

Rust for the rest of the world

RUSTAsia 2025

Tim McNamara

rust@timclicks.dev

28 Mar 2025

github.com/timClicks/talks

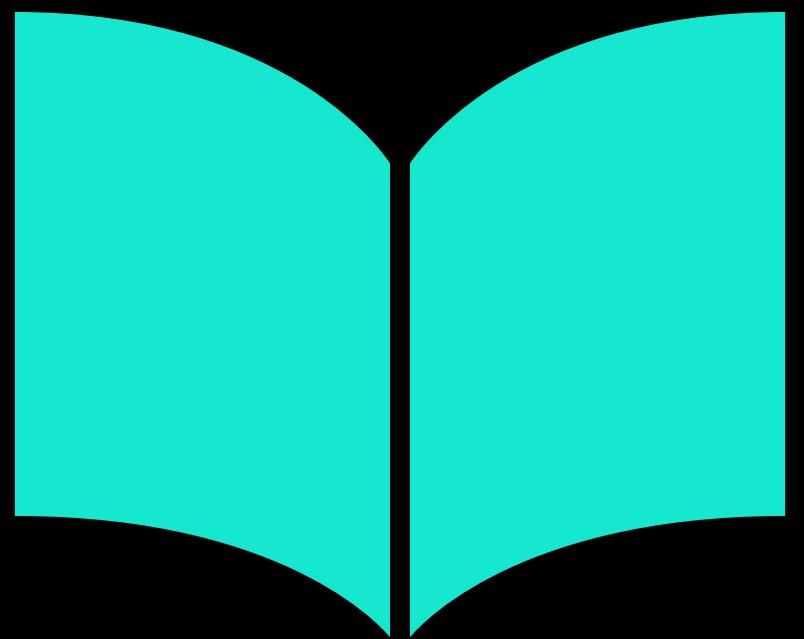
pub mod agenda;

Agenda

- Introduction
- Problem statement: Rust's development is too heavily concentrated
- How to influence change: within yourself, your team, your city and the world

pub mod introduction;

