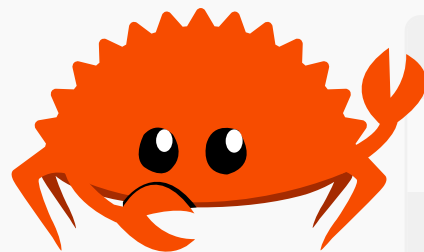


Rust

A boring and expressive language

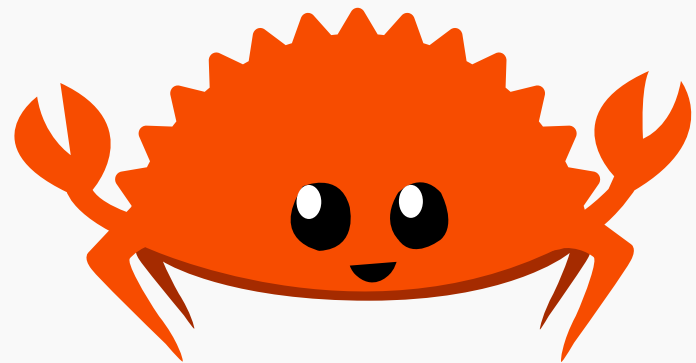
Victor Diez Ruiz



```
fn main() {  
    println!("Hello world 🦀");  
}
```

Why Rust rocks

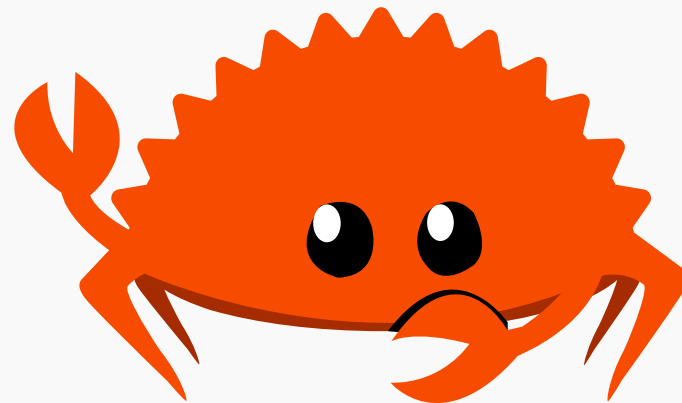
1. Lifetimes & Ownership
2. Immutability by default
3. Algebraic Data Types
4. Pattern Matching
5. Traits
6. Macros



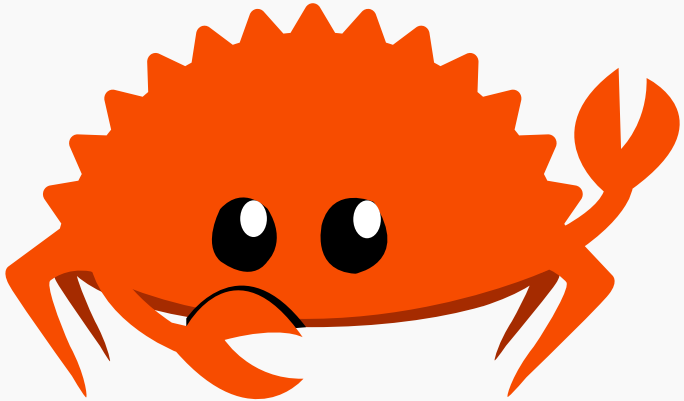
Lifetimes & Ownership

Scopes

```
fn main() {  
    let a = 2;  
    let b = 3;  
    println!("{}", a + b);  
}
```

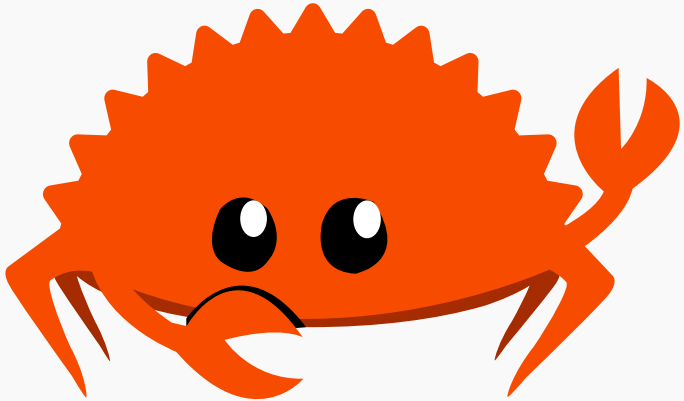


Lifetimes



```
fn main() {  
    let a = 2;  
    { let b = 3; }  
    println!("{}", a + b);  
}
```

Lifetimes

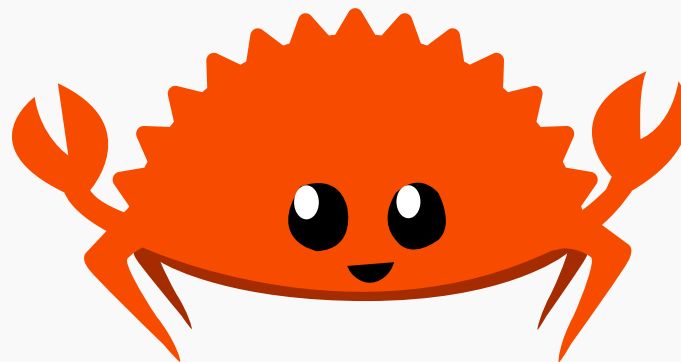


```
fn main() {  
    let a = 2;  
    { let b = 3; }  
    println!("{}", a + b);  
}
```

Ownership

```
fn main() {  
    let a = 2;  
    let b = 3;  
    println!("{}", a + b);  
}
```

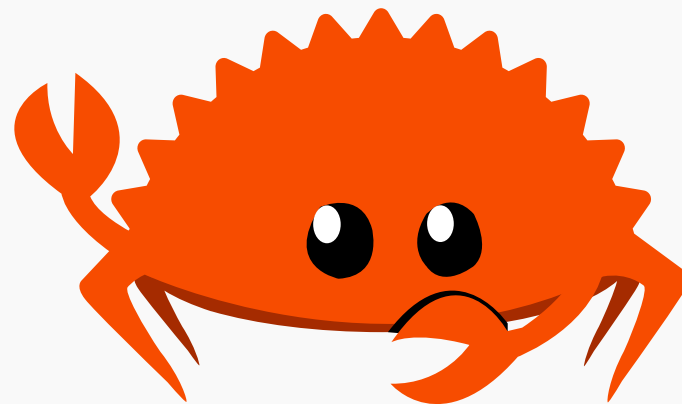
Memory representation
todo



Inmutability by default

Inmutability by default

```
fn main() {  
    let a = 2;  
    let mut b = 3;  
  
    a = 3; // ✖ error  
    b = 2; // ✔ ok  
}
```



Algebraic Data Types

Algebraic Data Types

product sum power

Pattern Matching

Pattern Matching

deestructurar por deestructurar

Traits

Traits

interfaces pero mucho mejor

Macros



python en rust?!?!?

Something very important