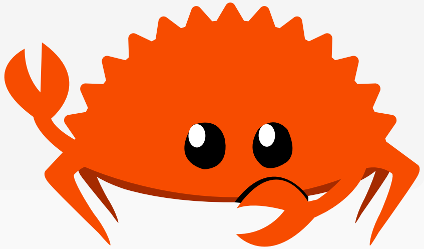
Rust

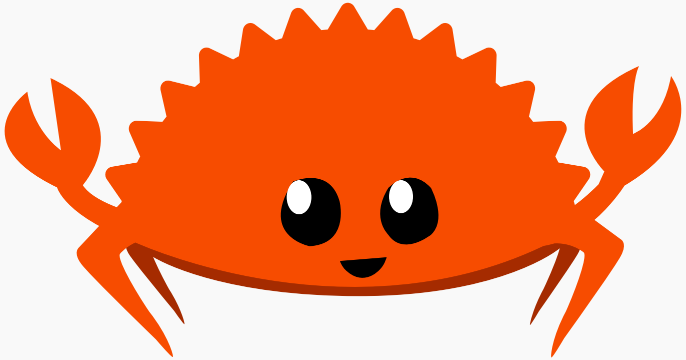
A boring and expressive language

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



Why Rust rocks

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

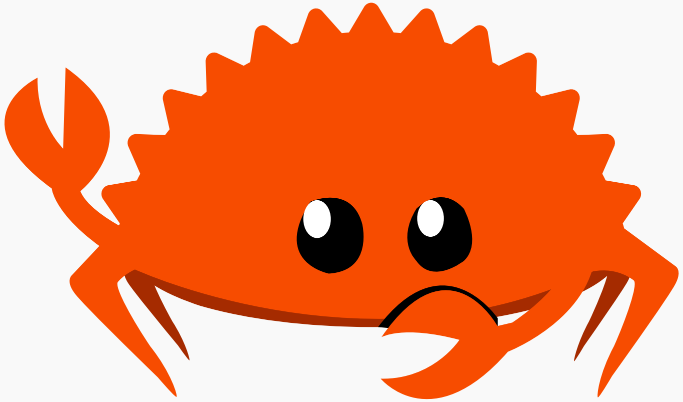


Lifetimes & Ownership

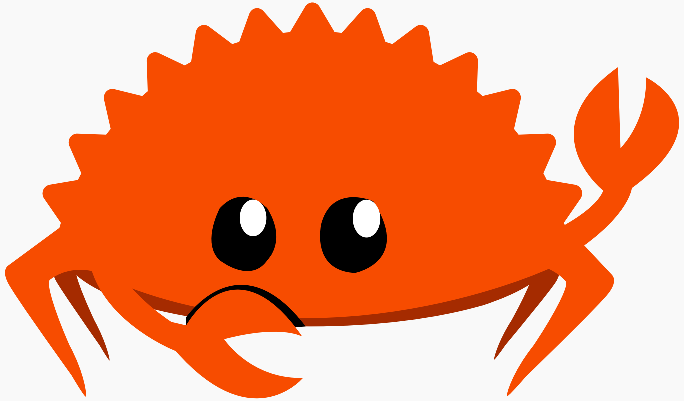
Scopes

```rust
fn main() {
  let a = 2;
  let b = 3;

  println!("{}", a + b);
}
```



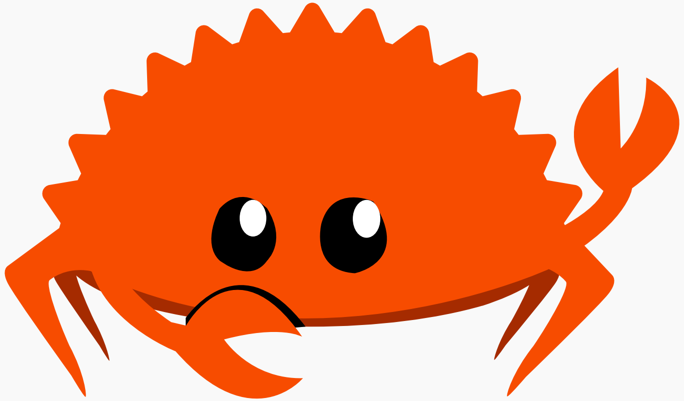
Lifetimes



```rust
fn main() {
  let a = 2;
  { let b = 3; }

  println!("{}", a + b);
}
```

Lifetimes



```rust
fn main() {
  let a = 2;
  { let b = 3; }

  println!("{}", a + b);
}
```

Ownership

```rust
fn main() {
  let a = 2;
  let b = 3;

  println!("{}", a + b);
}
```

M

to

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

destructurar por destructurar

Traits

Traits

interfaces pero mucho mejor

Macros

Macros

python en rust?!?!

Something very important