An unexpected lesson



Rust in Action



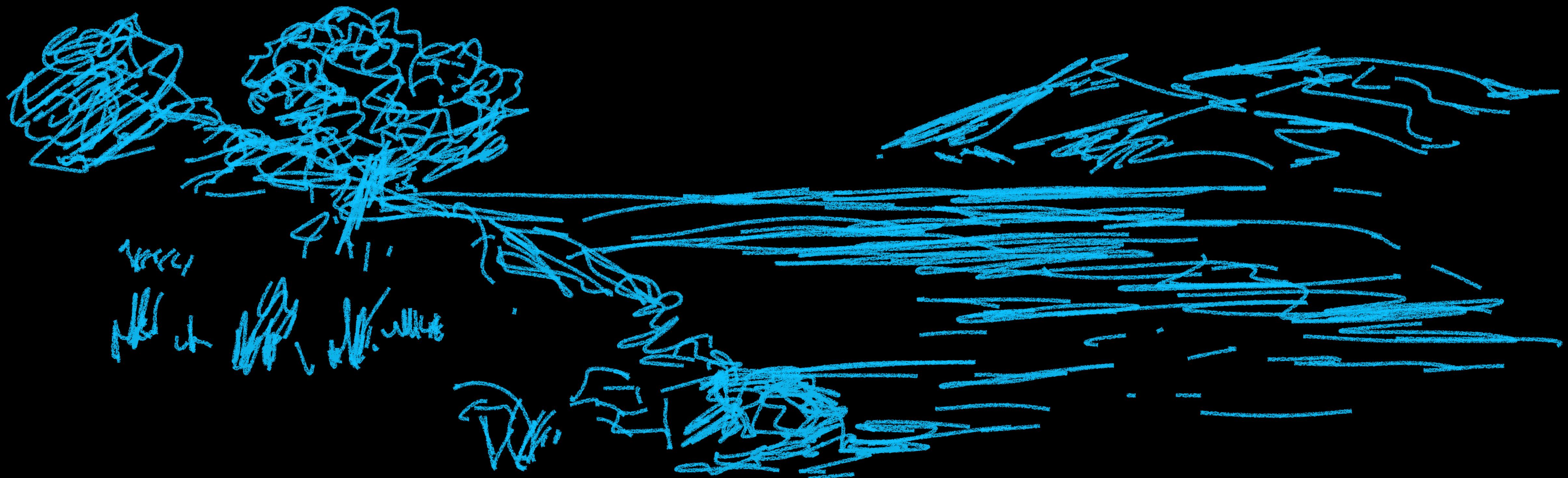
\$ whoami timclicks

Tim McNamara

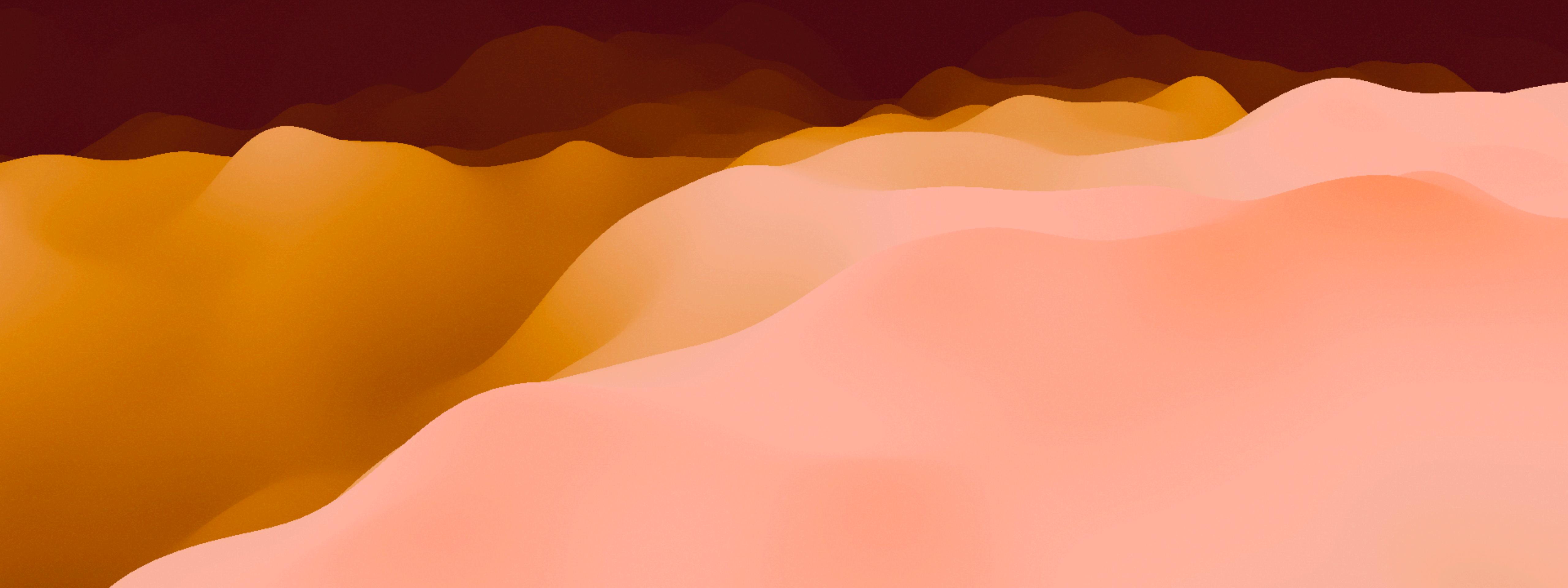
- Author of [Rust in Action](#)
- Known on YouTube and social media as [timClicks](#)
- Provide tutorials and videos at [timclicks.dev](#)
- Host of the [Compose](#) podcast
- Organiser of the [Rust Forge](#) conference
- Owner of Rust consultancy [Accelerant.dev](#)
- Father of [3 wonderful girls](#)

Rust 1.0

Where were you and what were you thinking?



Where is Rust going?





Did you know that you
can influence its direction?

Cohesion?



Chaos?

Light?

