

WHY I LOVE RUST



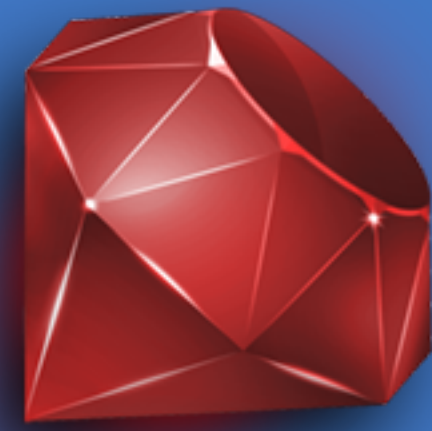
Stanko Krtašić Rusendić

 github.com/Stankec

 [@monorkin](https://twitter.com/monorkin)

01

BACKGROUND



Ruby

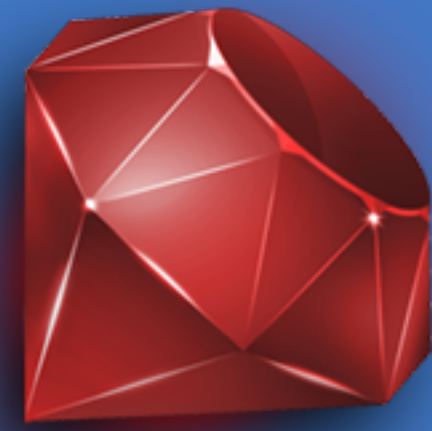


Swift / Objective-C / C++ / Java

JS



JS



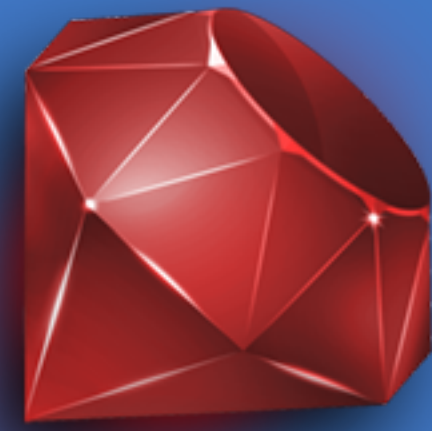
Ruby is elegant and has a lot of libraries



Swift / Objective-C / C++ / Java is fast



JS is popular



Ruby is SLOW



Swift / Objective-C / C++ / Java is
HIGH RITUAL AND NOT REALLY
SAFE



JS is WHAT THE F*!\$K

Ruby programs versus Go

all other Ruby programs & measurements

by benchmark task performance

binary-trees

source	secs	KB	gz	cpu	cpu load
<u>Ruby</u>	58.72	192,132	1123	166.36	67% 61% 68% 90%
<u>Go</u>	39.88	361,208	688	152.12	96% 95% 96% 96%

regex-dna

source	secs	KB	gz	cpu	cpu load
<u>Ruby</u>	7.98	108,480	529	23.28	95% 69% 64% 64%
<u>Go</u>	3.89	369,380	1229	8.29	43% 53% 61% 82%

k-nucleotide


```

public class DataRaces {
    public static void main(String[] args) {
        UseCounter c = new UseCounter();
        Thread t1 = new Thread(c);
        Thread t2 = new Thread(c);
        Thread t3 = new Thread(c);
        t1.start();
        t2.start();
        t3.start();

        Counter.count = 0;

        SynchronizedUseCounter sc = new SynchronizedUseCounter();
        Thread t4 = new Thread(sc);
        Thread t5 = new Thread(sc);
        Thread t6 = new Thread(sc);
        t4.start();
        t5.start();
        t6.start();
    }
}

```

```

class Counter {
    public static long count = 0;
}

class UseCounter implements Runnable {
    public static void increment() {
        Counter.count++;
        System.out.print(Counter.count + " ");
    }
    public void run() {
        increment();
        increment();
        increment();
    }
}

class SynchronizedUseCounter implements Runnable {
    public static synchronized void increment() {
        Counter.count++;
        System.out.print(Counter.count + " ");
    }
    public void run() {
        increment();
        increment();
        increment();
    }
}

```

It prints something like this:

```
1 2 3 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```



```
>>> '3' + 2
32
>>> '3' - 2
1
>>> [] + {}
[object Object]
>>> {} + []
0
>>> NaN
NaN
>>> NaN == NaN
false
>>> typeof NaN
number
>>>
```

RUST



No data races
No unexpected mutations
Safe threading
Fast

