Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

Intro to Rust

Florian “Florob” Zeitz

2025-07-02

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

What is Rust

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- systems programming language
- compiled
- strongly, statically typed
- low-level access, high-level abstractions
- "curly-braced" (C-like)
- large community
- multi-paradigm
- inspired by: C++, Erlang, Haskell, OCaml, Swift, ...

History (up to 1.0)

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

from 2006	personal project of Graydon Hoare compiler in OCaml
since 2009	development supported by Mozilla, as part of Mozilla Research
since 2011	self-hosting
since 2014	language changes through RFC process
May 2015	release of Rust 1.0

History (after 1.0)

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

2019	async-await syntax and futures
2020	Mozilla layoffs affecting Rust
since 2021	Rust Foundation
2018, 2021, 2024	new Rust “editions”

Philosophy

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- memory-safe
  - no use-after-free
  - no out-of-bounds accesses
- make data races impossible
- no runtime or mandatory garbage collector
- make costs explicit
- enforce handling error conditions
- immutable by default
- zero-cost abstractions

Fields of Application

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- CLI tools
- Web Services
- WebAssembly
- Embedded Systems
- ...

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features
- Enums
- Pattern Matching
- Iterators

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features
- Enums
- Pattern Matching
- Iterators

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

Enums

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
// C-like
enum Dir {
    North,
    East,
    South,
    West
}
let d: Dir = Dir::East;
```

```rust
// with associated data
enum Shape {
    Rect { x: f32, y: f32 },
    Circle { r: f32 }
}
let c: Shape = Shape::Circle {
    r: 23.0
};
```

- sum type
- similar to a tagged union

Example: Option


```
1 enum Option<T> {
2     None,
3     Some(T)
4 }
```

- equivalent to Haskell's Maybe monad (`return = Some`, `bind = and_then`)
- used instead of NULL-pointers, nil-objects, etc.

Example: Option

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
let entity = "#x21";
let val = entity
    .strip_prefix("#x")
    .and_then(|val| u32::from_str_radix(val, 16).ok());
assert_eq!(val.and_then(char::from_u32), Some('!'));
```

What is Rust

Features
Enums
Patterns
Iterators

Ownership & Borrowing
Motivation
Ownership
Borrowing

Data Races
Free-Threaded Python Example
C++ Example
Rust Example

Unsafe Rust

Questions

1 What is Rust

2 Features
- Enums
- Pattern Matching
- Iterators

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

match

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
let x = 2u32;

match x {
    1 => "One",
    2 | 3 => "Twree",
    5..9 => "Large small number",
    _ => "Fallthrough"
}
```

match

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
let d = Dir::East;

match d {
    Dir::North => println!("Northwards!")
    Dir::East => println!("Go East!")
    Dir::South => println!("Southwards!")
    Dir::West => println!("Go West!")
}
```

match

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
match x.checked_add(y) {
    Some(res) => println!("{res}"),
    None => println!("Overflow"),
}
```

match

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
let c = Shape::Circle { r: 1.0 };

match c {
    Shape::Rect { x, y } => println!("{x} x {y}"),
    Shape::Circle { r } => println!("{r}"),
}
```

let

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
let Person { age, name } = marv;
let (x, y) = point;
let Person { age: edad, name: nombre } = marv;
```

if let

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
1 if let Ok(dir) = std::env::var("HOME") {
2     println!("Home directory is {}", dir);
3 }
```

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features
- Enums
- Pattern Matching
- Iterators

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

Iterators

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- implemented with an associated function
  `fn next(&mut self) -> Option<Item>`
- `for`-loops are syntactic sugar for repeatedly calling `next()` until it returns `None`
- lots of "adapters" for functional-style programming

