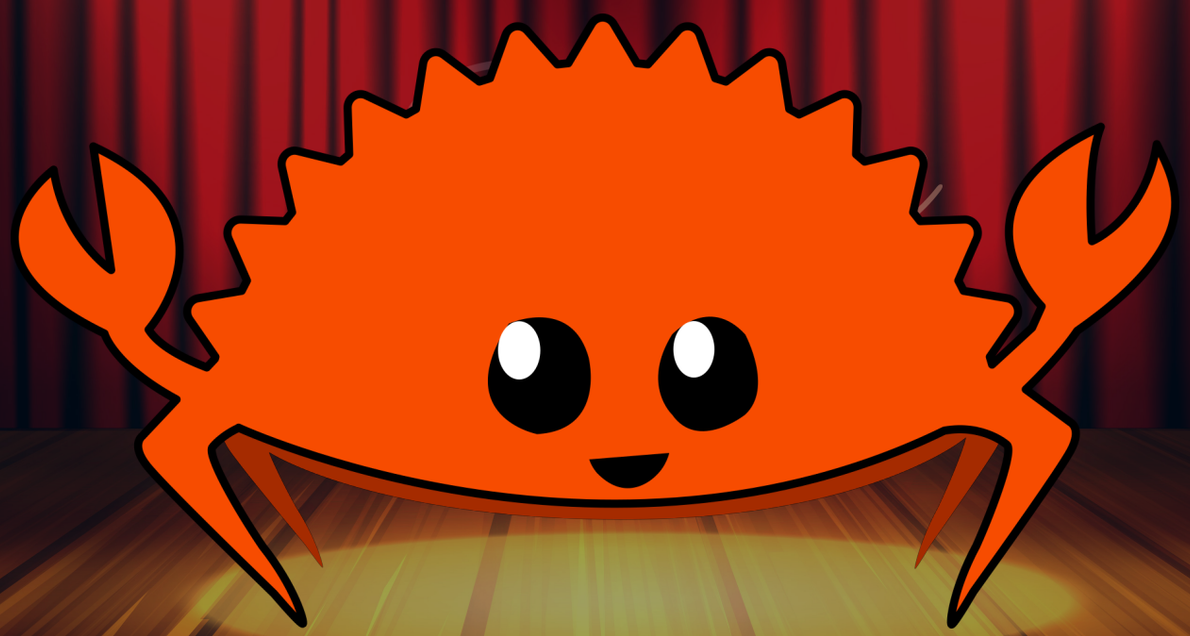
What Is Rust Doing Behind the Curtains?



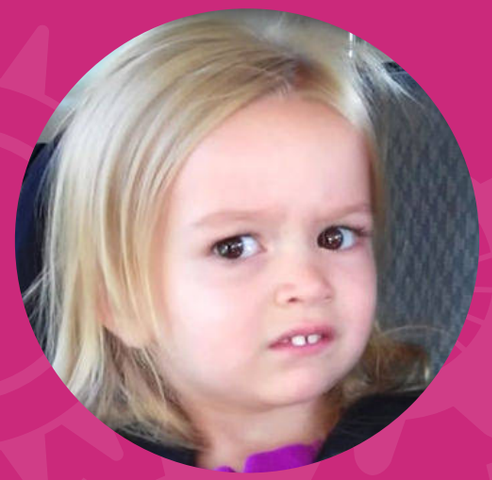
Hi! I am

Matthias Endler

You might know me from...

My YouTube channel!

My Blog!



Not at all!

trivago®

- Hotel Search Platform
- 2.5m+ Hotels/Accommodations
- IT departments in Düsseldorf, Leipzig, Palma, Amsterdam
- Java, Kotlin, Go, PHP, Python (, Rust?)
- tech.trivago.com

Why should I care?

Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety. It aims to bring modern language design and an advanced type system to systems programming. Rust does not use a garbage collector, using advanced static analysis to provide deterministic drops instead.

Rust is a systems programming language that runs blazingly fast, prevents **segfaults**, and guarantees **thread safety**. It aims to bring modern language design and an **advanced type system** to systems programming. Rust does not use a **garbage collector**, using advanced **static analysis** to provide **deterministic drops** instead.

