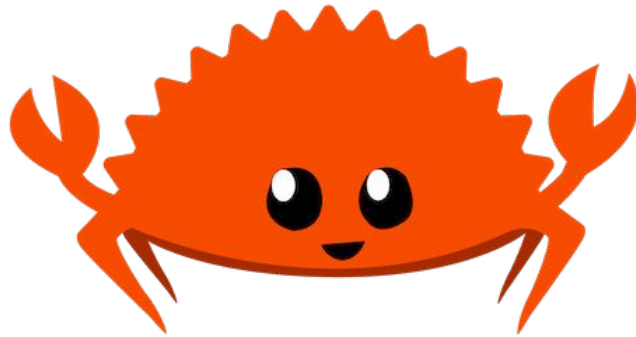


Rust



Some facts

- compiled
- statically typed
- imperative language
- no runtime / no gc

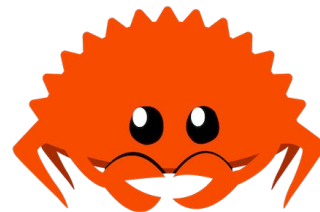


Self-description:

A language empowering everyone to build reliable and efficient software.

Use cases?

- Web-servers / Networking / IO-heavy! <https://www.arewewebyet.org/>
- WebAssembly!
- Kernels! <https://www.kernel.org/doc/html/next/rust/index.html>
- Embedded! <https://www.rust-lang.org/what/embedded>
- Games! <https://arewegameyet.rs/>
- Data crunching? Maybe. <https://rust-bio.github.io/>
- GUI? - meh... <https://areweguiyet.com/>

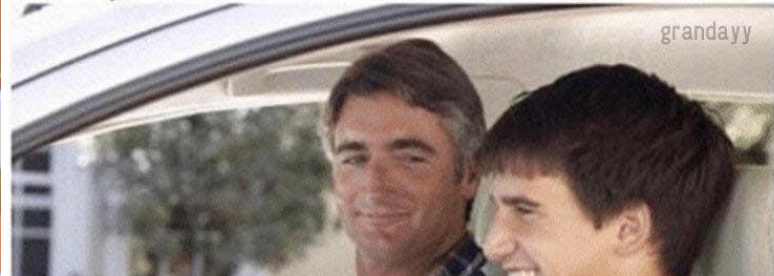


Rust EVANGELISM STRIKE FORCE



"Dad why is my sisters name Rose"
"Because your mother loves Roses"
"Thanks Dad"
"No problem Rust enums"

grandayy



BEWARE OF THE PIPELINE



Ownership & Borrowing

```
fn main() {  
    let s1 = String::from("hello");  
    let (s2, len) = calculate_length(s1);  
  
    println!("The length of '{}' is {}.", s2, len);  
}  
  
fn calculate_length(s: String) -> (String, usize) {  
    let length = s.len();  
    (s, length)  
}
```

Ownership & Borrowing

```
fn main() {  
    let s1 = String::from("hello");  
    let len = calculate_length(&s1);  
  
    println!("The length of '{}' is {}.", s1, len);  
}  
  
fn calculate_length(s: &String) -> usize {  
    s.len()  
}
```

Ownership & Borrowing

```
fn main() {  
    let mut s = String::from("hello");  
  
    change(&mut s);  
}  
  
fn change(some_string: &mut String) {  
    some_string.push_str(", world");  
}
```

Links and Resources

- The BOOK: <https://doc.rust-lang.org/book/>
- The Rustlings course: <https://github.com/rust-lang/rustlings/>
- Rust by example: <https://doc.rust-lang.org/rust-by-example/>
- A Gentle Introduction to Rust:
<https://stevedonovan.github.io/rust-gentle-intro/>
- Comprehensive Rust (by google):
<https://google.github.io/comprehensive-rust/>

Getting Rust: <https://rustup.rs/>