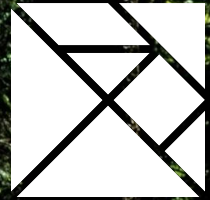


Next-Generation Programming: Rust and Elm

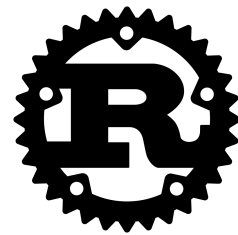


@rtfeldman

THE NEXT GENERATION



Rust and Elm



modern

non-mainstream

pushing boundaries

significantly different

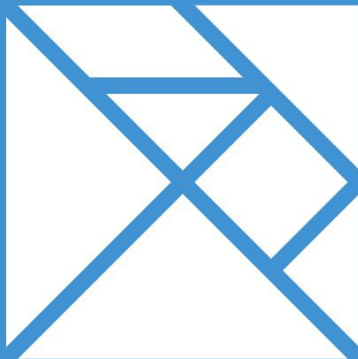
alternative to C++

Rust

A language empowering everyone
to build **reliable** and **efficient** software.

elm

alternative to JavaScript



A **delightful** language
for **reliable** webapps.

Try

Tutorial

compiles to binaries (no garbage collector)

C interop

packages, editor plugins, learning materials

Rust

A language empowering everyone
to build reliable and efficient software.

often compared to Go

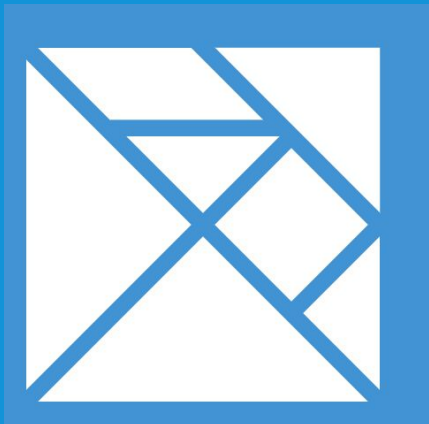
Go: simple, quick to learn, fast compilation

Rust: complex, lots to learn, ~slow compilation

Rust

A language empowering everyone
to build reliable and efficient software.

elm



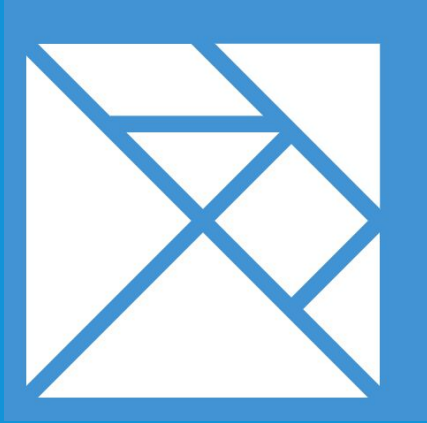
A delightful language
for reliable webapps.

fast, friendly compiler, great error messages

JavaScript interop

packages, editor plugins, learning materials

elm



A delightful language
for reliable webapps.

often compared to JS frameworks (React, etc.)

frameworks: write JavaScript or TypeScript

Elm: use a different programming language

Rust

Elm



battle tested

Rust: 2010 single dev → 2020 “radical openness”

RustConf 2018 Keynote youtu.be/J9OFQm8Qf1I

“Organizational Debt” boats.gitlab.io/blog/post/rust-2019

several full-time paid developers, community engineers, etc.
large, complex language, frequent compiler releases

Elm: 2012 single dev → 2020 core team w/ BDFL

“What is Success?” youtu.be/uGlzRt-FYto

“The Hard Parts of Open Source” youtu.be/o_4EX4dPppA

one full-time paid developer, the rest all volunteers
small, simple language, infrequent compiler releases

Rust conferences

RustConf since 2016

Rust Belt Rust since 2016

RustFest since 2016

RustCon Asia since 2019

Rust LATAM since 2019

Elm conferences

elm-conf since 2016

Elm Europe since 2017

Oslo Elm Days since 2017

Elm in the Spring since 2018

Elm Japan (announced for 2020, canceled due to COVID-19)

