#### HSPACE Rust 특강

## Introduction

Chris Ohk
utilForever@gmail.com

#### 발표자소개

#### • 옥찬호 (Chris Ohk)

- (전) EJN Tech Lead
- (전) Momenti Engine Engineer
- (전) Nexon Korea Game Programmer
- Microsoft Developer Technologies MVP
- C++ Korea Founder & Administrator
- Reinforcement Learning KR Administrator
- IT 전문서 집필 및 번역 다수
  - 게임샐러드로 코드 한 줄 없이 게임 만들기 (2013)
  - 유니티 Shader와 Effect 제작 (2014)
  - 2D 게임 프로그래밍 (2014), 러스트 핵심 노트 (2017)
  - 모던 C++ 입문 (2017), C++ 최적화 (2019)

utilForever@gmail.com





#### 亚자

- 주교재
  - Programming Rust (O'Reilly Media, 2021)
  - Comprehensive Rust (Google, 2023)
  - The Rust Programming Language, 2nd Edition (No Starch Press, 2023)
- 부교재
  - The Rustonomicon (The Rust Language, 2021)
  - Rust for Rustaceans (No Starch Press, 2021)
  - Rust Atomics and Locks (O'Reilly Media, 2023)
  - Effective Rust (O'Reilly Media, 2024)

## 일정

- 진행요일및시간
  - 강의 요일은 토/일요일 오전 11시 시작
  - 강의 시간은 보통 2시간 내외, 최대 3시간
- 참고사항
  - 진행자의 개인 사정에 따라 스터디 일정이 변경될 수 있음

- Week 1 (7/6, 7/7)
  - Part 1
    - Introduction
    - Key Features of Rust
  - Part 2
    - Fundamental Types
    - Ownership and Moves
    - References

- Week 2 (7/13, 7/14)
  - Part 1
    - Expressions
    - Error Handling
    - Crates and Modules
  - Part 2
    - Structs
    - Enums and Patterns

- Week 3 (7/20, 7/21)
  - Part 1
    - Traits and Generics
    - Operator Overloading
  - Part 2
    - Utility Traits
    - Closures

- Week 4 (7/27, 7/28)
  - Part 1
    - Iterators
    - Collections
  - Part 2
    - Strings and Text
    - Input and Output

- Week 5 (8/3, 8/4)
  - Part 1
    - Concurrency
  - Part 2
    - Asynchronous Programming

- Week 6 (8/10, 8/11)
  - Part 1
    - Macros
  - Part 2
    - Unsafe Code
    - Foreign Functions