The background features a vibrant, abstract design composed of overlapping organic shapes in various colors. Large areas of orange, purple, and blue are prominent, with smaller patches of green, brown, and yellow. The shapes are fluid and layered, creating a sense of depth and movement.

You choose



Empowering everyone

What does this actually mean?

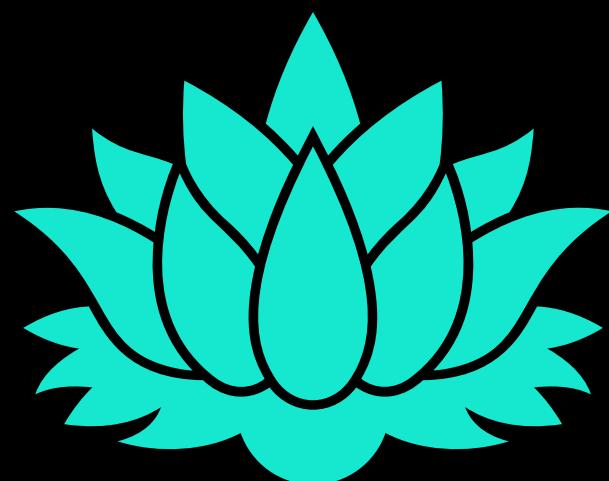
How Rust is structured

Rust users, Rust project, Rust Foundation

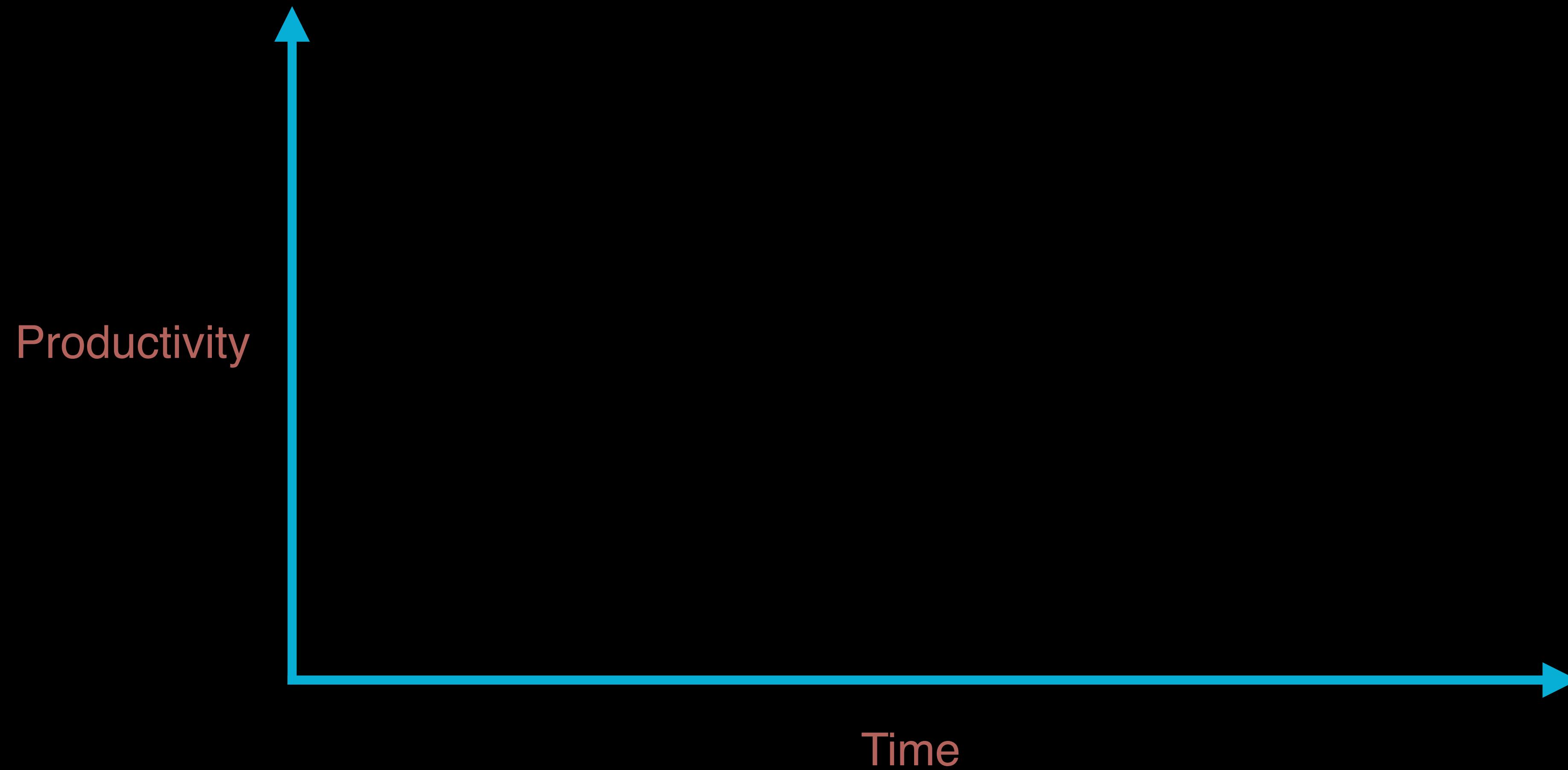
How to grow the language

How to grow the language

Work outwards from what you control



pub mod personal_actions;



Productivity with a new tool over time

Make a choice

To see progress, you must make progress.

Now make another choice

When will you abandon Rust?

How to learn Rust

How to learn how to program Rust

- Respect that learning a new skill is difficult
- Respect your prior knowledge
- Read lots of code
- Build many small projects

Respect that learning a new skill is difficult

Debug
or
Display

Clone
or
Copy

--debug
or
--release

String
or
&str

`String`

`or`

`&str`

`or`

`Box<str>`

`Cow<'a str>`

`Arc<str>`

nightly
or
stable

const
or
static

HashMap
or
BTreeMap

cargo
or
rustup

Option
or
Result

`::default()`
or
`::new()`

Why does Rust have so many small choices?

- Rust values precision
- Rust offers control
- Rust is explicit

Your mental model

Why learning a new skill is difficult

- Rust introduces many new concepts
- When learning Rust, your prior knowledge does not allow you to place these concepts into the correct categories

Which integer size is best? async? What is a reference?

