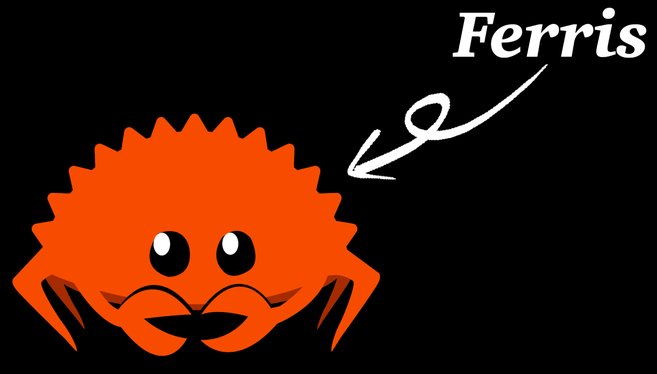
A perspective on Rust

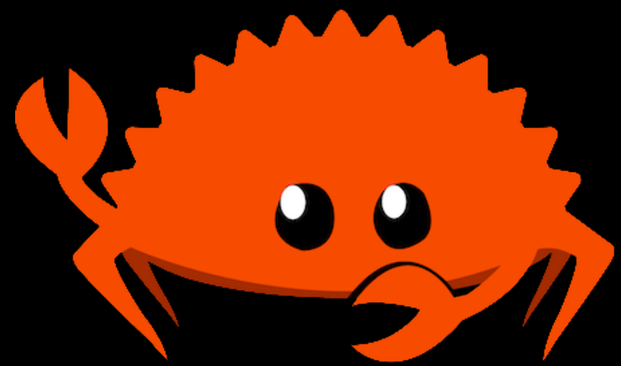
In Rust We Trust?

whoami

- Kotlin Fullstack Engineer by day (Thank you Thermondo)
- Rust dilettante in my free time and in some PRs reviewer
- Organizer in Berlin for: KUG & Rust & XTC & GDG Android
- Koin Champion but nobody knows it



whoami



What to expect?

- Just an short overview / No space rocket
- Key concepts
- A Rust Code breeze
- Some opinions
- Overview over the environment
- I try to stay in 30 min

Let’s make a short therapy session?!

THIS PAGE
INTENTIONALLY
LEFT BLANK

What is Rust?

What is Rust?

```
impl Deck {
    pub fn shuffle_encrypt<Rng: CryptoRng + RngCore>(
        &self,
        rsa_parameter: &RsaParameter,
        rng: &mut Rng,
    ) -> (EncryptedDeck, Rsa) {
        let deck: EncryptedDeck = self.to_encrypted_deck();
        deck.shuffle_encrypt(rsa_parameter, rng)
    }

    pub fn is_shuffle_encrypt_valid(&self, key: &Rsa, encrypted_deck: &EncryptedDeck)
    {
        let deck: EncryptedDeck = self.to_encrypted_deck();
        deck.is_shuffle_encrypt_valid(key, encrypted_deck)
    }
}
```

What is Rust?

History lesson:

2006: pet project of Graydon Hoare (Mozilla) [as a OOP Language]

2015: first stable version

2020: Corona caused the Rust Foundation

2021: First announcement of Google that the Android Project will support Rust

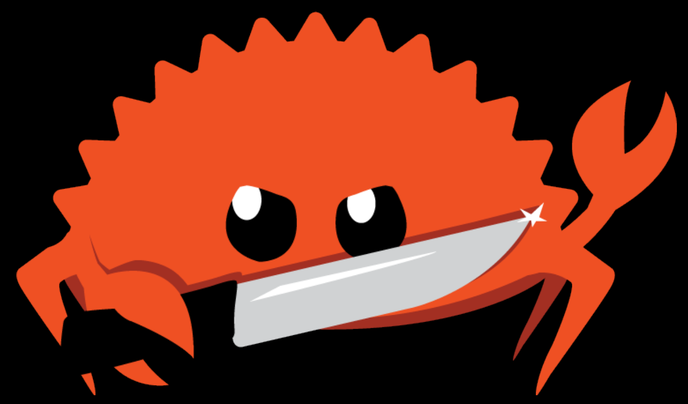
Till 2024: Rust is backed up by several big Corps (incl. MS and Google) and gets adopted by a steady growing number of projects (ICU-X, Linux, ...)

2030: Rust takes over the world

What is Rust?