Empowering everyone to build reliable and efficient software.

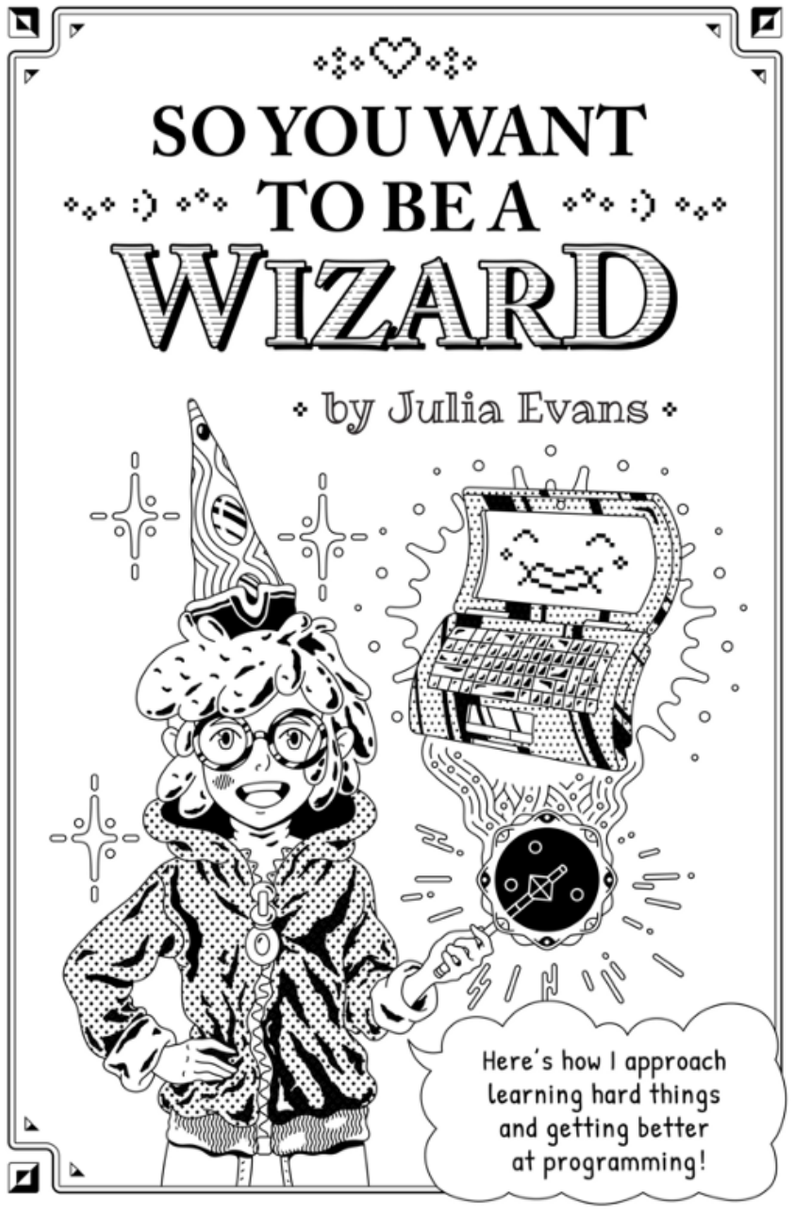

Transfraid.