What is a reference? Is unwrap safe? What is the heap?

Why does unwrapping something cause the program to crash?

What is a “trait”? Why can’t I sort floats? What is a box?

If a variable is the owner, does that mean there are no locks?

What is mutation? What is a lifetime? Move semantics?

Why do people say object when there are no objects in Rust?

Respect your prior knowledge

Use analogies that make sense for you

- If a term doesn't make sense, then try another term

Read lots of code

And your brain works best if you read it out loud

- Code from <https://github.com/timClicks/tutorials>

```
# [derive(Debug)]
struct Book {
    pages: u32,
}

# [derive(Debug)]
struct Shelf {
    books: Vec<Book>,
}

fn read<'a, 'b: 'a>(b: Option<&'a Book>, s: &'b Shelf) -> &'a Book {
    let book = b.unwrap_or_else(|| { &s.books[0] });
    println!("{}: {} pages", book.title, book.pages);
    book
}
```

```
use itertools::Itertools as _;

// copied from Rosetta Code
// https://rosettacode.org/wiki/Pascal's_triangle#Rust
fn pascal(rows: u64) {
    for row in 0..rows {
        let mut col = 1;

        for _ in 1..=2 * (rows - 1 - row) {
            print!(" ");
        }

        for position in 0..=row {
            print!("{}{:3} ",);

            col = col * (row - position) / (position + 1);
        }

        println!()
    }
}
```

Write lots of code

Smart with examples that are small and fun

- Dopamine likes rewards. Start with things that are easy.
- Rewrite a small project that you have already implemented.
- Write something that you would be comfortable writing in another language
- Then get bigger...