There was/is a time where with:

- Pseudo types
- Cryptic compiler messages
- Without a integrated build environment
- Where tests are optional



What is Rust?

Rust is a Language Goals are:

1. Modern syntax
2. Incremental replacement of C/C++
3. Zero abstraction overhead, while being memorysafe (means no garbage collector)
4. Fearless Concurrency
5. …

What is Rust?

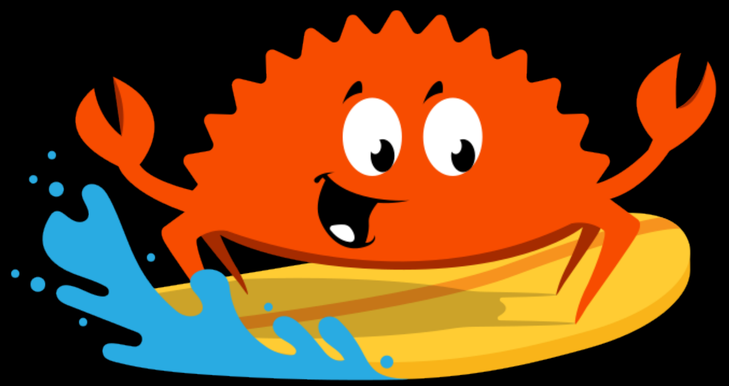
Rust is a Language with:

1. concise and compact (but still somehow complex)
2. Strongly typed
3. Has some multi paradigm capabilities but is always very opinionated
4. Low-level implementation but a high-level attitude
5. ...

What is Rust?

Rust enables development for:

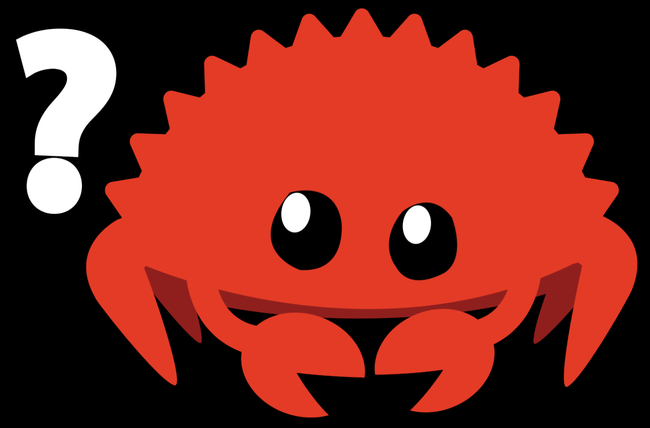
… almost everything aside from your toothbrush



How to interop with Rust?

Depending on the target:

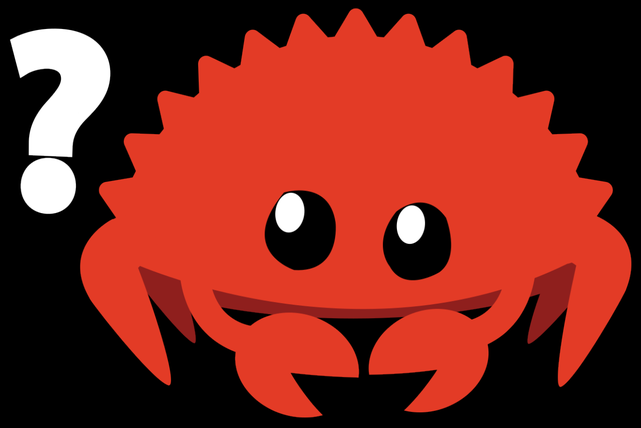
- Generators like UniFFI, Flutter_Bridge 🤝
- by Hand 🤡
- It is not need 💪



Key Concepts? - Ownership

References can:

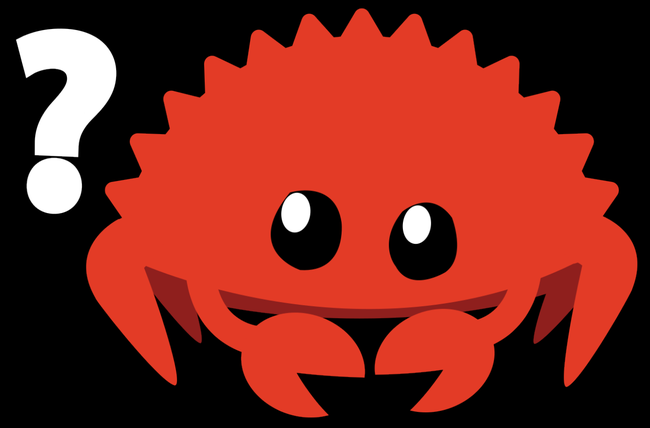
- Is owned by one (Highlander principal)
- Explicit move/borrow/copy semantic



