Rust Gotchas, Idioms and Tips

20 January 2022

Tim McNamara, Senior Software Architect TCDI (@timClicks)

Disclaimer

This talk is given in a personal capacity. Its contents should not be construed as the official position of TCDI.

Agenda

1. Intro
2. Magical tips
3. Where to find them

1 minute version

- rust-unofficial.github.io/patterns/idioms
- rust-lang.github.io/rust-clippy/stable/index.html
- rust-lang.github.io/api-guidelines
- rust-lang.github.io/unsafe-code-guidelines

Intro

A note on audience

1.

2.

3.

1. Interested
   never written Rust code, here for the vibe

2.

3.

1. Interested
   never written Rust code, here for the vibe
2.
3.

1. Interested
   never written Rust code, here for the vibe
2.
3.

1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language
   they're learning, confused and frustrated

3.

1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language they're learning, confused and frustrated

3.

1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language
   they're learning, confused and frustrated

3.

1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language they're learning, confused and frustrated

3. Experienced
   Has multiple toolchains installed, is opinionated about the use of match statements

1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language they're learning, confused and frustrated

3. Experienced
   Has multiple toolchains installed, is opinionated about the use of match statements

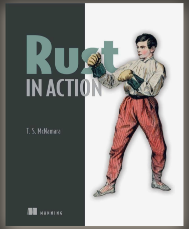
1. Interested
   never written Rust code, here for the vibe

2. Dabbler
   Rust is the next programming language they're learning, confused and frustrated

3. Experienced
   Has multiple toolchains installed, is opinionated about the use of match statements

A note on me

About
Tim McNamara (@timClicks)



Rust in Action

Rust Wellington

@timClicks
youtube.com/c/timClicks

Specialist in natural language processing and text mining, subfields of artificial intelligence (AI) and machine learning (ML).

Organiser of Rust Wellington, a meetup group in New Zealand

Regular (live) tutorials on YouTube/Twitch.

The good stuff

Use clippy

It is quite a good Rust programmer.

```
$ cargo clippy
```

Use TryInto

The as keyword can cause lots of insidious bugs. Use x.try_into().

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=73590b98414191e951df2859cc46ee0f

Single binary executable

Reduce the size of your containers and simplify your operations by compiling your app into a single binary that will work on any Linux system.

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

```
$ RUSTFLAGS='-C target-feature=+crt-static' \
> cargo run \
>   --release \
>   --target x86_64-unknown-linux-musl
```

::default() vs ::new()

Which one is which?

::default() vs ::new()

Which one is which?

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=2e3506695d5a01c2974d1e188b880a19

String concatenation

Use format!()

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=45d31c76252b18e87d2107cda6700d3d

Accepting any text type

You've heard about generics, but there's more.

https://play.rust-lang.org/?version=stable&mode=debug&edition=2021&gist=92de2819564f83ca310931f72f108332

Learning more

Rust Design Patterns

"A catalogue of Rust design patterns, anti-patterns and idioms"

rust-unofficial.github.io/patterns

Clippy

Read through some of the warn and style lints.

rust-lang.github.io/rust-clippy/stable/index.html

API Guidelines

A well-written collection of conventions for library crates (and other Rust!).

rust-lang.github.io/api-guidelines

Unsafe Code Guidelines

Useful for understanding how Rust looks to the compiler.

rust-lang.github.io/unsafe-code-guidelines/introduction.html

Thanks!

Questions?

Tim McNamara

tim@mcnamara.nz
https://tim.mcnamara.nz
https://twitter.com/timClicks
https://youtube.com/c/timClicks
https://linkedin.com/in/timmcnamaranz