A TUI with ratatui

Hack a magic badge

An app with WebAssembly

A parser for your work's custom
internal file format with pest

A native app with Tauri

Speed up Go/Python/Java/
JavaScript with a native extension

A smart contract with Solana

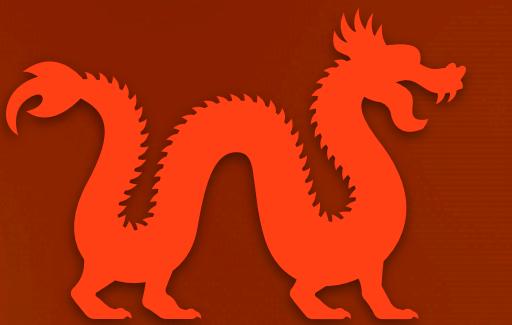
A fleet of autonomous drones with Zenoh

A high frequency trading engine

Run a secure unikernel

Create a certified compiler

RUSTAsia ftw



How to teach Rust

How to teach Rust

Teaching is also a skill

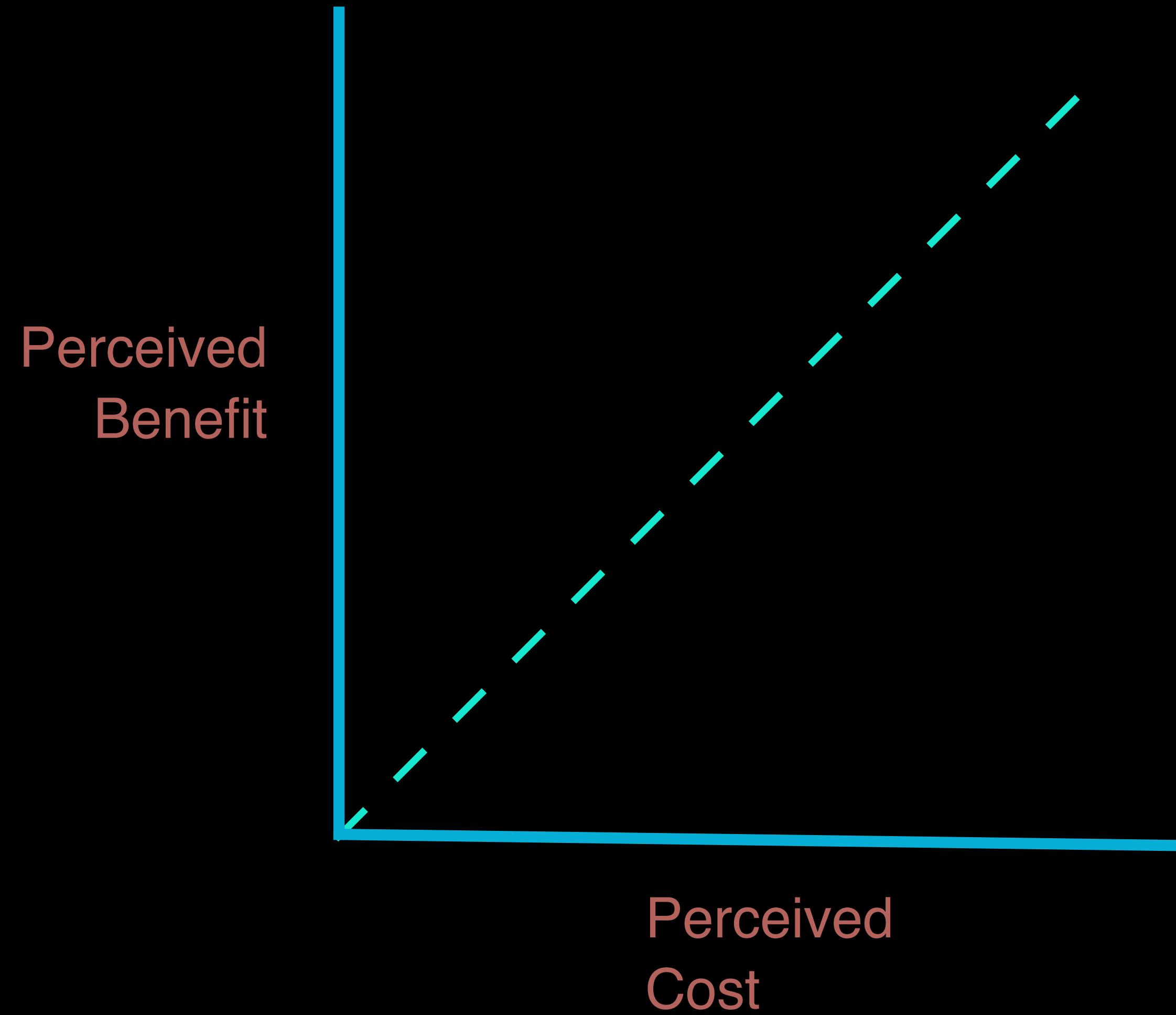
- Respect that learning a new skill is difficult
- Respect your prior knowledge
- Observe teachers teaching
- Practice often