Key Concepts? - Lifetimes

References have:

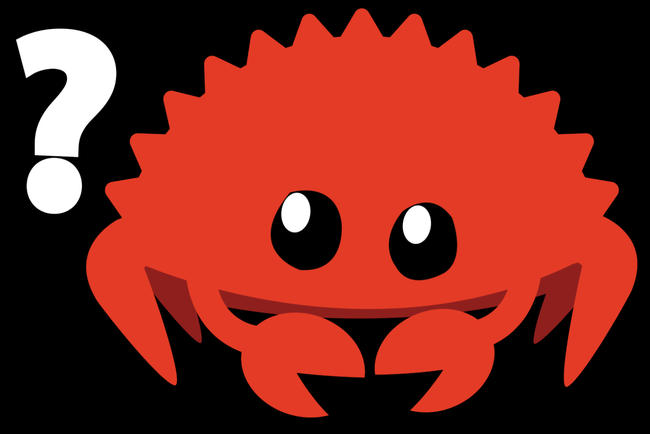
- a lifetime (explicit or implicit)
- a boundary (aka scope)



Key Concepts? - The Borrow Checker

Reference is:

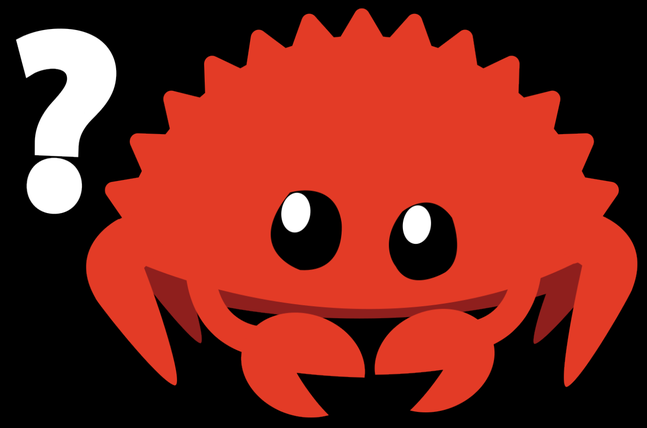
- tracked by the Borrow Checker (!mutability)
- Is (de-)allocated by the compiler (thx to the Borrow Checker)



Key Concepts? - The Borrow Checker

The Borrow Checker is:

- Dumb
- A source of pain
- A safeguard of race conditions
- …



How does Rust looks - Variables

```rust
let deck = Deck::new(); // immutable
let mut deck2 = Deck::new(); // mutable
const abc: i32 = 32;
static text: &str = "abc";
```

How does Rust looks - Functions

```
fn start(players: 3) -> Game {
    Self {
        latest_deck: None,
        published_keys: Default::default(),
    }
}
```

How does Rust looks - Lambdas

```rust
let lambda = |i: i32| -> i32 { i + 1 };
```

How does Rust looks - Structs

```
struct Game {
    latest_deck: Option<EncryptedDeck>,
    turn: u32,
}
```

How does Rust looks - Traits

```
trait GameRound {
    fn deal(&self);
}
```

How does Rust looks - Traits

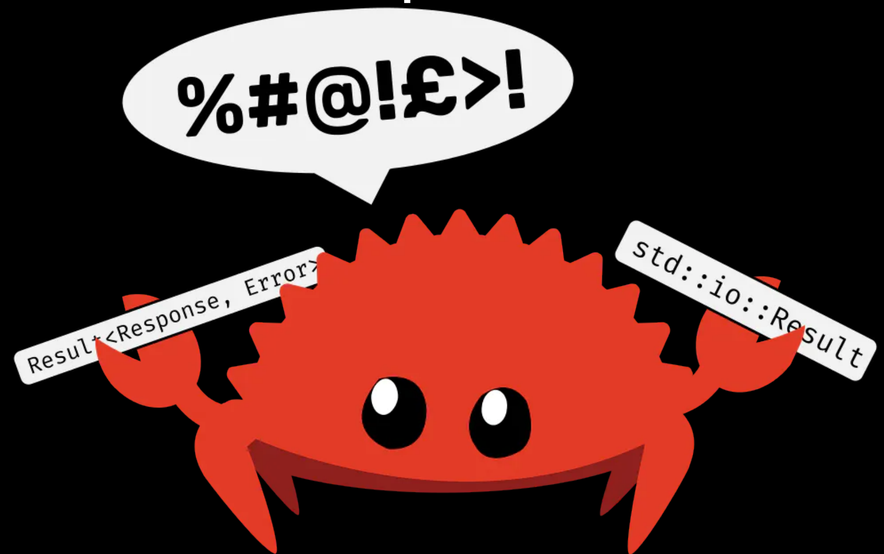
```
impl GameRound for Game {
    fn deal(&self) {
        ...
    }
}
```