```
1
2 fn main() {
3     println!("Hello World!");
4     print_fibbonaci(5)
5 }
6
7 fn print_fibbonaci(n: i64) {
8     println!(
9         "Fibbonaci of {} is {}",
10        n, fibbonaci(n)
11    )
12 }
13
14 fn fibbonaci(n: i64) → i64 {
15     if n == 0 {
16         return 0;
17     }
18     else if n == 1 {
19         return 1;
20     }
21
22     fibbonaci(n-1) + fibbonaci(n-2)
23 }
```

```
1
2 fn main() {
3     println!("Hello World!");
4     print_fibbonaci(5)
5 }
6
7 fn print_fibbonaci(n: i64) {
8     println!(
9         "Fibbonaci of {} is {}",
10        n, fibbonaci(n)
11    )
12 }
13
14 fn fibbonaci(n: i64) → i64 {
15     if n = 0 {
16         return 0;
17     }
18     else if n = 1 {
19         return 1;
20     }
21
22     fibbonaci(n-1) + fibbonaci(n-2)
23 }
```

Function name

Return type

Input name and type


```
1
2 fn main() {
3     println!("Hello World!");
4     print_fibbonaci(5)
5 }
6
7 fn print_fibbonaci(n: i64) {
8     println!(
9         "Fibbonaci of {} is {}",
10        n, fibbonaci(n)
11    )
12 } → Returns `()`
```

```
13
14 fn fibbonaci(n: i64) → i64 {
15     if n == 0 {
16         return 0;
17     }
18     else if n == 1 {
19         return 1;
20     }
21
22     fibbonaci(n-1) + fibbonaci(n-2)
23 }
```

Implicit return


```
68
69 match i {
70     1 => println!("Yay"),
71     2 => println!("Nooooooo!"),
72     _ => println!("What is this!?!")
73 }
74
75 loop {
76     println!("Hi!");
77 }
78
79 for greting in greetings {
80     println!(greting);
81 }
```

~

```
24
25
26
27
28
29
30 fn print_name(user: Option<User>) {
31     match user {
32         Some(user) => println!("My name is {}", user),
33         None => println!("No user given")
34     }
35 }
```

~

~

```
38
39
40
41
42
43
44 fn handle_book(book: Book) {
45     return_book(book);
46     borrow_book(book);
47 }
```

~

~

~

~

~

~

```
38
39
40
41
42
43
44 fn handle_book(book: Book) {
45     return_book(book);
46     borrow_book(book);
47 }
```



Won't compile – Book given as an argument but not owned

~
~
~
~
~
~
~

```
38
39
40
41
42
43
44 fn handle_book(book: Book) {
45     let book = return_book(book);
46     borrow_book(book);
47 }
48
49
50
51
52
53
54
```

38

39

40

41

42

43

44 **fn** handle_book(book: Book) {

45 return_book(&book);

46 borrow_book(&book);

47 }

48

49

50

51

52

53


```
38
39
40
41
42
43
44 fn handle_book(book: Book) {
45     return_book(&book);
46     borrow_book(&book);
47 }
```

& indicates that we are passing a reference