Rust

Elm



reliability

Rust Elm

static type-checking

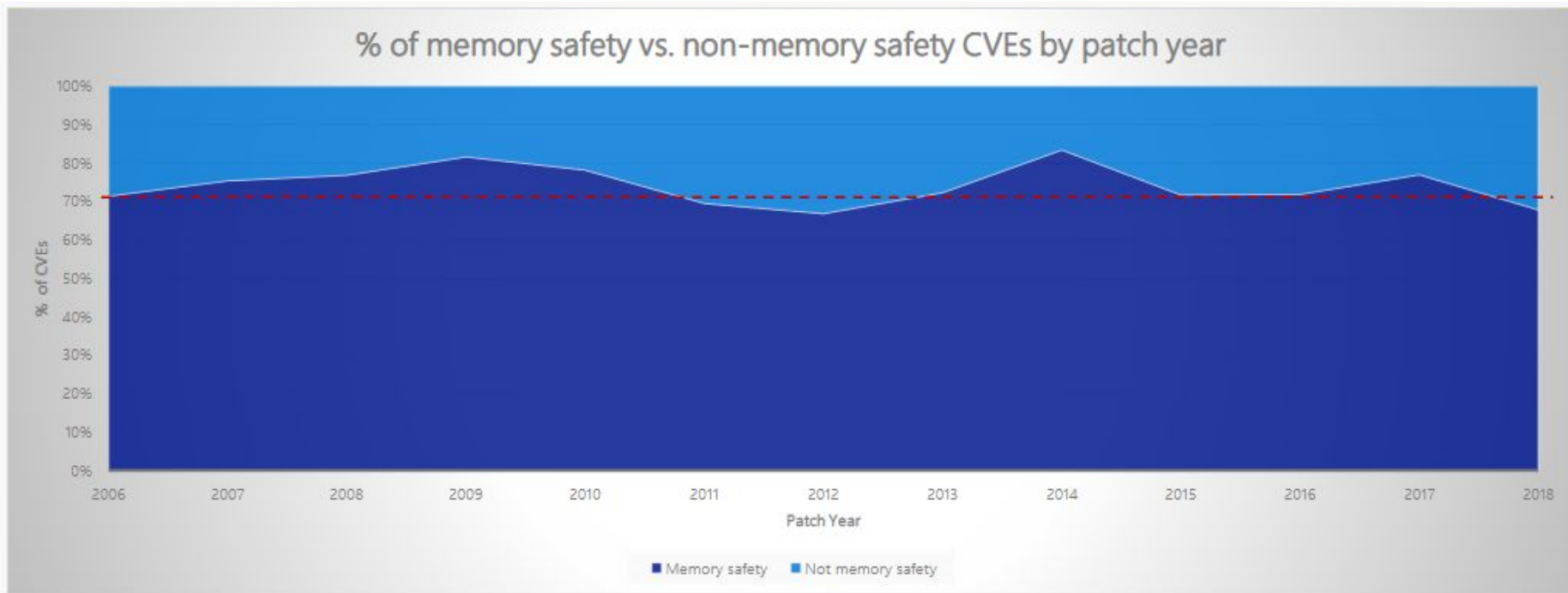
sound type system

no “any” type

