Rust Kata: Ideas in Improving Rust Teaching Tools

Tom Kunc

@tfpk_

tfpk

tom@tfpk.dev

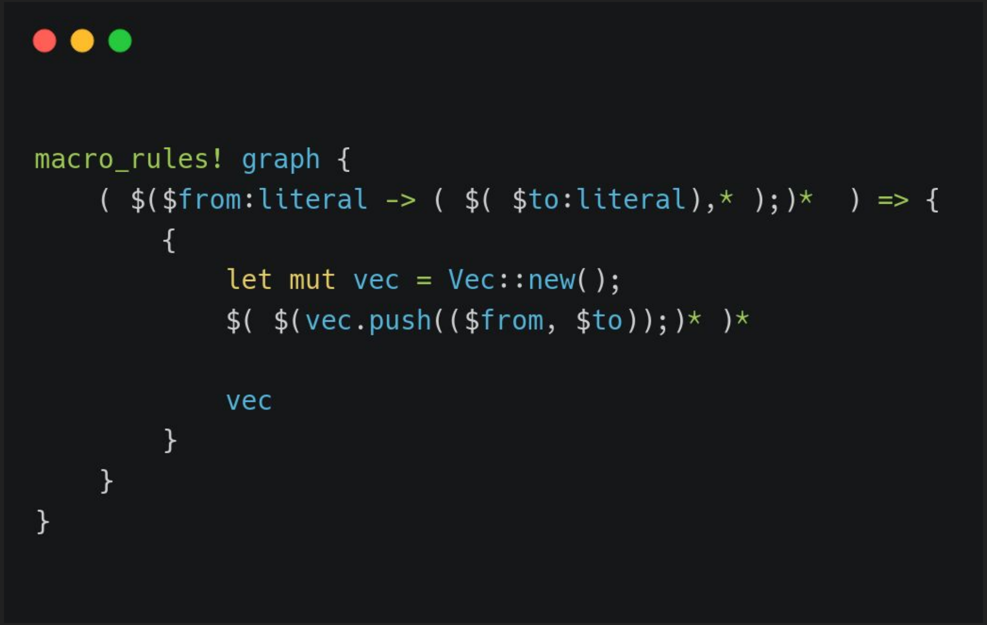
Outcomes

1. Highlight *MacroKata* and *LifetimeKata* as useful resources.
2. Discuss how Rust allows for building tools that help with learning.
3. Explore areas Rust could be further improved.

MacroKata

Survey

Who feels they understand this code? Who could write it?

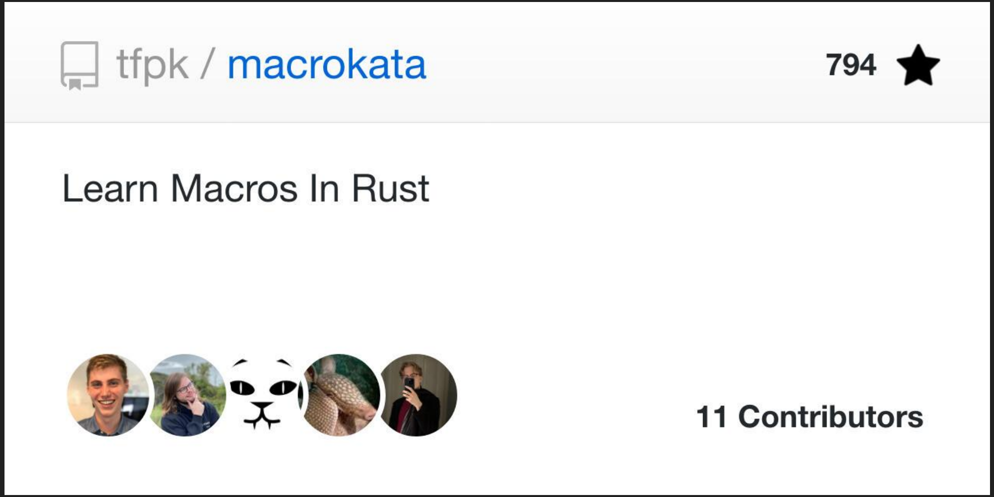


```
macro_rules! graph {
    ( $($from:literal -> ( $( $to:literal),* );)*  ) => {
        {
            let mut vec = Vec::new();
            $( $(vec.push(($from, $to));)* )*

            vec
        }
    }
}
```

MacroKata: Purpose

- Currently no good macro teaching tools
- TLBORM is a reference
- Rust Book too high-level


