

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Rust 101

Deploy, Write, and Improve Rust

E. Dunham
@qedunham

Feb 26, 2019

<http://talks.edunham.net/osulug>

- 1 Intro
 - About Me
 - About Rust
- 2 Run Rust
 - Editions
 - Channels
 - Installation Options
- 3 Write Rust
 - Errors
 - Syntax
 - Types & Traits
 - Safety
 - Using Libraries
- 4 Improve Rust
 - Level up
 - Find a project
 - Get involved

Welcome!

- Learn what Rust is & isn't
- Run Rust code
- Meet Rust's special features
- Improve Rust

Not:

- Hack time for you
- Unsafe or advanced Rust

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

This talk will teach you the basics of Rust, and help you ask the right questions as you continue to explore Rust in the future.

About Me:

- CloudOps at Mozilla
- Former DevOps for Mozilla Research ¹
- Former Rust Community Team member
- FOSS & robotics background
- NOT a compiler wizard

¹Infra is mostly Python

What's Rust?

- Systems language ²
- Safety + Performance
- Community (intolerant of intolerance)

²Contrast to Go, a language for sysadmins

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

Rust's Buzzwords

- Safety, Speed, Concurrency
- Memory safety without garbage collection
- Zero-cost abstractions

Safety & GC

- Memory must be reused
- C: “Just follow these rules perfectly, you’re smart”
- Java, JS, etc: “Wait a minute, I’ll take care of it”
- Rust: “I’ll prove correctness at compile time”

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

History

- Since ~2010
- 1.0 Stable in May 2015
- Currently ³ version 1.14.0
- Mozilla sponsorship & support

³until January 26th

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

Notable Projects

- `servo.org` Browser Engine
- `habitat.sh` Infrastructure Tooling
- Dropbox (internal use)
- <https://www.rust-lang.org/en-US/friends.html>

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

Where is Rust a good tool?

- Speed + Safety essential
- LLVM-supported architecture
- Team ♥ new technology

Where might Rust be a bad tool?

- Timeframe prohibits new learning
- Need code reuse ⁴
- Can't handle CoC

⁴Corrode can translate C to unsafe Rust

Rust 101

E. Dunham
@qedunham

Intro

About Me

About Rust

Run Rust

Write Rust

Improve Rust

Questions about Rust's place in the world?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Editions

Channels

Installation Options

Write Rust

Improve Rust

- 1 Intro
 - About Me
 - About Rust
- 2 Run Rust
 - Editions
 - Channels
 - Installation Options
- 3 Write Rust
 - Errors
 - Syntax
 - Types & Traits
 - Safety
 - Using Libraries
- 4 Improve Rust
 - Level up
 - Find a project
 - Get involved

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Editions

Channels

Installation Options

Write Rust

Improve Rust

- Opportunities to break backwards compatibility
- opt-in
- “use” replaces “extern crate” in 2018 edition

Rust's channels:

- Nightly: Trying neat ideas
- Beta: Release candidates
- Stable: Always backwards-compatible within edition ⁵

⁵<https://blog.rust-lang.org/2015/05/15/Rust-1.0.html>

Which channel to use?

- Stable → nightly beats nightly → stable
- Dependencies may require nightly
- New project? Pick stable ⁶

⁶Unless you need an unstable feature

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Editions

Channels

Installation Options

Write Rust

Improve Rust

Testing Stable with Crater

- Compile all published libraries
- Diff results from stable and candidate
- <https://github.com/rust-lang-nursery/crater>

Installation:

- Online: `play.integer32.com` or `play.rust-lang.org`
- Many rusts: `rustup.rs`
- System package manager ⁷
- Tinfoil hat: Compile from source

⁷Or add Rust to your favorite!