# Rust Playground

https://play.rust-lang.org/

```
Buse std::sync::{Arc, Mutex};

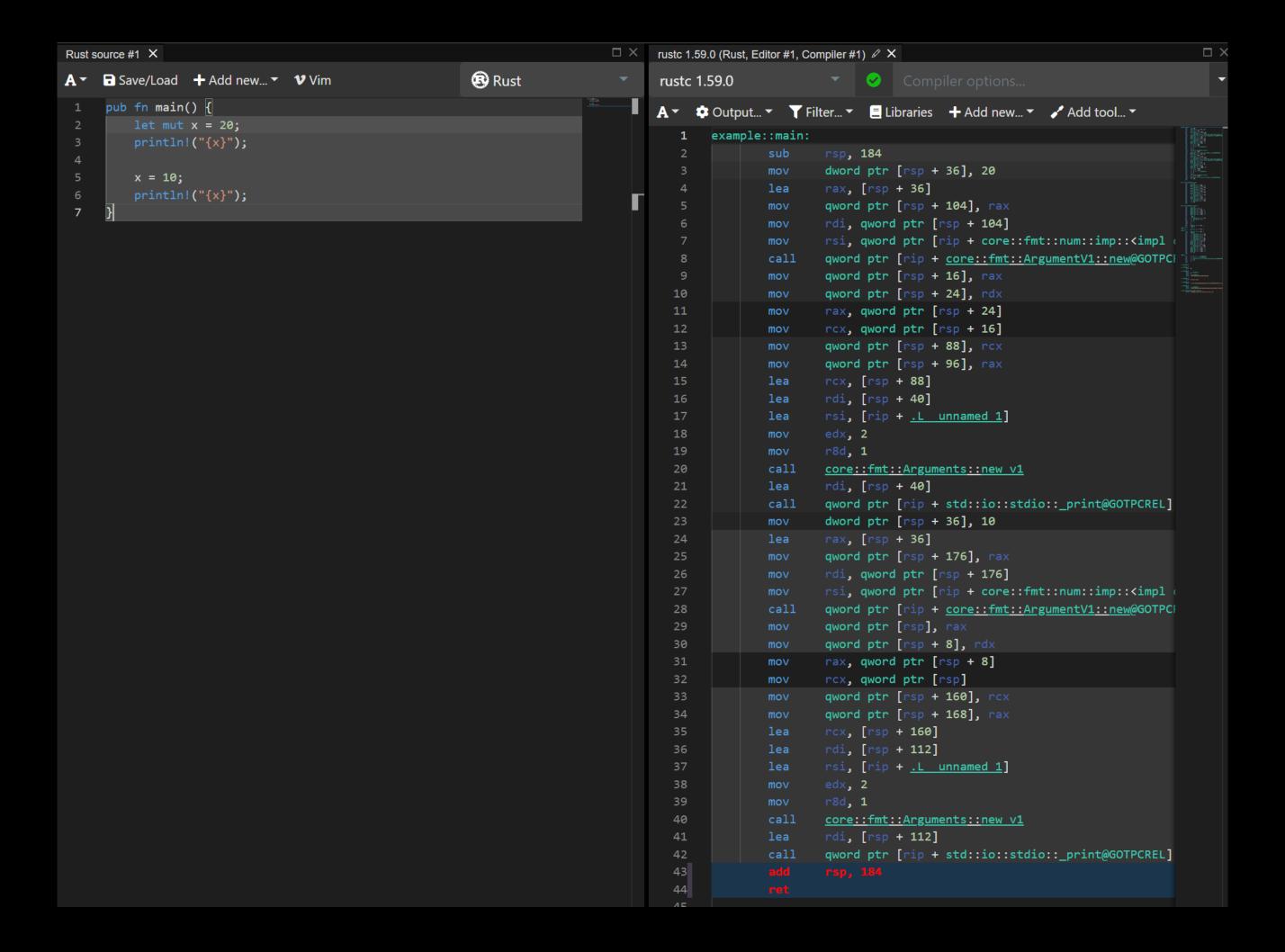
use std::sync::{Arc, Mutex};

fn main() {
    let mut arc_mutex = Arc::new(Mutex::new(()));
    let arc_mutex: &mut Arc<Mutex<()>> = &mut arc_mutex;

    let _guard = arc_mutex.lock().unwrap();
}
```

#### Rust 디스어셈블리

https://rust.godbolt.org/



#### Rust 설치 방법

- Windows
  - 32bit: https://static.rust-lang.org/rustup/dist/i686-pc-windows-msvc/rustup-init.exe
  - 64bit: https://static.rust-lang.org/rustup/dist/x86\_64-pc-windows-msvc/rustup-init.exe
- Windows Subsystem for Linux
  - curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
- Linux and MacOS
  - curl https://sh.rustup.rs -sSf | sh -s -- --help

#### 프로그램 설정

- 여러 프로그램에서 Rust를 사용할 수 있다. 두 프로그램을 많이 사용한다.
  - Visual Studio Code
  - JetBrains RustRover
- Visual Studio Code와 같이 사용하면 좋은 Extensions
  - rust-analyzer
  - crates
  - Even Better TOML

## Cargo 프로젝트 만들기

- 바이너리 파일을 생성하는 프로젝트
  - cargo new [프로젝트명]
  - Cargo.toml과 main.rs가 생성됨
- 라이브러리 파일을 생성하는 프로젝트
  - cargo new [프로젝트명] --lib
  - Cargo.toml과 lib.rs가 생성됨

## Cargo 프로젝트 빌드 및 실행

- 프로젝트 빌드 방법
  - cargo build [--release]
  - 기본은 디버그 모드로 빌드된다.
  - 릴리즈 모드로 빌드하고 싶다면 --release를 추가하면 된다.
- 프로젝트 실행 방법
  - cargo run [--release]
  - 기본은 디버그 모드로 빌드 후 실행된다.
  - 릴리즈 모드로 빌드 후 실행하고 싶다면 --release를 추가하면 된다.

# 子人は上に

utilForever@gmail.com

https://github.com/utilForever

Facebook, Twitter: @utilForever