```
53
54 fn print_name(user: Option<User>) → Error<NoUserError, User> {
55     match user {
56         Some(user) ⇒ {
57             println!("My name is {}", user);
58             return Ok(user);
59         },
60         None ⇒ {
61             println!("No user given");
62             return Err(NoUserError {})
63         }
64     }
65 }
66
```

OS

IN ACTION



Subscribe 1,613

69,074 views

 Add to Share More

165 2

Published on Jul 16, 2016

Category	Music
License	Standard YouTube License

Up next

Autoplay

Denis & Denis - Soba 23 (video 1985)

dzonikg
260,483 views

4:11

**Mix - Denis & Denis - Program
tvog kompjutera (1984)**

YouTube

50+
VIDEO
((▶))

♪ EX YU POP-ROCK MIX #1 ♪

Simeon Delev
Recommended for you

1:02:00

dow Help


why_i_love_rust — Edited


Chart

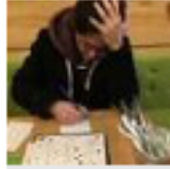
Notifications


Recent Files


Collaborate


2017-02-09 18.59.48.jpg
1 day ago

2017-02-09 18.59.46.jpg
1 day ago

2017-02-09 18.39.37.jpg
1 day ago

2017-02-12 17.03.02.jpg
1 day ago

2016-12-31 16.15.20.mp4
5 days ago

2016-12-31 10.46.50.mov
5 days ago

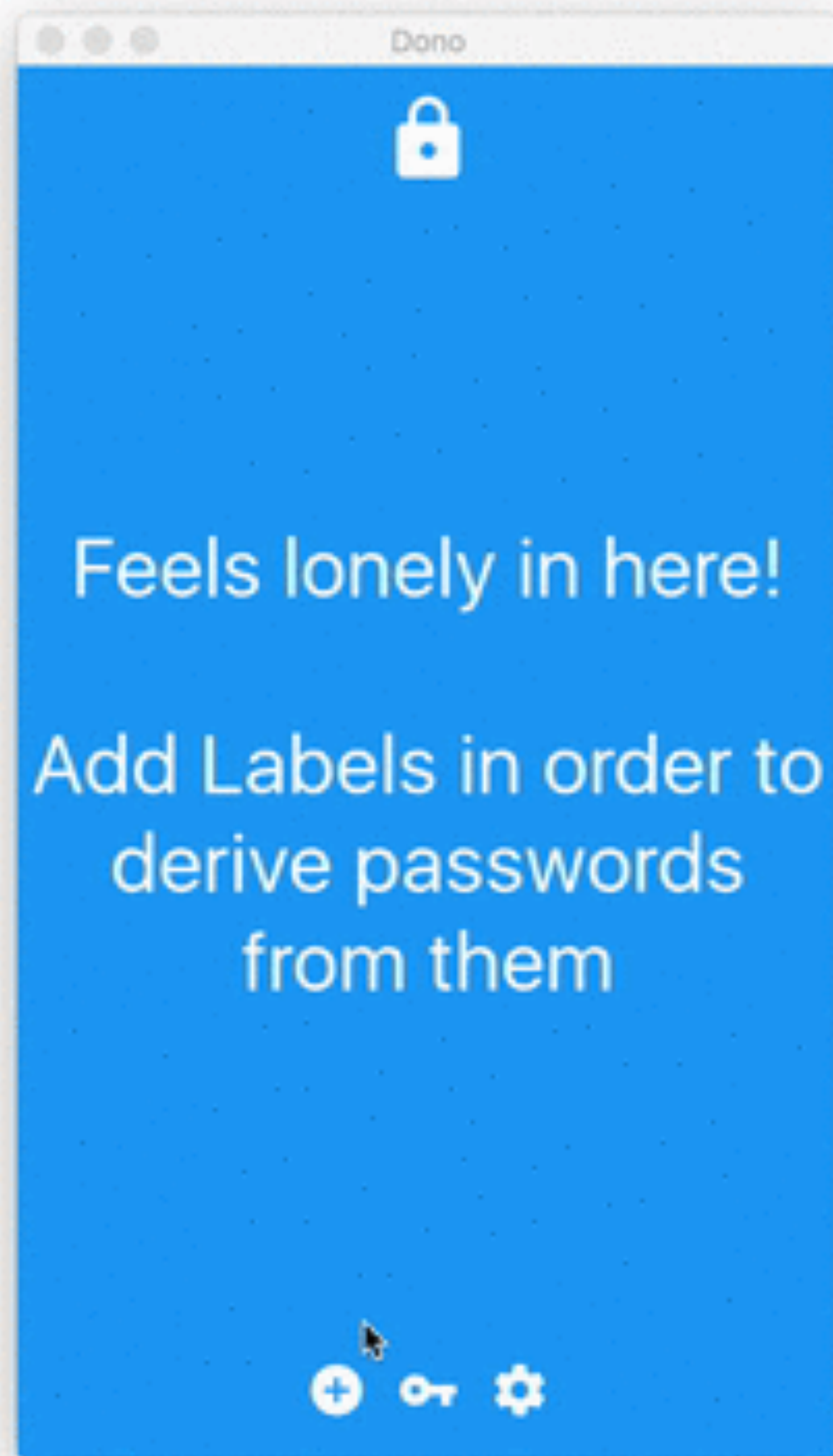
✓ Up to date

Party mix

260,483 views

4:11

Mix - Denis & Denis - Program
tvoj kompiuter (1984)



04

LEARN IT

<https://doc.rust-lang.org/stable/book/>