Yanlin Duan
System Architect

Zhuo Kong
Tester

Emily Meng
Language Guru

Shiyu Qiu
Manager
Okay, human.
Huh?
Before you hit compile, listen up.

You know when you're falling asleep, and you imagine yourself walking or something.

And suddenly you misstep, stumble, and jolt awake?

Yeah!

Well, that's what a segfault feels like.
Double-check your damn pointers, okay?
Project Management

October 9, 2016 - December 18, 2016

To select a time span double click near its start point and drag to its end point.
Features

- Zero-cost abstractions that help catch data races and segfaults
- Heap-allocated arrays and structs provide more flexibility
- Scoping rules that support shadowing
- Statically and strongly-typed system catches errors at compile-time
Syntax

Basic Functionality

```rust
fn main() -> int {
    let mut x:int = 0;
    while (x < 5) {
        println(x);
        x = x + 1;
    }
    return 0;
}
```

Structs

```rust
fn main() -> int{
    struct Point {
        x : int,
        y : int
    }
    let origin : Point = { x : 10, y : 1 };
    println(origin.x + origin.y);
    println("yay");
}
```

Memory Safety

```rust
fn mem1 () -> int {
    let v : int = 42;
    let v2 : &int = &v;
    println(*v2); /* 42 */
    println(v);   /* 42 */
    return 0;
}
```
Testing

- Binary/unary operations
- Control flow
- Functions
- Structs
- Memory safety
Demos