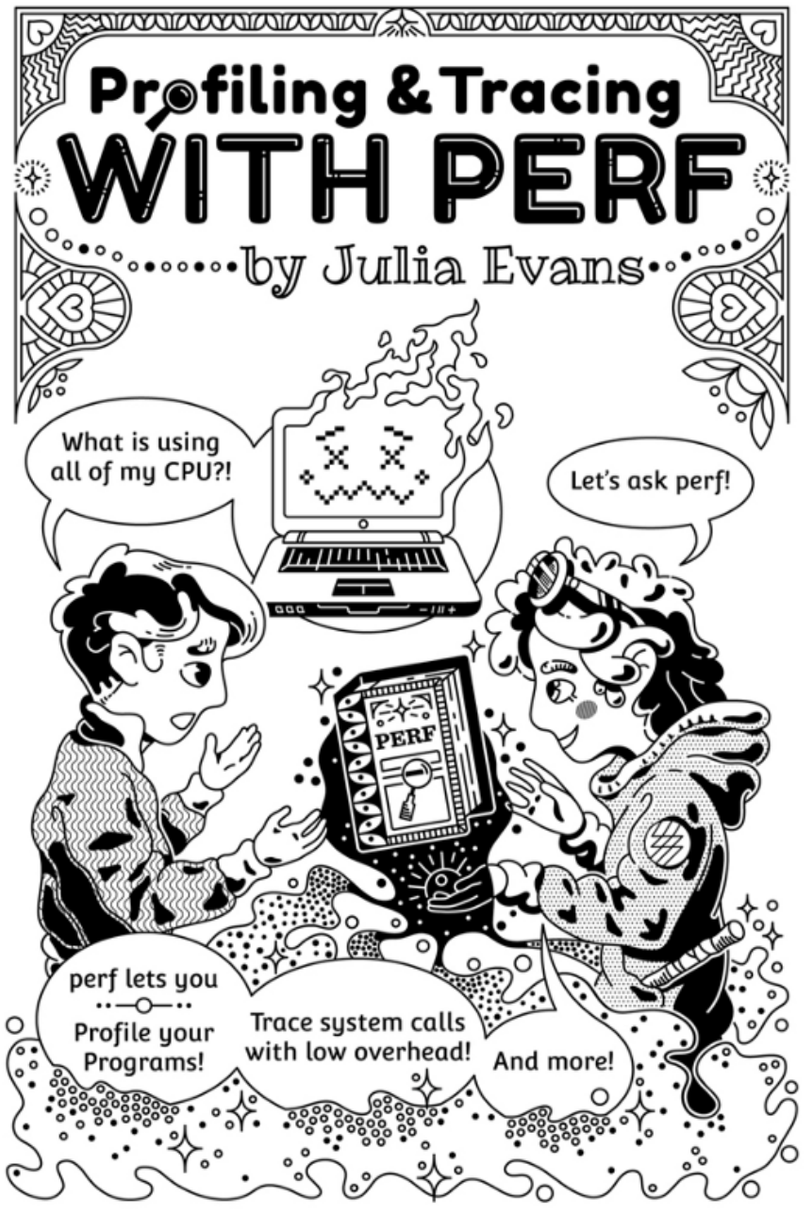

Welcome to
2018
Rust
CONF
Presented by TILDE
Registration
Rust CONF

Being curious is an amazing trait!
We should embrace it,
and help people be curious.