Playpens:

- Source at <https://github.com/rust-lang/rust-playpen>
- Choose output, LLVM IR, or ASM
- Gist your progress
- Config alters editor settings

Rustup:

- `https://www.rust-lang.org/tools/install`
- Docs at `https://github.com/rust-lang-nursery/rustup.rs`
- `~/.cargo/bin`
- *rustup install nightly*
- *rustup run nightly cargo new*

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Editions

Channels

Installation Options

Write Rust

Improve Rust

Security:

- All releases GPG signed
- Key only held by Rust Core Team members
- keybase.io/rust (proved by local signing)
- Rustup checks signatures for you
- github.com/rust-lang/rust

Deployment

- Check README for system dependencies
- *git clone projecturl*
- *cd project*
- *cargo run*
 - Downloads any dependencies
 - Compiles deps & code
 - Executes *src/main.rs*

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Editions

Channels

Installation Options

Write Rust

Improve Rust

Questions about channels & installation?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

- 1 Intro
 - About Me
 - About Rust
- 2 Run Rust
 - Editions
 - Channels
 - Installation Options
- 3 Write Rust
 - Errors
 - Syntax
 - Types & Traits
 - Safety
 - Using Libraries
- 4 Improve Rust
 - Level up
 - Find a project
 - Get involved

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

IDE support:

- <https://areweideyet.com/>
- <http://www.jonathanturner.org/2017/01/rls-alpha-release.html>

Read-Evaluate-Print Loop

- RFC: <https://github.com/rust-lang/rfcs/issues/655>
- Use the playpen
- <https://repl.it/languages/rust>
- https://crates.io/crates/evcxr_repl

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Rust wants you to succeed.

- Rules catch things that look unsafe
- “Unsafe” directive is an override
- Errors deserve helpful docs
- Click error number in playpen!

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Erroneous Errors?

- Search the web!
- Gist your code
- Ask on IRC #rust-beginners
- File a bug

Aside: Other helpful tools

- <https://github.com/nrc/rustfmt>
- Rustfmt standardizes style for you
- <https://github.com/manishearth/rust-clippy>
- Clippy gives helpful suggestions

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Questions about errors?

Scope Syntax:

- Everything between matched `{}`
- Scopes can nest
- `{Outer Scope {Inner Scope}}`
- Pay attention to a value's scope!

Function Syntax:

- `fn myfunction { ... }`
- `fn myfunction (arg: type, arg: type) - } resulttype { ... }`
- Type signatures are like Mad Libs
- Function has at least name and scope

Macro Syntax:

- Shorthand for functions with variable number of arguments
- *macroname!(foo, bar, baz)*
- `doc.rust-lang.org/beta/book/macros.html`
- You'll see "println!"

A Function:

```
fn halve(x: i32) -> i32 {  
    return x / 2;  
}  
  
fn main(){  
    println!("{}", halve(4));  
}
```

Punctuation Matters:

- Expressions end with a semicolon
- Bare expression, last thing in function body, returns its result
- Spaces separate tokens: `i32` is not `i 32`
- Whitespace is mostly irrelevant

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

```
fn halve
(x          :
i32)->i32
{
return
x
      /
      2

; } fn main
()
{ println!("{}", halve
(
4
));}
```

Control Flow Syntax:

- Conditionals and loops are familiar
 - *if* x { ... }
 - *loop* { ... }
 - *while* x { ... }
 - *for* x in *1..100* { ... }
- Match statements combine conditionals
- <https://doc.rust-lang.org/book/if.html>
- <https://doc.rust-lang.org/book/loops.html>
- <https://doc.rust-lang.org/book/match.html>

Matching on a variable:

```
fn main() {  
    let day = 5;  
    println!("March {} 2017 was:", day);  
    match day {  
        2 => println!("Open Infra Day"),  
        1 | 6 => println!("Travel Time"),  
        3...5 => println!("The Conference"),  
        _ => println!("not SCaLE at all"),  
    }  
}
```

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety


Using Libraries

Improve Rust

Questions about basic syntax?

Why Types & Traits?

- Describe characteristics of inputs and outputs
- Avoid allocating unneeded memory
- Remind humans how code works ⁸

⁸Even type signatures that the compiler could infer must be spelled out. 

Built-in types

- <https://doc.rust-lang.org/book/primitive-types.html>
- Primitives, arrays, strings, tuples

Custom types

- http://rustbyexample.com/custom_types.html
- Structs & Enums
- Use types from your dependencies (example in a few slides)

Traits

- <https://doc.rust-lang.org/book/traits.html>
- Traits describe a type's abilities
- You can tell Rust how type has trait with "impl"
- Generalize function's input and output

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Questions about types & traits?

