RUST

23학번 고은이

INDEX

RUST 란?

Rust 실습

RUST

REASON

Stack Overflow가 조사한 7년 연속 가장 인기있는 언어 1위

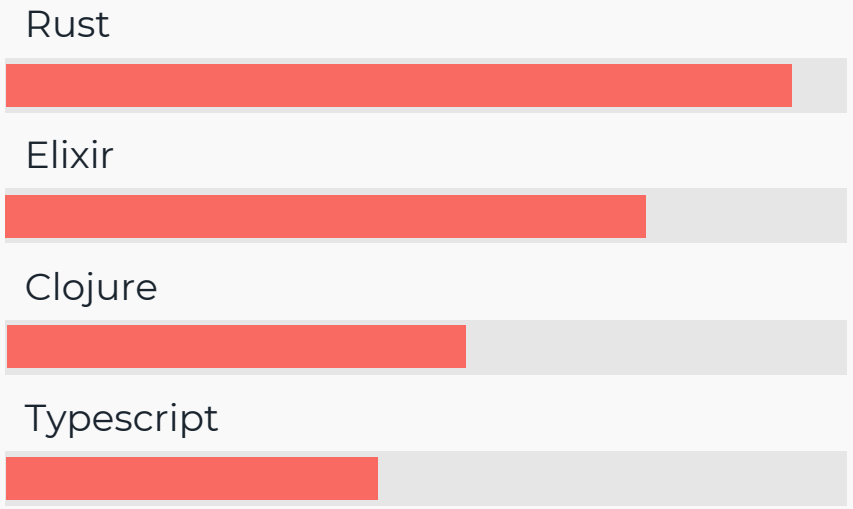
“ Go 나 Rust 와 같은 비주류 언어로 제작한 렌섬웨어 공격이 늘고 있다. “

” Go나 Rust를 활용해 만든 랜섬웨어는 하나의 코드로 윈도, 맥 OS, 리눅스 등 다양한 운영체제를 공격할 수 있다. 기존 주류 언어인 ‘C’와 ‘C++’로 제작된 것보다 분석 데이터가 부족해 탐지 확률도 낮다. “

“ 최근 사이버 보안 분야에서 인기를 얻고 있는 프로그래밍 언어다. “

RUST

Loved



RUST??

2010년도에 모질라 재단에서 발표하고 러스트 재단에서 개발 되고 있는 프로그래밍 언어로 메모리 안전성과 성능, 편의에 중점을 둔 언어다.
객체 지향 프로그래밍 스타일을 지원한다.
Rust언어를 고안한 그레이든 호아레는 생물학에 관심이 많아 녹병곰팡이라는 생물의 이름을 붙였다.

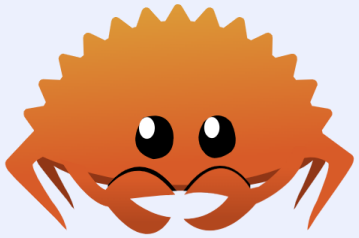
Cargo 라는 빌드 시스템을 가지고 있다. 빌드 과정에서 사전에 불안정한 결과를 방지할 수 있다.
C 또는 C++과 유사한 모양을 하고 있으나 의미상으로는 다른 문법을 가지고 있다.

FERRIS

Ferris

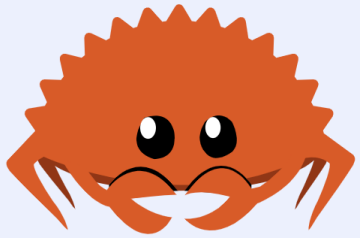
Rust + Crustacean = Rustacean

Original



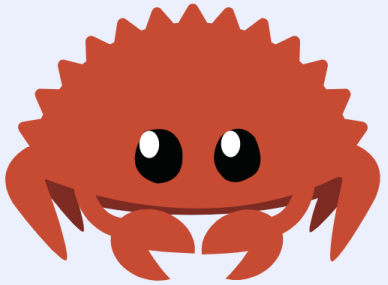
[.png] [.svg]

Flat



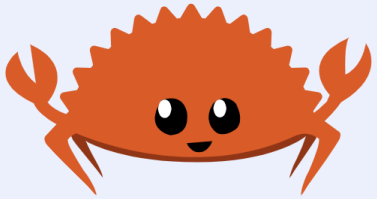
[.png] [.svg]

Extra-cute



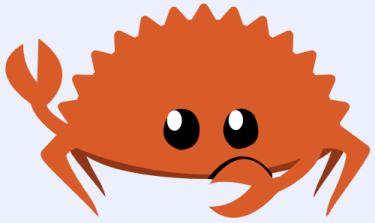
[.png] [.svg]