Example: Iterator Adaptors

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
fn main() {
    let mult = (1u32..)
        .filter(|x| (1..11).all(|y| x.is_multiple_of(y)))
        .next()
        .unwrap();
    println!("{}", mult);
}
```

- finds the smallest number evenly divisible by every number from 1 through 10

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing
- Motivation
- Ownership
- Borrowing

4 Data Races

5 Unsafe Rust

Why?

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- systems programming can be scary
- a lot of bugs concern memory safety and data races
- (most) systems language do not protect against them
- Rust's ownership model rules out these classes of bugs

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing
- Motivation
- Ownership
- Borrowing

4 Data Races

5 Unsafe Rust

C++: Realloc

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```cpp
#include <iostream>
#include <vector>
#include <string>

int main() {
    auto v = std::vector<std::string> { "Foo" };

    std::cout << "Capacity: " << v.capacity() << '\n';
    auto const &x = v[0];
    v.emplace_back("Bar");
    std::cout << x << '\n';
}
```

C++: Realloc

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
$ clang++ -Wall -O1 vector.cc -o vector-cc
$ ./vector-cc
Capacity: 1
Segmentation fault (core dumped)
```

C++: Iterator Invalidation

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```cpp
#include <iostream>
#include <string>
#include <vector>

int main() {
    std::vector<std::string> v = { "F", "o", "o" };

    for (auto const &it : v) {
        v.push_back(it + it);
    }
    for (auto const &it : v) {
        std::cout << it << '\n';
    }
}
```

C++: Iterator Invalidation


```
$ clang++ -Wall iter.cc -o iter-cc
$ ./iter-cc
F
o
o
FF


$
```

C++: Use After Free


```cpp
#include <iostream>
#include <memory>

int& f() {
    auto i = std::make_unique<int>(42);
    auto &i_ref = *i;
    return i_ref;
}

int main() {
    int &i = f();
    std::cout << i << '\n';
}
```

C++: Use After Free

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
$ clang++ -Wall after-free.cc -o after-free
$ ./after-free
-662549570
$
```

C++: Type punning

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
3  struct Foo {
4      int field;
5  };
6
7  void f(Foo &foo, float const *x) {
8      auto a = *x + 42.0;
9      foo.field = 0x7fffffff;
10     auto b = *x + 42.0;
11     std::cout << a << ' ' << b << '\n';
12 }
13
14 int main() {
15     Foo foo { 12 };
16     f(foo, reinterpret_cast<float*>(&foo.field));
17 }
```

C++: Type punning

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
$ clang++ -Wall field.cc -o field-cc
$ ./field-cc
42 nan
$ clang++ -Wall -O1 field.cc -o field-cc
$ ./field-cc
42 42
```

Observation

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

Problems arise when combining:

Mutability

```
v.emplace_back(...)
v.push_back(...)
foo.field = 0x7fffffff;
```

+

Aliasing

```
auto const &x = v[0]
auto const &it : v
foo, &foo.field
```

Rust prevents this by ensuring an object is never mutable and aliased.

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing
- Motivation
- Ownership
- Borrowing

4 Data Races

5 Unsafe Rust

Ownership: Bindings

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
1  struct Crop;
2
3  fn main() {
4      let c = Crop;
5
6      // moves c to _miller1
7      let _miller1 = c;
8
9      // error: use of moved value: `c`
10     let _miller2 = c;
11 }
```

Ownership: Functions

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
1  struct Crop;
2  struct Flour;
3
4  fn grind(_c: Crop) -> Flour {
5      Flour
6      // _c is freed here
7  }
8
9  fn main() {
10     let c = Crop;
11
12     grind(c); // c moves into grind()
13     // error: use of moved value: `c`
14     grind(c);
15 }
```

Returning Ownership

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
struct Book { page: u32 }

fn read(b: Book) -> Book {
    println!("I read page {}", b.page);
    b
}

fn main() {
    let b = Book { page: 1 };
    // b moves into `read()`
    let b1 = read(b);
    // error: use of moved value: `b`
    // let b2 = read(b);
    let _b2 = read(b1);
}
```

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing
- Motivation
- Ownership
- Borrowing