tfpk / macrokata

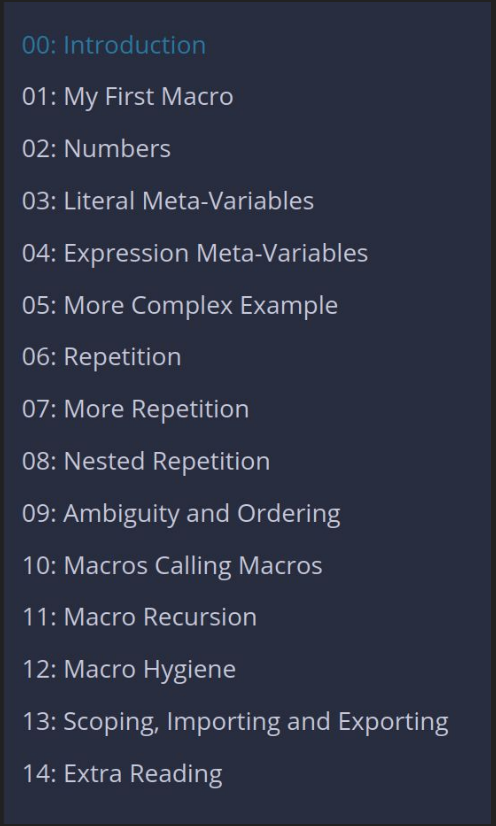
794

Learn Macros In Rust

11 Contributors

MacroKata: Structure

- Teaches incremental steps
- Short reading followed by short exercise
- Exercises are machine checked: they have to produce identical *code*



MacroKata: How The Language Helped

- Using *cargo-expand* to *show* what the macro does.
- mdBook is a great tool

Evaluation

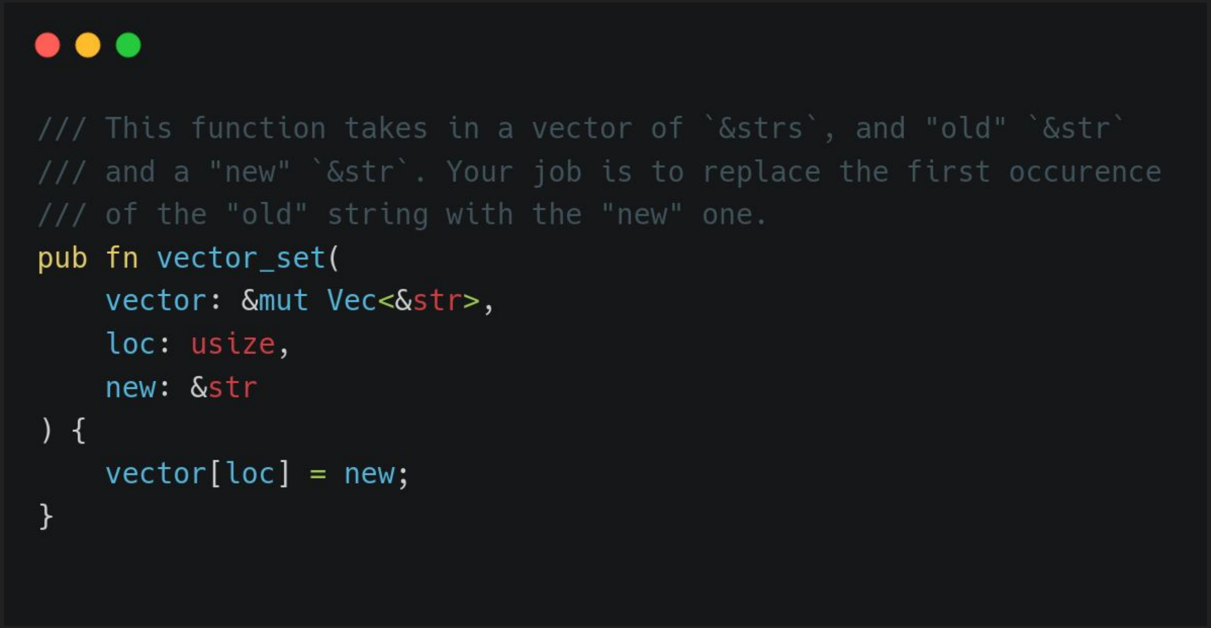
- People clearly saw a need for this (800 stars, good feedback so far)
- Used in COMP6991 (UNSW Rust Course)

Made me wonder, are there other places this structure could help us teach…

LifetimeKata

Survey

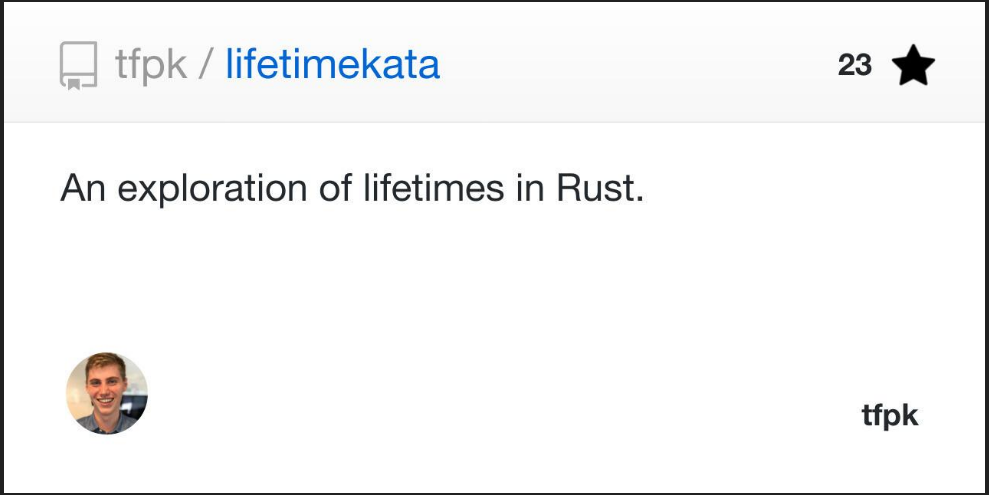
Who feels they could write the lifetimes for this code?



```rust
/// This function takes in a vector of `&strs`, and "old" `&str`
/// and a "new" `&str`. Your job is to replace the first occurence
/// of the "old" string with the "new" one.
pub fn vector_set(
    vector: &mut Vec<&str>,
    loc: usize,
    new: &str
) {
    vector[loc] = new;
}
```

