rust has great documentation, a friendly compiler with useful error messages, and top-notch tooling — an integrated package manager and build tool, smart multi-editor support with auto-completion and type inspections, an auto-formatter, and more.