4 Data Races

5 Unsafe Rust

Shared Borrow

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
1  struct Book { page: u32 }
2
3  fn read(b: &Book) {
4      println!("I read page {}", b.page);
5  }
6
7  fn main() {
8      let b = Book { page: 1 };
9      let l = &b;
10
11     read(&b);
12     read(l);
13     read(&b);
14 }
```

Mutable Borrow

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
7  fn turn_page(b: &mut Book) { b.page += 1; }
8
9  fn main() {
10     let mut b = Book { page: 1 };
11
12     read(&b);
13     turn_page(&mut b);
14     read(&b);
15
16     let l = &b;
17     // turn_page(&mut b); // error: cannot borrow `b` as
18                           // mutable because it is also
19                           // borrowed as immutable
20     read(l);
21 }
```

Exception: Copy Types

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
struct Dress;
#[derive(Copy, Clone)]
struct Mp3;

fn main() {
    let shop_dress = Dress;
    let _your_dress = shop_dress;
    // error: use of moved value: `shop_dress`
    let _their_dress = shop_dress;

    let shop_mp3 = Mp3;
    let _your_mp3 = shop_mp3;
    // This is fine
    let _their_mp3 = shop_mp3;
}
```

Summary

Intro to Rust

Florob

What is Rust

Features

Enums

Patterns

Iterators

Ownership & Borrowing

Motivation

Ownership

Borrowing

Data Races

Free-Threaded Python Example

C++ Example

Rust Example

Unsafe Rust

Questions

- Ownership: `T`
  - one owner
  - readable
  - mutable†
  - can be moved or borrowed

Summary

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- Ownership: `T`
  - one owner
  - readable
  - mutable†
  - can be moved or borrowed
- Shared borrow: `&T`
  - arbitrarily shareable (aliasing)
  - readable
  - immutable

Summary

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- Ownership: `T`
  - one owner
  - readable
  - mutable†
  - can be moved or borrowed
- Shared borrow: `&T`
  - arbitrarily shareable (aliasing)
  - readable
  - immutable
- Mutable borrow: `&mut T`
  - only one at a time
  - readable
  - mutable

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races
- Free-Threaded Python Example
- C++ Example
- Rust Example

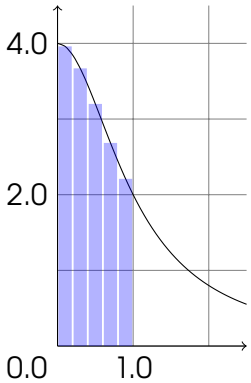
5 Unsafe Rust

Example: Calculating π

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

$$\pi = 4 \arctan(1)$$
$$= \int_0^1 \frac{4}{1+x^2} \, \mathrm{d}x$$

- calculate π by Riemann integration
- approximate the area with thin rectangles
- embarrassingly parallel



What is Rust

Features
Enums
Patterns
Iterators

Ownership & Borrowing
Motivation
Ownership
Borrowing

Data Races
Free-Threaded Python Example
C++ Example
Rust Example

Unsafe Rust

Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races
- Free-Threaded Python Example
- C++ Example
- Rust Example

5 Unsafe Rust

Free-Threaded Python: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```python
from threading import Thread

NUM_THREADS = 4
NUM_STEPS = 100_000_000
THREAD_STEPS = NUM_STEPS // NUM_THREADS
STEP = 1.0 / NUM_STEPS

pi = 0.0

def threadFunc(lower, upper):
    global pi
    for j in range(lower, upper):
        x = (j + 0.5) * STEP
        pi += 4.0 / (1.0 + x * x) * STEP
```

Free-Threaded Python: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
16 threads = []
17 for i in range(NUM_THREADS):
18     lower = THREAD_STEPS * i
19     upper = THREAD_STEPS * (i + 1)
20     t = Thread(target=threadFunc, args=(lower, upper))
21     threads.append(t)
22
23 for t in threads:
24     t.start()
25
26 for t in threads:
27     t.join()
28
29 print(f"Pi = {pi}")
```

Free-Threaded Python: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
$ python3.13t pi.py
Pi = 1.0351948517209917
$ python3.13t pi.py
Pi = 1.0590036387978847
```

- thread A reads `pi = 0.1423`
- thread B reads `pi = 0.1423`
- thread A writes `pi = 0.7609`
- thread B writes `pi = 0.5768`
- `pi = 0.5768`, thread A's calculation is lost
- this is a classical data race

