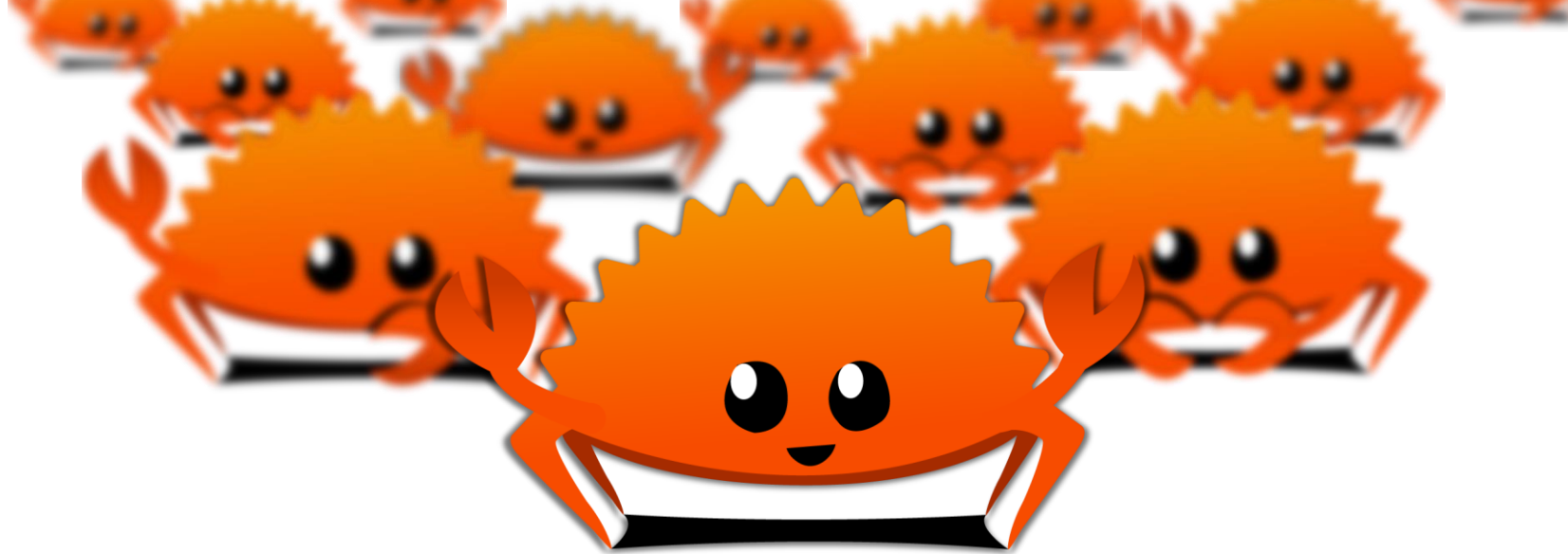


# 20.

---

Rust Community & Open Source

---



“The Rust programming language has many qualities, but Rust’s *greatest strength is the community* of people who come together to make working in Rust a rewarding experience.”

➔ <https://www.rust-lang.org/community.html> ←

# Übersicht

## Discourse Forum

### Users

- Für alles rund um Rust
- Fragen, Ideen, ...

[users.rust-lang.org](https://users.rust-lang.org)

### Internals

- Entwicklung der Sprache
- Pre-RFCs, Ideen, ...

[internals.rust-lang.org](https://internals.rust-lang.org)

## IRC

- **#rust-beginners**: Einfache Fragen
- **#rust**: Alles rund um Rust
- **#rust-internals**: Sprachentwicklung
- **#rust-de**: Deutscher Channel

[Alle \(mit Beschreibung\)](#)

## Reddit



- **/r/rust/**: Hauptsubreddit
  - Neuigkeiten/Ankündigungen
  - Fragen, Blogs, Diskussionen, ...
- **/r/rust\_gamedev/**: Spiele & so

[reddit.com/r/rust](https://reddit.com/r/rust)

## StackOverflow



- **[rust]**: Haupttag
- Fragen nach typischen SO-Regeln
  - “We prefer questions that can be answered, not just discussed.”