Ownership:

- `let myint = 42;`
- “myint” is a variable binding
- “myint” **owns** the value 42
- Every value has exactly one owner
- See <https://doc.rust-lang.org/book/ownership.html>

Mutability:

- Owner can only change value if it's **mutable**
- *let mut myint = 42;*

Aside: Copy types

- Primitives have **Copy** trait
- Contents duplicated on assignment to new var
- Custom and other types are not **Copy**
- We'll use *Box*

Changing the owner:

```
fn main() {  
    let first = Box::new(42);  
    // Box forces heap allocation,  
    // regular int would just copy  
    println!("{}", first);  
  
    let second = first;  
    println!("{}", second);  
  
    // this would be an error:  
    // println!("{}", first);  
}
```

Borrowing

- Grant temporary access to a value
- 1 mutable borrow XOR unlimited immutable borrows
- Syntax: `&myvar`
- <https://doc.rust-lang.org/book/references-and-borrowing.html>

Lifetimes

- Remember {scopes}?
- Variables disappear when their scope ends!
- No borrow may outlive its value's owner.

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Questions about safety?

Package Management

- Package manager: Cargo
- Libraries: Crates
- Package index: crates.io
- See “Package Managers All The Way Down”, Tasman B, 3:40pm today
- See <http://doc.crates.io/guide.html>

Create a binary or library

- *cargo new -bin*
 - Binary
 - You run main.rs
 - cargo.lock stores state of last good build
- *cargo new*
 - Library
 - src/lib.rs, no main function

'cargo new' creates:

- Cargo.toml
- src/main.rs or src/lib.rs
- .git if absent

Depending on a crate

- Search crates.io
- Search the web, check recent blogs
- Check docs, license, & project policies
- Add it to 'dependencies' section of cargo.toml
 - *name = "0.1"*
 - *name = { git = "https://github.com/org/repo.git" rev = "123abcd" }*
- *extern crate rand; use rand::Rng;*

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Errors

Syntax

Types & Traits

Safety

Using Libraries

Improve Rust

Questions about libraries?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

- 1 Intro
 - About Me
 - About Rust
- 2 Run Rust
 - Editions
 - Channels
 - Installation Options
- 3 Write Rust
 - Errors
 - Syntax
 - Types & Traits
 - Safety
 - Using Libraries
- 4 Improve Rust
 - Level up
 - Find a project
 - Get involved

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Always...

- Respect others' licenses
- License your own code
- Document & share what you learn!

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Read a Book

- <https://doc.rust-lang.org/stable/book/>
- <https://doc.rust-lang.org/nomicon/>
- O'Reilly Book coming soon

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project
Get involved

Follow the News

- <http://www.newrustacean.com/> podcast
- <https://soundcloud.com/posix4e/sets/rustyradio> interviews
- <https://this-week-in-rust.org/> Weekly newsletter
- <https://blog.rust-lang.org/> Official Blog

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project
Get involved

Practice

- <https://github.com/carols10cents/rustlings>
- <http://rustbyexample.com/>

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Questions about learning more Rust?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Join a project

- <https://crates.io/>, find popular crates
- Search GitHub “is:issue label:easy language:rust”

Port something

- <https://github.com/jameysharp/corrode>
- <https://blog.rust-lang.org/2015/04/24/Rust-Once-Run-Everywhere.html>

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Questions about finding a project?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

File or Fix issues

- github.com/rust-lang/rust
- Triaged regularly
- If in doubt, ask on IRC first

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Chat Online

- IRC: #rust, #rust-beginners on irc.mozilla.org
- users.rust-lang.org Users Forum
- Reddit, StackOverflow, etc.

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Find or join a meetup

- Search for your area + Rust meetup
- Portland might be nearest
- Community Team Calendar, goo.gl/EJ2iRb
- #rust-community on Mozilla IRC

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Attend a conference

- rustconf.com Oregon, September
- www.rust-belt-rust.com, Pennsylvania, October
- www.rustfest.eu, Europe, month varies

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Questions about getting involved?

Rust 101

E. Dunham
@qedunham

Intro

Run Rust

Write Rust

Improve Rust

Level up

Find a project

Get involved

Thank you!