Pascal Hertleif –
Rust's approach of getting things right

Julia Evans

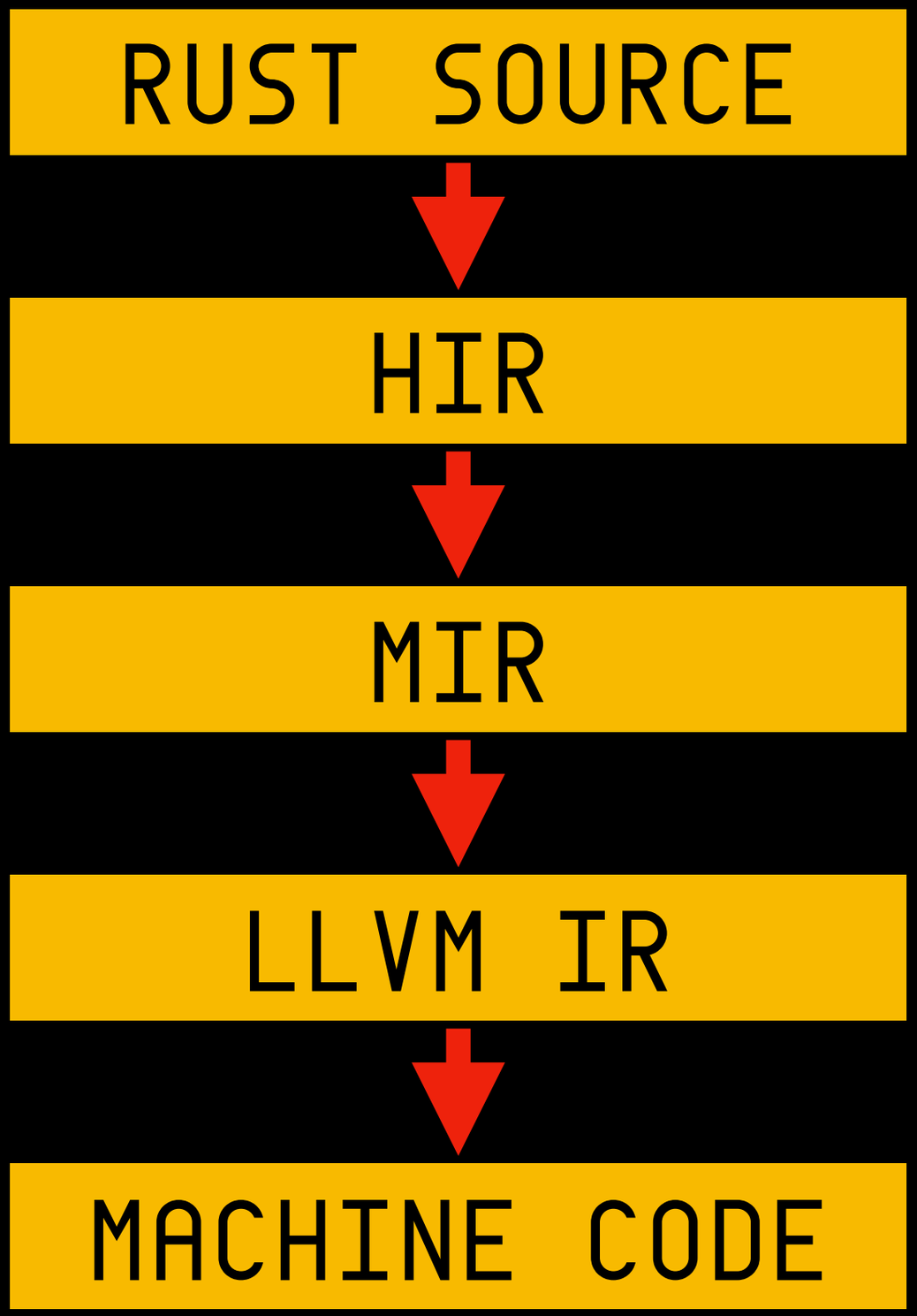

SO YOU WANT TO BE A
WIZARD
by Julia Evans
Here's how I approach learning hard things and getting better at programming!


Profiling & Tracing
WITH PERF
by Julia Evans
What is using all of my CPU?!
Let's ask perf!
PERF
perf lets you
Profile your Programs!
Trace system calls with low overhead!
And more!


let's learn
tcpdump
by Julia Evans
WHAT ARE THOSE COMPUTERS SAYING TO EACH OTHER?
WITH TCPDUMP, YOU CAN FIND OUT!
TCP
UDP