What is about Tests in Rust?

- Build-In feature
- 3 types by default - Unit Test, Integration Test, DocTests
- UnitTest are tightly coupled (even in the same file)
- IntegrationTests are separated by structure (test folder)
- DocTest are tightly coupled to the code it is covered

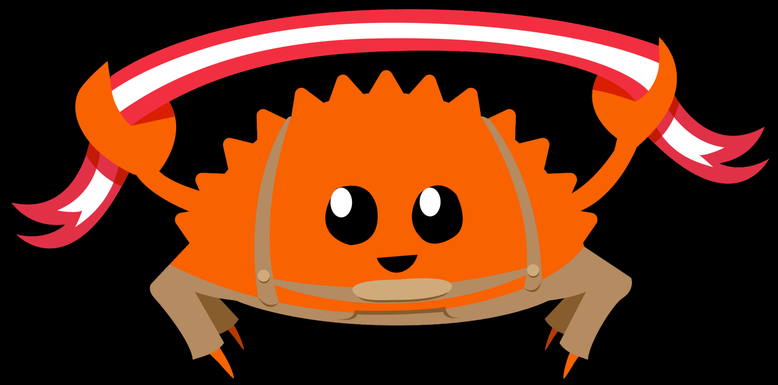
What is about Auxiliary Language Tools?

- Supports Metaprogramming (Macros/Preprocessor)
- Macros have there own small language on top
- Macros let you manipulate the existing Source Code on compile time



What is about other cool Rust stuff?

- You get monads
- You get pattern matching
- You UTF-8 support build-in
- You get Smart Pointer
- …



What is about Build Tools?

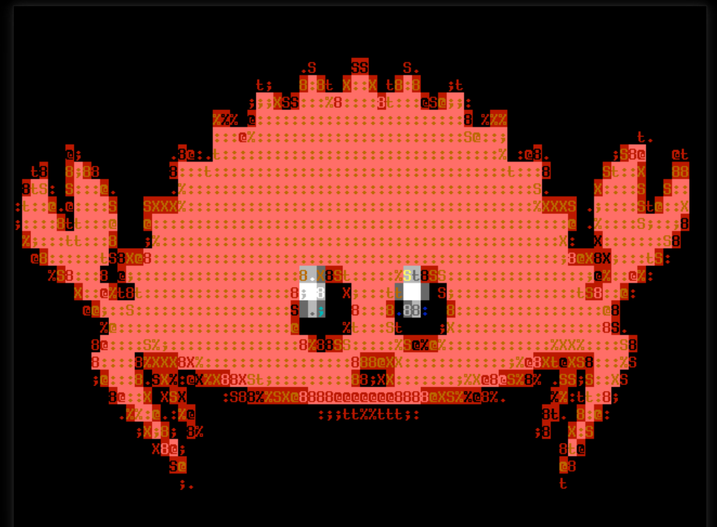
- Rust comes with its own build tool – cargo
- cargo gives you in essence everything what other Tools (like Gradle) give you, while it is only supporting Rust
- cargo supports also things like code coverage, modularisation,…

What is about the Community?

- Just great
- Rust & Tell
- join us (Berlin) online for our Rust Hack & Learn

Helpers

- Rust by Example
- Comprehensive Rust (by Google)
- Rust CookBook
- Rust Web Development (a Berlin Person)
- Your local community (if there is non, make one)



Summery

- Rust is super opinionated about almost everything
- Rust ecosystem is rich and gets richer every day
- Rust community is just awesome and very inclusive
- Rust can be a valley of pain
- Rust is a learning curve

Feedback

Done?!

Thank you!!!!!

Questions?