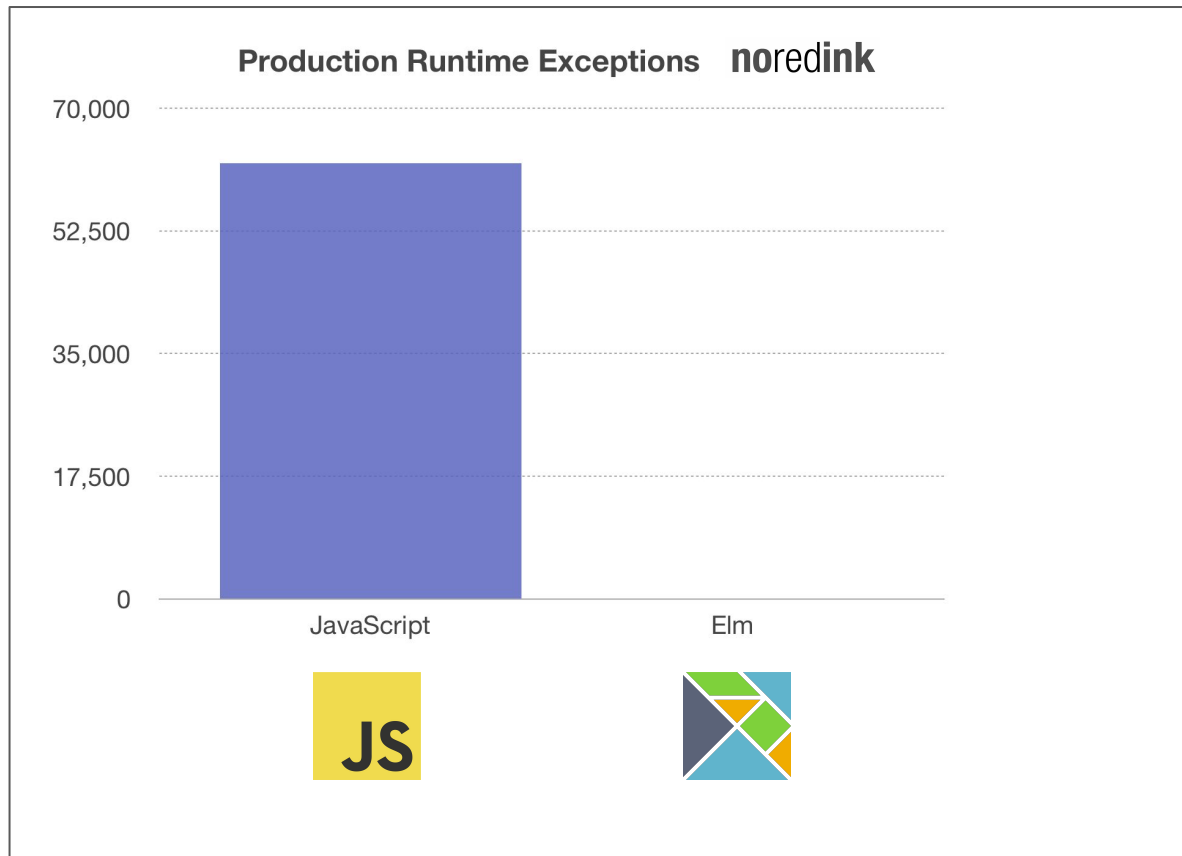
no “billion dollar mistake” (`null`)

immutable by default

Rust: “Memory safety, no data races”



Elm: “No runtime exceptions”