Why's Poignant Guide to Ruby

The Rust Compiler



Parsing

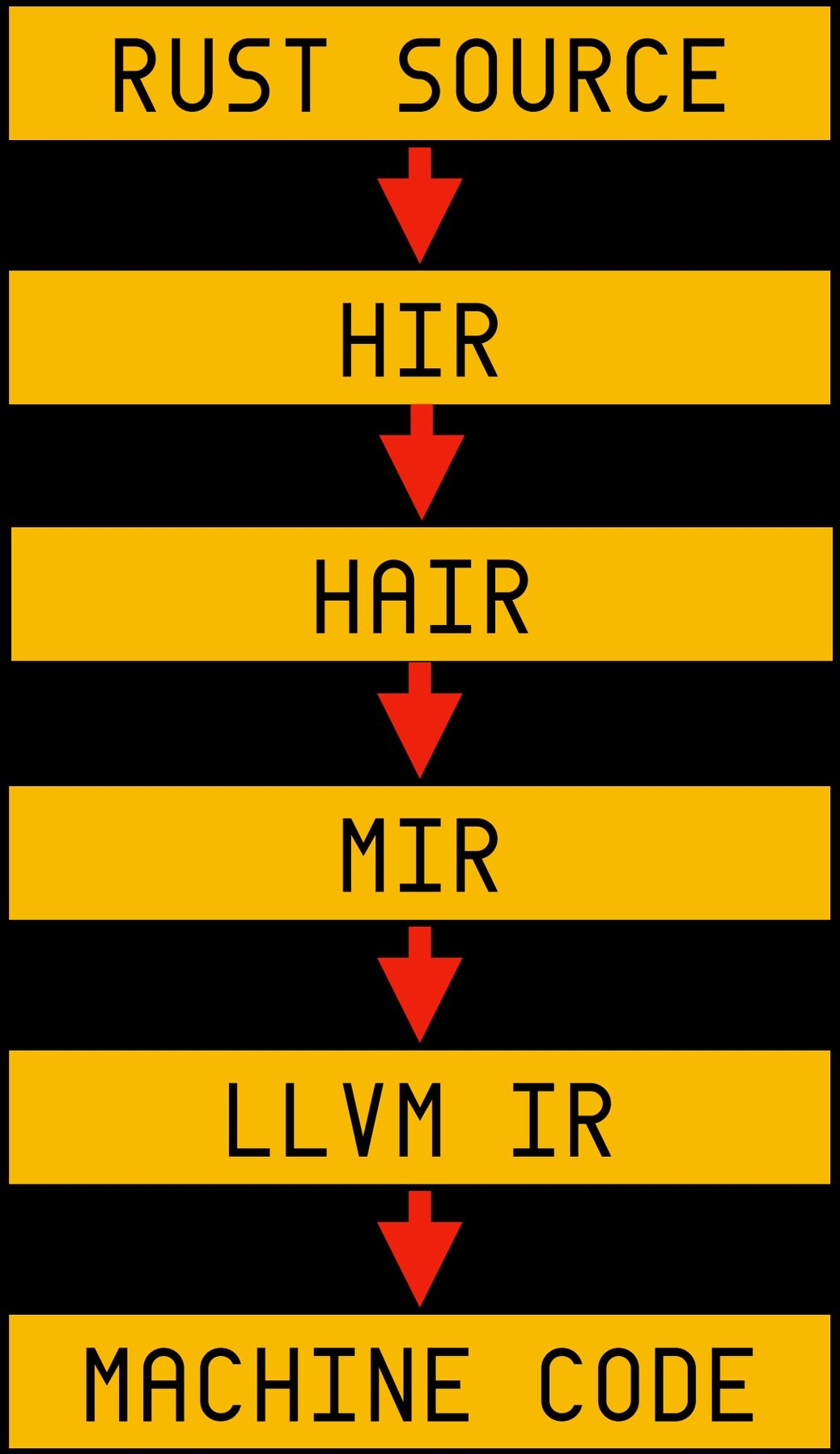
Desugaring

Borrow-checking

Optimization

Optimization

Optimization



Parsing

Desugaring

Optimization

Optimization

Borrow-checking

Optimization

Optimization

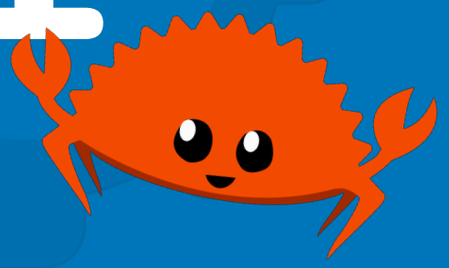
Desugaring...

Candy designed by Freepik, Vegetables by Macrovector

At last...

Code examples!