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races
- Free-Threaded Python Example
- C++ Example
- Rust Example

5 Unsafe Rust

C++: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```cpp
#include <cstdint>
#include <print>
#include <thread>
#include <vector>

int main() {
  constexpr uint64_t NUM_THREADS = 4;
  constexpr uint64_t NUM_STEPS = 100'000'000;
  constexpr uint64_t THREAD_STEPS = NUM_STEPS / NUM_THREADS;
  constexpr double STEP = 1.0 / NUM_STEPS;

```

C++: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
12    double pi = 0;
13
14    std::vector<std::thread> threads;
15
16    for (uint64_t i = 0; i < NUM_THREADS; ++i) {
17      uint64_t lower = THREAD_STEPS * i;
18      uint64_t upper = THREAD_STEPS * (i + 1);
19      threads.emplace_back([=, &pi]() {
20        for (uint64_t j = lower; j < upper; ++j) {
21          double x = (j + 0.5) * STEP;
22          pi += 4.0 / (1.0 + x * x) * STEP;
23        }
24      });
25    }
```

C++: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
26
27      for (auto &t : threads)
28        t.join();
29
30      std::println("Pi = {:.10}", pi);
31
32      return 0;
33  }
```

C++: Data Race

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
$ clang++ -Wall -std=c++23 -lpthread pi.cc -o pi-cc
$ ./pi-cc
Pi = 1.156130797
$ ./pi-cc
Pi = 1.099799814
```

- thread A reads `pi = 0.1423`
- thread B reads `pi = 0.1423`
- thread A writes `pi = 0.7609`
- thread B writes `pi = 0.5768`
- `pi = 0.5768`, thread A's calculation is lost
- this is a classical data race

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races
- Free-Threaded Python Example
- C++ Example
- Rust Example

5 Unsafe Rust

A Naïve Port


```rust
use std::thread;

fn main() {
    const NUM_THREADS: u32 = 4;
    const NUM_STEPS: u32 = 100_000_000;
    const THREAD_STEPS: u32 = NUM_STEPS / NUM_THREADS;
    const STEP: f64 = 1.0 / NUM_STEPS as f64;

```

A Naïve Port

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
9       let mut pi: f64 = 0.0;
10
11      let mut guards = Vec::new();
12      for i in 0..NUM_THREADS {
13          let lower: u32 = THREAD_STEPS * i;
14          let upper: u32 = THREAD_STEPS * (i + 1);
15          let pi_ref: &mut f64 = &mut pi;
16          guards.push(thread::spawn(move || {
17              for j in lower..upper {
18                  let x: f64 = (f64::from(j) + 0.5) * STEP;
19                  *pi_ref += 4.0 / (1.0 + x * x) * STEP;
20              }
21          }));
22      }
```

A Naïve Port


```
23
24      for g in guards {
25          g.join().unwrap();
26      }
27
28      println!("Pi = {:.10}", pi);
29  }
```

A Naïve Port

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
error[E0597]: `pi` does not live long enough
  --> pi.rs:15:32
   |
9  |       let mut pi: f64 = 0.0;
   |           ------ binding `pi` declared here
...
15 |           let pi_ref: &mut f64 = &mut pi;
   |                                   ^^^^^^^ borrowed value does not live long enough
16 |           guards.push(thread::spawn(move || {
   |  _____________________-
17 | |             for j in lower..upper {
18 | |                 let x: f64 = (f64::from(j) + 0.5) * STEP;
19 | |                 *pi_ref += 4.0 / (1.0 + x * x) * STEP;
20 | |             }
21 | |         }));
   | |__________- argument requires that `pi` is borrowed for `'static`
...
29 |   }
   |   - `pi` dropped here while still borrowed
```

Spawned threads could live past `main()`.
Therefore, borrowed data needs to live indefinitely.