[\[rust\]](#)

# Code of Conduct

“We are committed to providing a *friendly*, *safe* and *welcoming* environment for all, regardless of gender, sexual orientation, disability, ethnicity, religion, or similar personal characteristic. Our **code of conduct** sets the standards for behavior in all official Rust forums.”

---

- Regeln zum freundlichen Umgang
- Werden überall (Reddit, IRC, ...) durchgesetzt

“Please be *kind* and *courteous*. There’s no need to be *mean* or *rude*.” — [Code of Conduct](#)

# IRC

- Keine Meta-Fragen: Direkt in erster Nachricht Frage stellen
  - Smalltalk vor eigentlichem Anliegen nur ablenkend
  - Kein „Hallo! Ist gerade jemand da?“
- Viele parallele Diskussionen sind normal
- Viele Nutzer sind nicht aktiv online
- Client
  - [Mibbit](#): Im Browser, einfach für kurze Fragen
    - Bitte nicht zufälligen **mibbit\_f73r89** Namen behalten
  - [HexChat](#): Kostenlos für Windows & Linux, Open Source
  - [u.v.m.](#)

# StackOverflow



- Strengere Regeln: [„How to ask?“](#)
    - Kein Forum!
    - Häufige Meinung: „SO grenzt Neulinge aus“ (*jein!*)
  - Wichtig: Deine Frage gehört nicht dir! ***Es geht nicht um dich!***
    - Kollaborative Sammlung von Wissen in Q&A Form
    - Bearbeiten von fremden Fragen/Antworten vollkommen in Ordnung
  - Downvote heißt nur: „*Im Moment keine gute Frage*“
    - Verbessern durch Bearbeiten wird Punkte & Antworten bringen
- ➔ Regeln befolgen und darauf einlassen!

# Reddit



- Alles rund um Rust
  - Viele Blog Posts
  - Fragen, Diskussionen, ...
  - Neuigkeiten über die Sprache an sich
  - Einfach abonnieren, um auf dem neuesten Stand zu bleiben!
- Bei Nichtübereinstimmung ***nicht downvoten!***
  - Nur konstruktive Kritik erwünscht
- Aber auch: Keine Memes
  - Außer um Silvester ;-)

# Andere Informationsquellen

- Twitter: [@rustlang](#)
- [Offizieller Rust Blog](#)
  - Wichtige Ankündigungen/Änderungen
  - Neue Compiler Versionen
- [This Week in Rust](#)
  - Fasst alle Ereignisse der Woche zusammen
    - Blogs, Community, „Crate of the Week“, ...
    - RFCs, Sprachänderungen, ...
  - Entwicklung von Blog Open Source



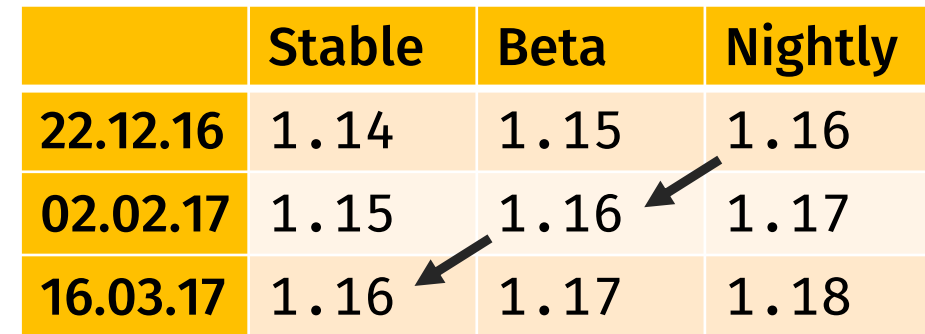
# Entwicklung von Rust

- „Rust Team“: verwalten das Rust Projekt
  - Core Team: Generelle Richtung des Projekts, Leitung
    - Werden fast alle von Mozilla bezahlt
  - [Mehr Infos über Teams](#)
- Sprachdesign, Implementierung, u.s.w. komplett öffentlich
  - Neue Features über RFCs
  - Jeder kann überall mitdiskutieren