Example 1



```rust
fn main() {}
```

```rust
#[macro_use]
extern crate std;

#[prelude_import]
use std::prelude::v1::*;

fn main() {}
```

```
Types:  std::boxed::Box
        std::option::Option::{self, Some, None}
        std::result::Result::{self, Ok, Err}
        std::string::String;
        std::vec::Vec

Traits: std::borrow::ToOwned
        std::clone::Clone
        std::cmp::{PartialEq, PartialOrd, Eq, Ord }
        std::convert::{AsRef, AsMut, Into, From}
        std::default::Default
        std::iter::{DoubleEndedIterator, ExactSizeIterator}
        std::iter::{Iterator, Extend, IntoIterator}
        std::marker::{Copy, Send, Sized, Sync}
        std::ops::{Drop, Fn, FnMut, FnOnce}
        std::slice::SliceConcatExt
        std::string::ToString

Functions:  std::mem::drop
```

Types: Box, Option, Result, String, Vec

Traits:

PartialEq, PartialOrd, Eq, Ord — Ordering things

AsRef, AsMut, Into, From, ToOwned, Clone, ToString — Converting things

Default — Default values

DoubleEndedIterator, ExactSizeIterator
Iterator, Extend, IntoIterator — Iteration

Copy, Send, Sized, Sync — Marker traits

Drop, Fn, FnMut, FnOnce — Calling/Dropping objects

SliceConcatExt — Concatenate objects (like strings or vectors)

Example 2

Ranges

```
for i in 0..3 {
    // do something with i
}
```

```
let range = 0..3;
for i in range {
    // do something with i
}
```

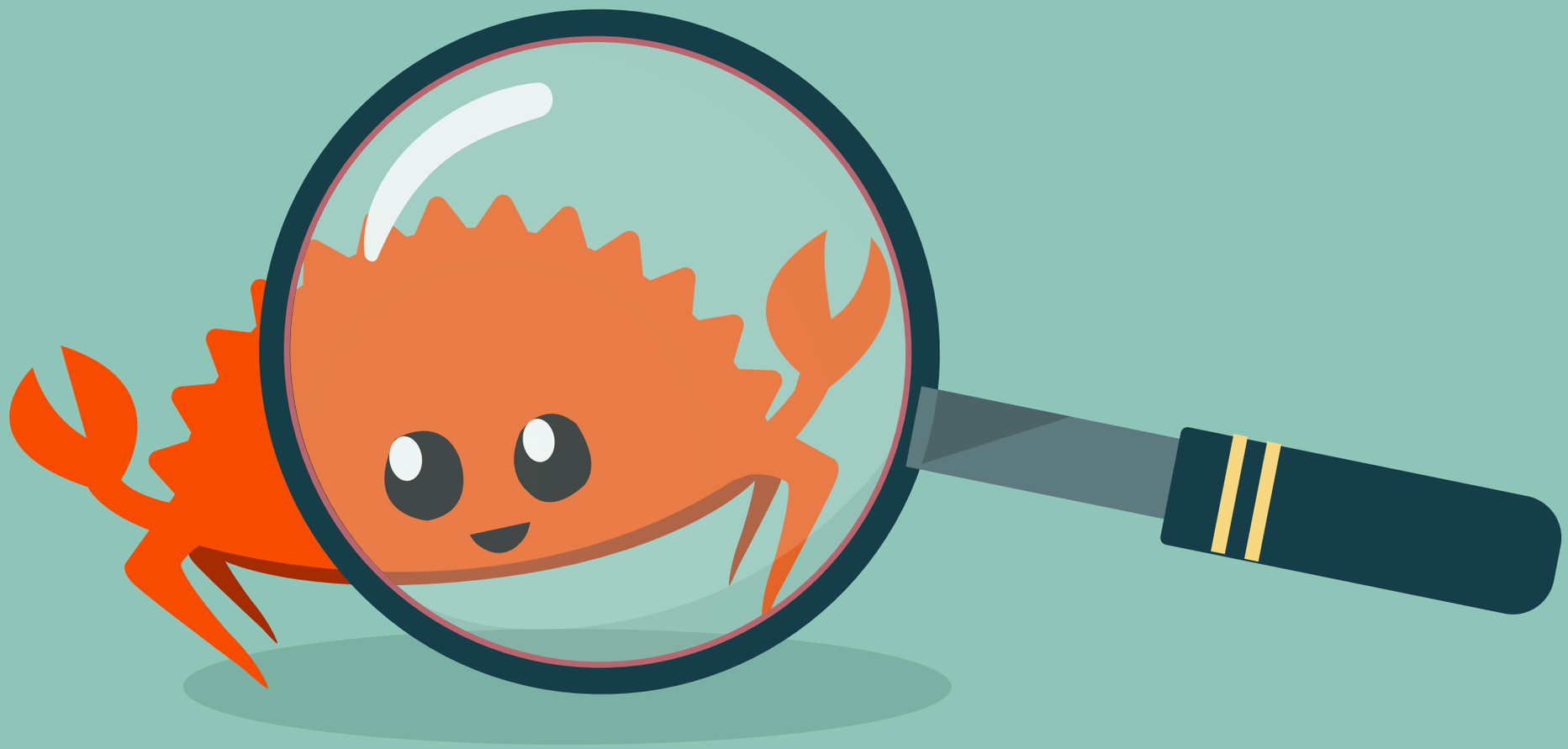
```
let range = Range {0, 3};
for i in range {
    // do something with i
}
```

```
let range = Range {0, 3};
for i in range {
    // do something with i
}
```

```rust
use std::ops::Range;

let range = Range { start: 0, end: 3 };
for i in range {
    // do something with i
}
```

```rust
use std::iter::IntoIterator;
use std::ops::Range;

let range = Range { start: 0, end: 3 };
let mut iter =
IntoIterator::into_iter(range);

while let Some(i) = iter.next() {
    // do something with i
}
```

```rust
use std::iter::IntoIterator;
use std::ops::Range;

let range = Range { start: 0, end: 3 };
let mut iter =
IntoIterator::into_iter(range);
loop {
    match iter.next() {
        Some(i) => { /* use i */ },
        None => break,
    }
}
```