How to teach Rust

Some specific advice

- Make mistakes
- Sacrifice accuracy for understanding
- Be mindful of expert blindness
- Never teach alone
- Always get feedback

pub mod team;



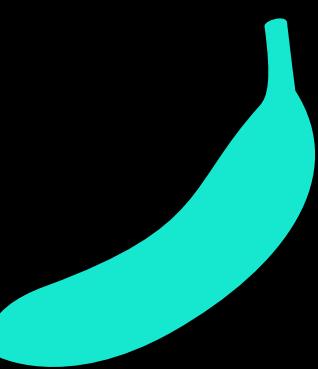
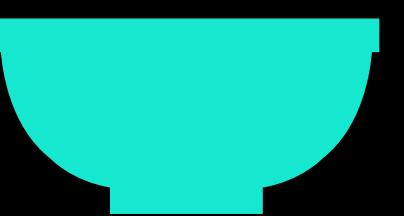
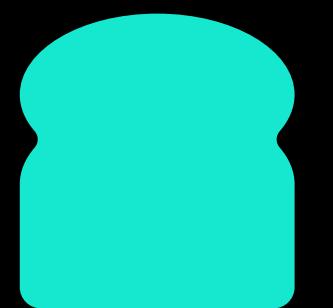
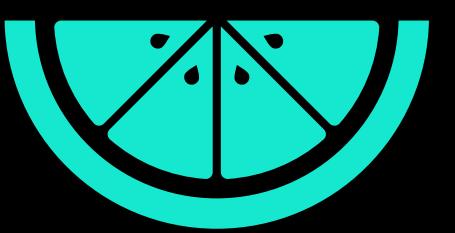
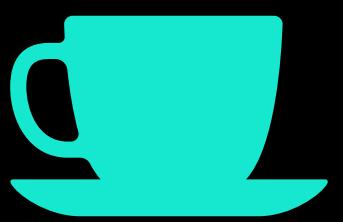
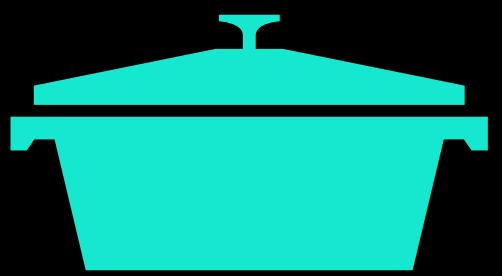
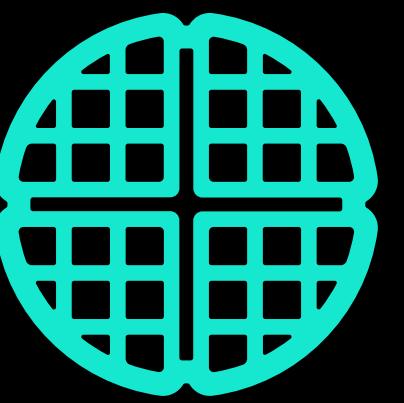
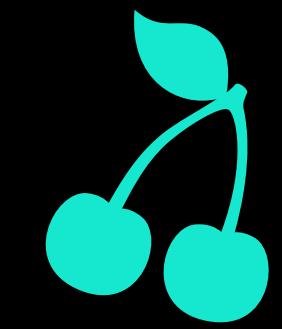
When advocating for Rust at work, remember that it is a business decision, not a technical one.