LifetimeKata: Purpose

- No intermediate resources on learning Lifetimes
- Lifetimes are introduced too late (Lifetime Elision considered harmful)
- “The first time you understand lifetimes is when you write them out”



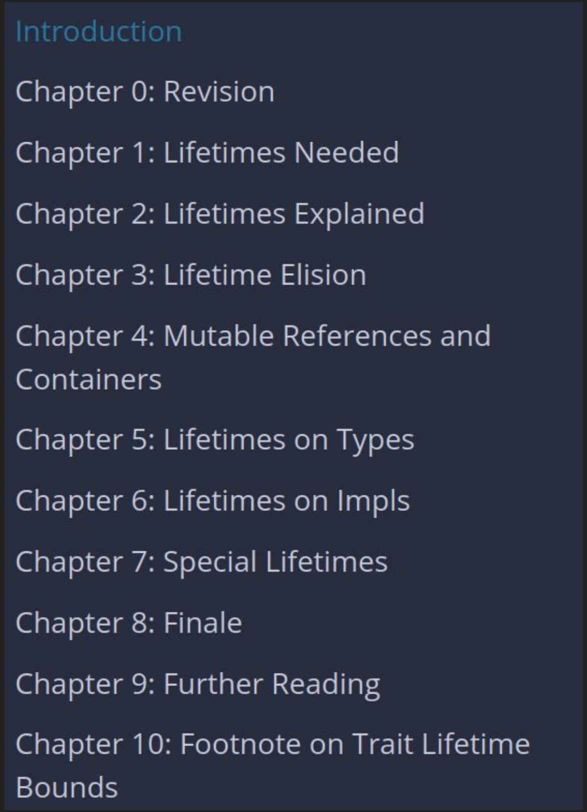
tfpk / lifetimekata 23

An exploration of lifetimes in Rust.

tfpk

LifetimeKata: Structure

- Similar to MacroKata
- Turn off lifetime elision while completing the tasks.
- Not all exercises are programming exercises.



LifetimeKata: How The Language Helped

- The `require_lifetimes` proc macro forces the use of lifetimes everywhere.
- Being able to inspect the code makes the macro possible.

Observations

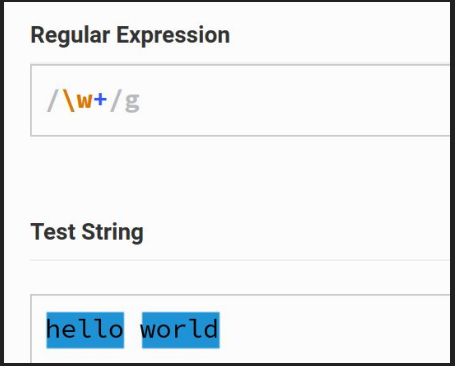
- Teaching “intermediate” concepts in Rust is hard.
- We can use the tools in the language to make teaching easier.
- There’s still lots of work to do!

Promising Innovations

- RustViz
- RustEdu
- RustAnalyzer
- Rust Book with Quizzes
- Rustlings
- Learn Rust in a Month of Lunches

Future Ideas

- A "regexpal" approach to macros



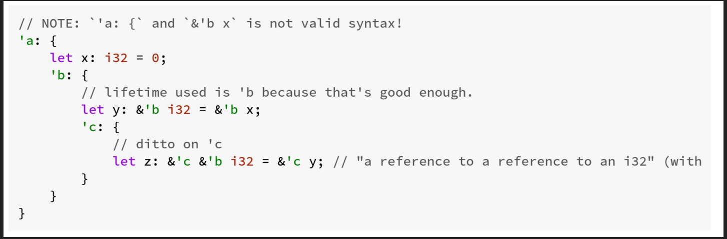
Regular Expression

```
/\w+/g
```

Test String

```
hello world
```

- Lifetime Annotations within a function
- Integrate Lifetime-Viz as part of Rust-Analyzer



```
// NOTE: `'a: {` and `&'b x` is not valid syntax!
'a: {
    let x: i32 = 0;
    'b: {
        // lifetime used is 'b because that's good enough.
        let y: &'b i32 = &'b x;
        'c: {
            // ditto on 'c
            let z: &'c &'b i32 = &'c y; // "a reference to a reference to an i32" (with
        }
    }
}
```

- Rust-Book annotated with exercises as the default (linking Rustlings to the Book)

What Next?

- Are those ideas the right direction to go?
- What other areas of the language need this sort of support?
- What other tools can be built to support this?