cargo inspect

```
cargo-install cargo-inspect

cargo inspect foo.rs
```

Example 3

Ranges - Part II

```rust
for i in 0..=3 {
    // do something with i
}
```

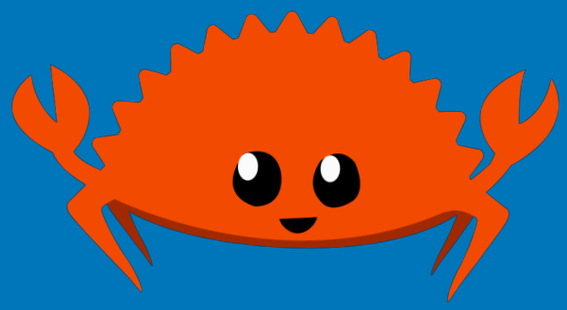
```rust
use std::iter::IntoIterator;
use std::ops::RangeInclusive;

let range = RangeInclusive::new(0, 3);
let mut iter =
IntoIterator::into_iter(range);
```

```rust
loop {
    match iter.next() {
        Some(i) => { /* use i */ },
        None => break,
    }
}
```

```
cargo inspect --diff foo.rs,bar.rs
```

Example 4

Opening Files



```rust
use std::fs::File;
use std::io::Error;

fn main() -> Result<(), Error> {
    let f = File::open("file.txt")?;
    Ok(())
}
```

```rust
use std::fs::File;
use std::io::Error;

fn main() -> Result<(), Error> {
    let f = match File::open("file.txt") {
        Ok(file) => file,
        Err(err) => return Err(err),
    };
    Ok(())
}
```

```rust
use std::fs::File;
use std::io::Error;
use std::convert::From;

fn main() -> Result<(), Error> {
    let f = match File::open("file.txt") {
        Ok(file) => file,
        Err(err) => return Err(From::from(err)),
    };
    Ok(())
}
```

main.rs

src ▸ main.rs

```rust
fn main() {
    let mut vec = vec![1, 5, 10, 2, 15];
    vec.sort();
}

```

cargo-inspect-vscode

More cargo tools!

