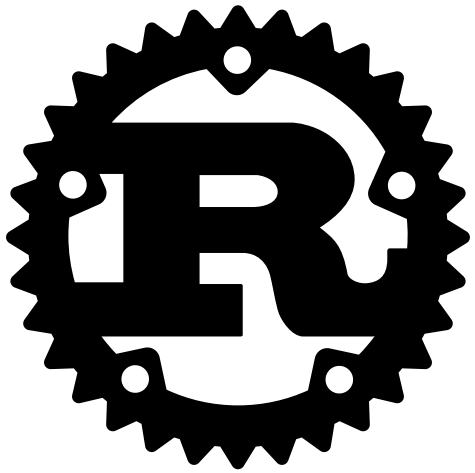

Rust in the Web

Eric Kunze

Inhalt

1 Historie

2 Warum Rust?

3 Was macht Rust sicher?

4 Are we Web yet?

5 Iron

6 Demo

Rust - Historie

- 2006 persönliches Projekt von Graydon Hoare
- ab 2009 Projekt bei Mozilla
- 15. Mai 2015 Veröffentlichung der Version 1.0

- Entwicklung einer neuen Browserenging $\rightarrow$ Servo

Warum Rust?

- Warum nicht C++ oder Java?

- Rust
  - schnell
  - Kontrolle über das System
    - minmale Runtime $\rightarrow$ Stack
    - keine Garbage Collection
  - sicher
  - Features von höheren- und funktionalen Programmiersprachen

Was macht Rust sicher?

- Ownership and Borrowing
- Ownership
  - jede Ressource hat genau einen Besitzer

```
fn foo() {
  let mut y: Vec<i32>
    = Vec::new();

  y.push(4);
  bar(y);
  y.push(5); // Compiler Error
}
```

```
fn foo(x: Vec<i32>) {
  ...
}
```

Was macht Rust sicher?

- Borrowing
  - jede Ressource kann verliehen werden
- shared borrow

```
fn foo() {
  let mut y: Vec<i32>
    = Vec::new();

  y.push(4);
  bar(&y);
  y.push(5);
}
```

```
fn bar(x: &Vec<i32>){
  println!("{}",x[0]);   // Ok
  x.push(1); // Compiler Error
}
```

Was macht Rust sicher?

- mutable borrow
  - nur eine Referenz auf eine Ressource und diese hat nur einen Besitzer

```rust
fn foo() {
  let mut y
    = vec![1, 2, 3, 4, 5];
  let mut v: Vec<i32>
    = Vec::new();

  bar(&y, &mut v); // Ok
  bar(&y, &mut y);
    // Compiler Error
}
```

```rust
fn bar(x: &Vec<i32>,
    y: &mut Vec<i32>){

  for v in x{
    y.push(*v);
  }
}
```

Was macht Rust sicher?

- Aliasing und Mutation zur gleichen Zeit wird verhindert
- Bsp. C++

```
1  void foo(){
2    int *y = new int[10];
3    for(int i=0;i<10;i++)
4      y[i] = i;
5
6    int *x = &y[9];
7    y=bar(y);
8    delete[] y;
9    cout<<*x<<endl;
10 }
```

```
1  int* bar(int *v){
2    delete[] v;
3    v = new int[5]
4    for(int i=0;i<10;i++){
5      v[i] = i*2;
6    }
7    return v;
8  }
```

Are we Web yet?

Bild http://arewewebyet.com/

Iron

- Webframework
- basiert auf Hyper
- hochgradig Nebenläufig
- keine unnötigen Features im Basisframework, aber leicht erweiterbar
- bietet Infrastruktur, um das Framework an individuelle Bedürfnisse anzupassen

Demo