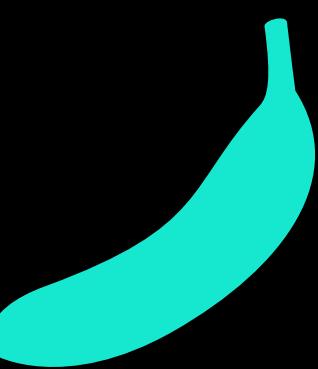
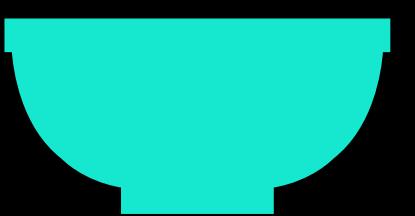
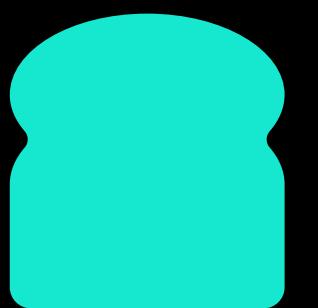
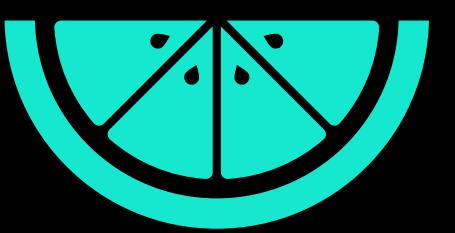
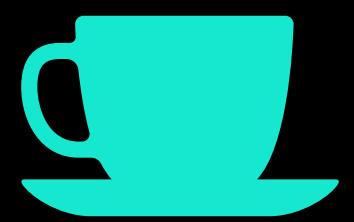
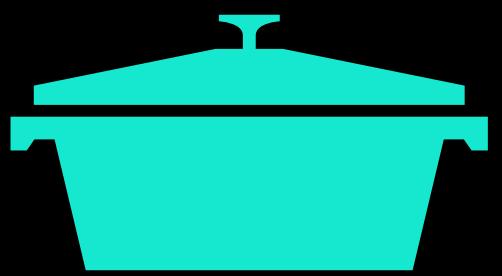
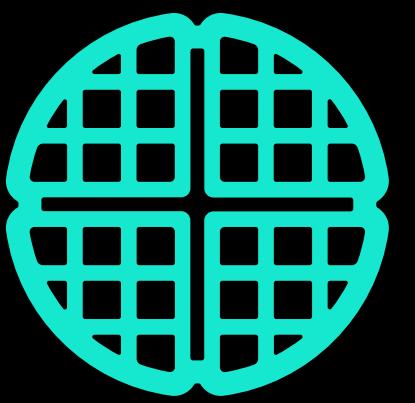
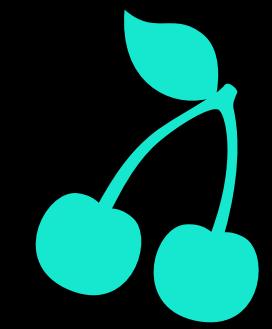
Sharing Rust with your team

It's harder than it looks

- Research how other languages were adopted
- Find an area that is having difficulty, but is not critical
- Expect to receive questions as the “Rust expert” on the team

pub mod local;





pub mod worldwide;

Rust vision 2025 project

A chance for you to contribute to the Rust you want to exist

- <https://rust-lang.zulipchat.com/#narrow/channel/486265-vision-doc-2025>
- Interviews in April, to contribute to a planning session at the all hands meeting in the Netherlands in May

Rust Project

You are welcome there

- Participate in Zulip threads
- Review RFCs
- Comment on GitHub issues

Rust Foundation

Dedicated community liaison available

- Ernest Kissiedu
- Focusing first on Africa, Middle East, Latin America, but very keen to learn how to help

Accelerant - Rust adoption support

tim@accelerant.dev

LinkedIn.com/in/timmcnamaranz

@timClicks

<https://timclicks.dev>

rust@timclicks.dev

@timClicks

Bluesky, Discord, Github,
YouTube, Mastodon, Twitter