- cargo-expand
- cargo-asm
- cargo-bloat

Rust Playground

SHOW MIR ▶ ··· DEBUG STABLE ··· SHARE TOOLS CONFIG ?

```
1  pub fn square(num: i32) -> i32 {
2      num * num
3  }
4
```

Execution | Miri | MIR | Close

```
    bb0: {
        StorageLive(_2);                 // bb0[0]: scope 0 at src/lib.rs:2:5: 2:8
        _2 = _1;                         // bb0[1]: scope 0 at src/lib.rs:2:5: 2:8
        StorageLive(_3);                 // bb0[2]: scope 0 at src/lib.rs:2:11: 2:14
        _3 = _1;                         // bb0[3]: scope 0 at src/lib.rs:2:11: 2:14
        _4 = CheckedMul(move _2, move _3); // bb0[4]: scope 0 at src/lib.rs:2:5: 2:14
        assert(!move (_4.1: bool), "attempt to multiply with overflow") -> bb1; // bb0[5]: scope 0 at src/lib.rs:2:5: 2:14
    }

    bb1: {
        _0 = move (_4.0: i32);           // bb1[0]: scope 0 at src/lib.rs:2:5: 2:14
        StorageDead(_3);                 // bb1[1]: scope 0 at src/lib.rs:2:13: 2:14
        StorageDead(_2);                 // bb1[2]: scope 0 at src/lib.rs:2:13: 2:14
        return;                          // bb1[3]: scope 0 at src/lib.rs:3:2: 3:2
    }
}
```

Compiler Explorer

COMPILER EXPLORER

Add... More

Get cool gear in the Compiler Explorer shop ×

Share Other Policies

Rust source #1 ×

A Save/Load + Add new... Rust

```
1  // Type your code here, or load an example.
2  pub fn square(num: i32) -> i32 {
3      num * num
4  }
5
```

rustc 1.31.0 (Editor #1, Compiler #1) Rust ×

rustc 1.31.0 Compiler options...

A 11010 .LX0: .text // \s+ Intel Demangle Libraries

+ Add new... Add tool...

```
1  example::square:
2          push    rax
3          imul    edi, edi
4          seto    al
5          test    al, 1
6          mov     dword ptr [rsp + 4], edi
7          jne     .LBB0_2
8          mov     eax, dword ptr [rsp + 4]
9          pop     rcx
10         ret
11 .LBB0_2:
12         lea     rdi, [rip + .L__unnamed_1]
13         mov     rax, qword ptr [rip + core::panicking::p
14         call    rax
15         ud2
16
17 str.0:
18         .ascii  "/tmp/compiler-explorer-compiler1181114-
19
20 str.1:
21         .ascii  "attempt to multiply with overflow"
22
23 .L__unnamed_1:
24         .quad   str.1
25         .quad   33
26         .quad   str.0
27         .quad   65
28         .long   3
29         .long   5
```

Output (0/0) rustc 1.31.0 - *cached (3629B)*

godbolt.org

Lessons Learned

- Rust allows for lots of syntactic sugar
- It's good to be reminded about that sometimes
- Tools help us understand what's going on behind the curtains.

Now go and build cool things!

mre / cargo-inspect

Unwatch 4 | Star 116 | Fork 3

Code | Issues 7 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Filters | is:issue is:open | Labels | Milestones | New issue

7 Open | 7 Closed | Author | Labels | Projects | Milestones | Assignee | Sort

- **Add flag to change colorscheme** good first issue help wanted
  #17 opened Feb 2, 2019 by mre
- **Write an IntelliJ plugin for cargo-inspect** good first issue help wanted
  #15 opened Feb 2, 2019 by mre
- **Use a random spinner while compiling** 5
  #13 opened Dec 6, 2018 by mre
- **Add support for diffing two programs** 1
  #11 opened Dec 1, 2018 by mre
- **Support line ranges** good first issue help wanted
  #10 opened Nov 30, 2018 by mre
- **Add support for not expanding macros** good first issue help wanted
  #8 opened Nov 19, 2018 by mre
- **Add links to documentation** good first issue help wanted
  #7 opened Nov 19, 2018 by mre

Credits

- Stage background from freepik.com designed by starline
- Lucy with a Rocket engine
- Rustlang MIR documentation
- Rust compiler guide