Happy



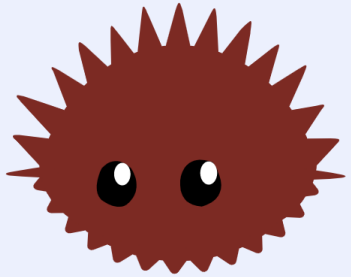
[.png] [.svg]

Gesturing



[.png] [.svg]

Corro the Unsafe Rusturchin



[.svg]

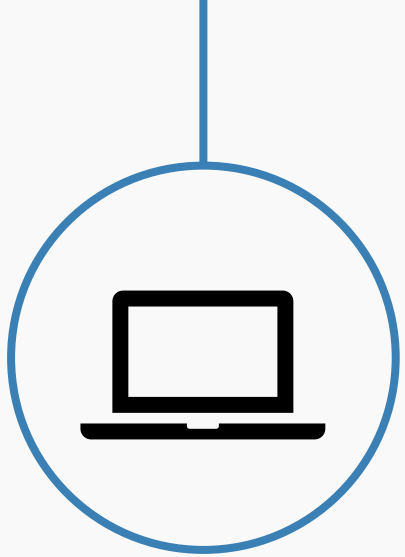
RUST 장점

메모리 안전성

Rust는 컴파일 시 메모리 안전성을 확인하기 때문에 컴파일이 된 이후에는 이런 문제를 걱정하지 않아도 된다.

안전한 동시성 프로그래밍

두개 이상의 스레드가 동시에 메모리에 접근하는 문제를 borrow checker라는 것을 이용하여 컴파일 중에 방지한다.



Zero-cost Abstraction

쓰지 않는 기능에 대해서는 성능에 영향을 주지 않는다. 추상적으로 코드를 작성하더라도 컴파일 할 때 가장 low-level하게 실행된다.

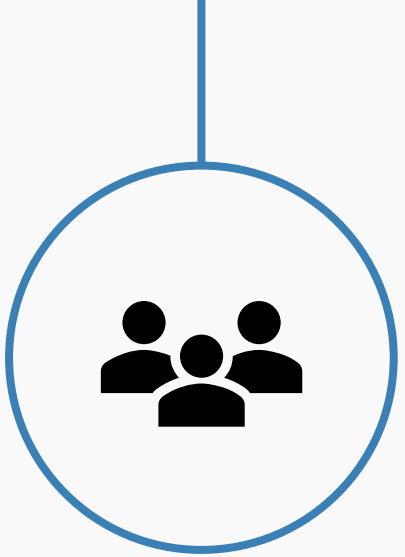
RUST 단점

범용성

요즘 조금씩 인기를 얻고 있다고 하지만 이미 많이 쓰이고 있는 언어에 비해 하는 사람도 찾는 사람도 적다.

난이도

Rust의 개념들을 배우는데 시간이 오래 걸리고 많은 공부를 요구한다.



커뮤니티 부족

다른 언어에 비해 커뮤니티의 수가 적고 크기도 작다.

RUST 를 사용한 기업

Discord
백엔드에 go 를 사용 했지만 Rust로 옮겼다.

마이크로소프트
윈도우 커널 개발에 C++부분을 Rust로 대체 중이다.

트위터
인메모리 캐시를 Rust로 재작성 하였다.

드롭박스
파일 동기화 엔진을 Rust로 재작성 하였다.

간 단 한

RUST 실 습

Hello, SCP !

```rust
fn main() {
    println!("hello, scp");
}
```

Println! 을 사용하여 문자열 출력

```
hello, scp
```

LET 변수

```
fn main(){
    let my_name = "EUNI";
    let my_age = 20;
    println!("My name is {} and my age is {}", my_name, my_age);
}
```

```
My name is EUNI and my age is 20
```

Let 을 사용하여 변수 선언

Let 으로 선언한 변수는 기본적으로 값을 변경 할 수 없다.

MUT 변수

```
fn main(){
    let mut x = 4;
    println!("x : {}", x);
    x = 5;
    println!("x : {}", x);
}
```

```
x : 4
x : 5
```

Mut 키워드를 사용하여 변수의 값을 변경할 수 있다.

SHADOWING

```
fn main() {
    let x = 4;
    let x = x + 3;
    let x = x * 2;
    println!("x : {}", x);
}
```

```
x : 14
```

Shadowing 은 mut를 선언하지 않고 let 키워드를 반복하여 새 변수를 선언할 수 있다.

변수의 값만 바꿀 수 있는 mut 와 달리 shadowing은 데이터 타입도 변경 가능하다.

Thank you!