Adding Some Scope

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
9       let mut pi: f64 = 0.0;
10
11      thread::scope(|scope| {
12          for i in 0..NUM_THREADS {
13              let lower: u32 = THREAD_STEPS * i;
14              let upper: u32 = THREAD_STEPS * (i + 1);
15              let pi_ref: &mut f64 = &mut pi;
16              scope.spawn(move || {
17                  for j in lower..upper {
18                      let x: f64 = (f64::from(j) + 0.5) * STEP;
19                      *pi_ref += 4.0 / (1.0 + x * x) * STEP;
20                  }
21              });
22          }
23      });
```

Adding Some Scope

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
error[E0499]: cannot borrow `pi` as mutable more than once at a time
  --> pi-scoped.rs:15:36
   |
11 |       thread::scope(|scope| {
   |                      ----- has type `&'1 Scope<'1, '_>`
...
15 |               let pi_ref: &mut f64 = &mut pi;
   |                                      ^^^^^^^ `pi` was mutably borrowed here
   |                                              in the previous iteration of the loop
16 | /             scope.spawn(move || {
17 | |                 for j in lower..upper {
18 | |                     let x: f64 = (f64::from(j) + 0.5) * STEP;
19 | |                     *pi_ref += 4.0 / (1.0 + x * x) * STEP;
20 | |                 }
21 | |             });
   | |_____________- argument requires that `pi` is borrowed for `'1`
```

We can only have one mutable borrow at a time, not one per thread.
This effectively makes the data race impossible.

Adding a Mutex

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
10      let pi: Mutex<f64> = Mutex::new(0.0);
11      thread::scope(|scope| {
12          for i in 0..NUM_THREADS {
13              let lower: u32 = THREAD_STEPS * i;
14              let upper: u32 = THREAD_STEPS * (i + 1);
15              let pi_ref: &Mutex<f64> = &pi;
16              scope.spawn(move || {
17                  for j in lower..upper {
18                      let x: f64 = (f64::from(j) + 0.5) * STEP;
19                      *pi_ref.lock().unwrap()
20                          += 4.0 / (1.0 + x * x) * STEP;
21                  }
22              });
23          }
24      });
```

Adding a Mutex

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
25
26      println!("Pi = {:.10}", pi.lock().unwrap());
27  }
```

```
1  $ ./pi-locked
2  Pi = 3.1415926536
```

A Nicer Solution

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```
9       let pi: f64 = thread::scope(|scope| {
10          let mut guards = Vec::new();
11          for i in 0..NUM_THREADS {
12              let lower: u32 = THREAD_STEPS * i;
13              let upper: u32 = THREAD_STEPS * (i + 1);
14              guards.push(scope.spawn(move || {
15                  (lower..upper)
16                      .map(|j| {
17                          let x: f64 = (f64::from(j) + 0.5) * STEP;
18                          4.0 / (1.0 + x * x) * STEP
19                      })
20                      .sum::<f64>()
21              }));
22          }
23
24          guards.into_iter().map(|t| t.join().unwrap()).sum()
25      });
```

What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

1 What is Rust

2 Features

3 Ownership & Borrowing

4 Data Races

5 Unsafe Rust

Unsafe Rust

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- borrowing rules impose restrictions making some things impossible to express
- `unsafe` allows some additional things
  - calling functions marked `unsafe`
  - FFI calls
  - dereference arbitrary pointers
- keeps regular language semantics in place
- used to create safe abstractions

Unsafe Rust

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

```rust
use std::mem::{self, MaybeUninit};

let data = {
    let mut data: [MaybeUninit<Vec<u32>>; 1000] = unsafe {
        MaybeUninit::uninit().assume_init()
    };

    for elem in &mut data[..] {
        elem.write(vec![42]);
    }

    unsafe { mem::transmute::<_, [Vec<u32>; 1000]>(data) }
};
```

Resources

Intro to Rust
Florob
What is Rust
Features
Enums
Patterns
Iterators
Ownership & Borrowing
Motivation
Ownership
Borrowing
Data Races
Free-Threaded Python Example
C++ Example
Rust Example
Unsafe Rust
Questions

- The Book
- Rustlings
- Rust by Example

Thank you for your attention. Any questions?

https://babelmonkeys.de/~florob/talks/RC-2025-07-02-rust-intro.pdf