# Train Model

- Drei Compiler Versionen: Stable, Beta, Nightly
- Stable: Release alle 6 Wochen
  - Semantic Versioning (später mehr Details)
- Beta: Testphase in der Wildbahn
  - Release mit Stable & für Bugfixes u.ä.
- Nightly: Aktueller Stand von **master**
  - Release „jede Nacht“
  - Erlaubt Nutzung von *unstable* Features

	Stable	Beta	Nightly
<b>22.12.16</b>	1.14	1.15	1.16
<b>02.02.17</b>	1.15	1.16	1.17
<b>16.03.17</b>	1.16	1.17	1.18



# RFC System

- Alle neuen „Sprachfeatures“ müssen in RFC definiert sein
- [Beispiel RFC: Field Init Shorthand](#)
- Tolle Idee? Folgendes Vorgehen:
  - Im Internals Forum ein „Pre-RFC“ Thread anlegen (erste Einschätzung)
  - RFC schreiben und als PR abschicken
  - RFC diskutieren, anpassen und verbessern
  - Wenn gemerged: Im *Tracking-Issue* Implementation besprechen
  - Eventuell: Implementieren und PR abschicken

[github.com/rust-lang/rfcs](https://github.com/rust-lang/rfcs)

# Rust Compiler

- Lebt in: [github.com/rust-lang/rust](https://github.com/rust-lang/rust)
- In Rust geschrieben: Self-Hosted Compiler
  - [Interessanter Artikel über Trusting Trust Attacks für Rust](#)
- Bugfixes, interne Änderungen u.ä. brauchen kein RFC
- Bug Reports: “please report liberally!”
- Weitere Informationen: [CONTRIBUTING.md](#)
- Alles wird mit von drei Bots verwaltet:
  - rust-highfive, bors & rfcbot

# Bots und Crater

- *rust-highfive*: Begrüßt ☺ und weist Reviewer zu
- *rfcbot*: Verwaltet Zustimmung/Ablehnung von Reviewern
- *bors*: Build- & Merge-Bot
  - Wenn PR akzeptiert: Bors Bescheid geben!
  - PR wird [eingereiht](#), um auf allen Plattformen getestet zu werden
  - Wenn alle Tests OK: bors merged
- *Crater*:
  - Testet fragwürdige, rückwärts-inkompatible Änderungen mit allen Crates auf crates.io

# Mithelfen?

- *Warum?* → Erfahrung, Job-Chancen, Spaß, ...
- An Rust/Cargo direkt:
  - Bugs melden (schlechte Fehlermeldungen sind auch Bugs!)
  - Code verbessern, Dinge implementieren, ... (**E-mentor**/**E-easy**)
  - Am Sprachdesign beteiligen
- An anderen Crates
  - Eigene Crate schreiben
  - Bei anderen Crates mithelfen (Bugs melden/PR senden)
- Verunsichert? → Leute fragen!

# Extra: Arten von Rust-Crates

## Bindings

- Name meist **foobar-sys**
- Geben direkten Zugriff auf fremde Bibliothek
- Alle Funktionen unsafe!

➔ Eher unschön!

## Wrapper

- Basiert meist auf **-sys** Crate
- Idiomatic API
- Weiterhin native Abhängigkeiten
- Gutes Design schwierig

## Rein Rust

- Ohne Abhängigkeiten
- Alles in Rust implementiert
- Idiomatic API
- Große Bibliotheken noch selten



Mehr dazu am Mittwoch!