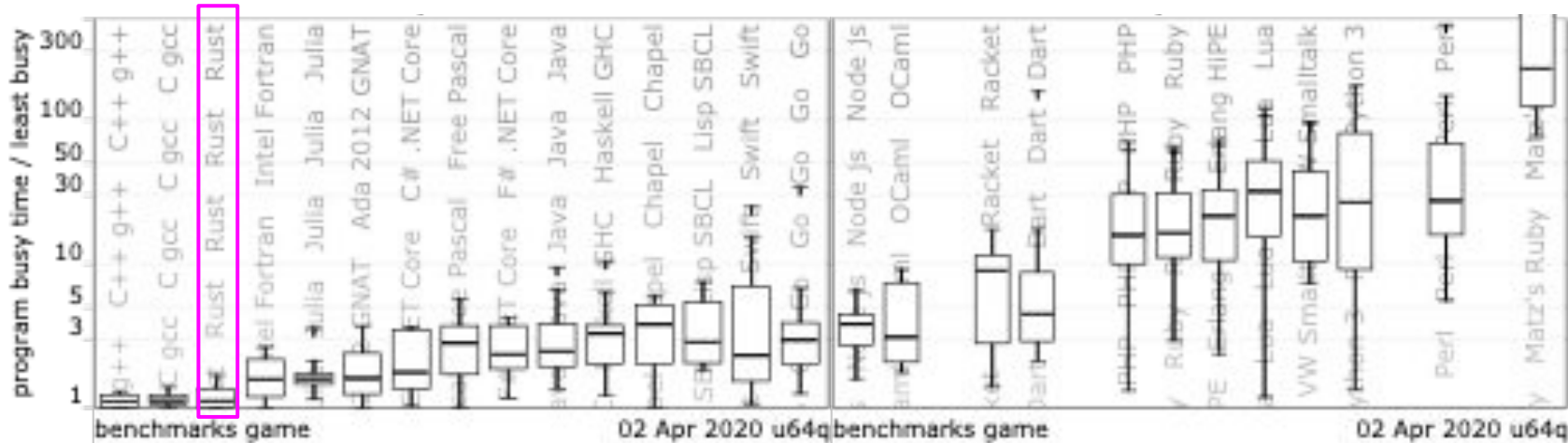
not **zero**,
but negligible

Elm

Rust

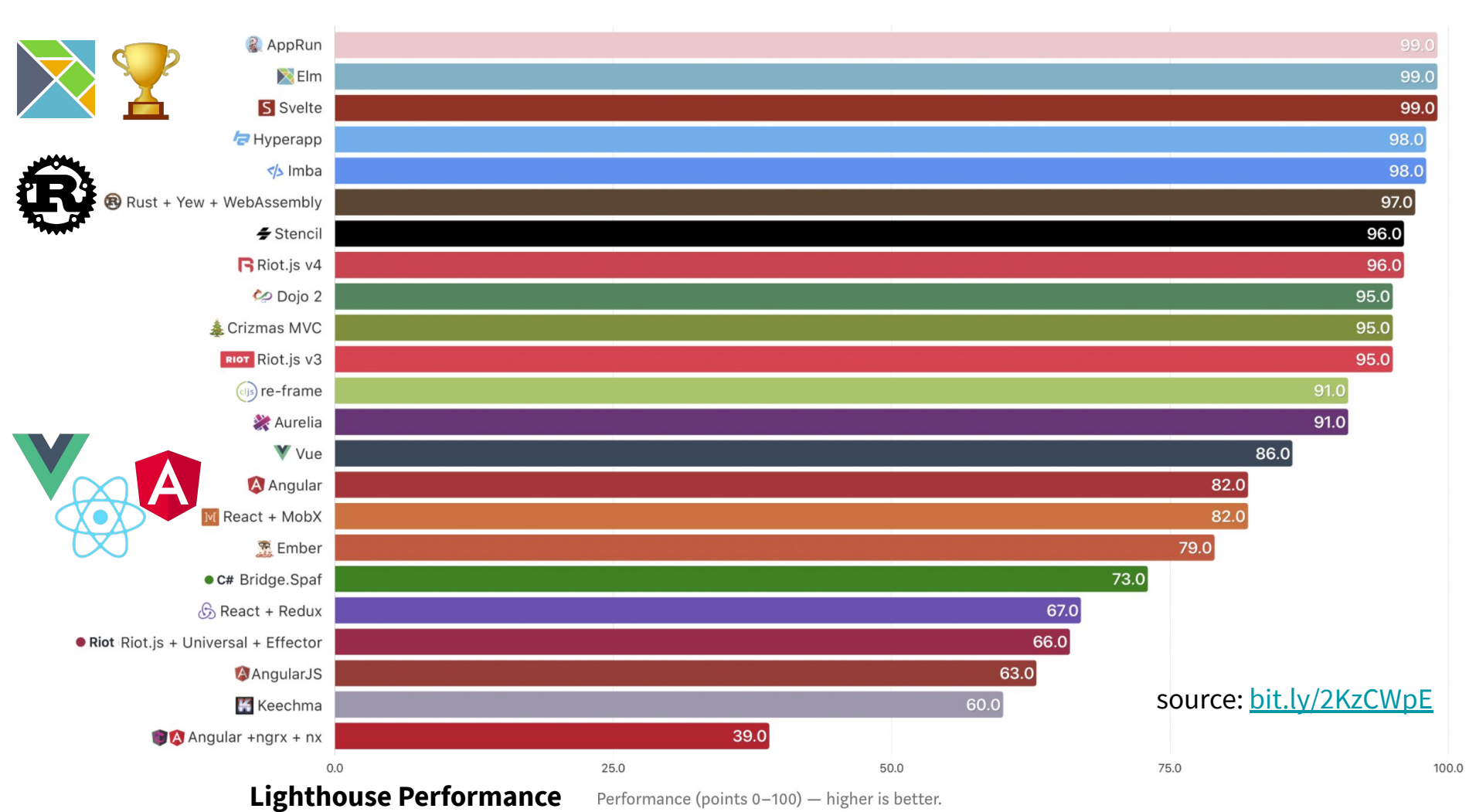


runtime performance

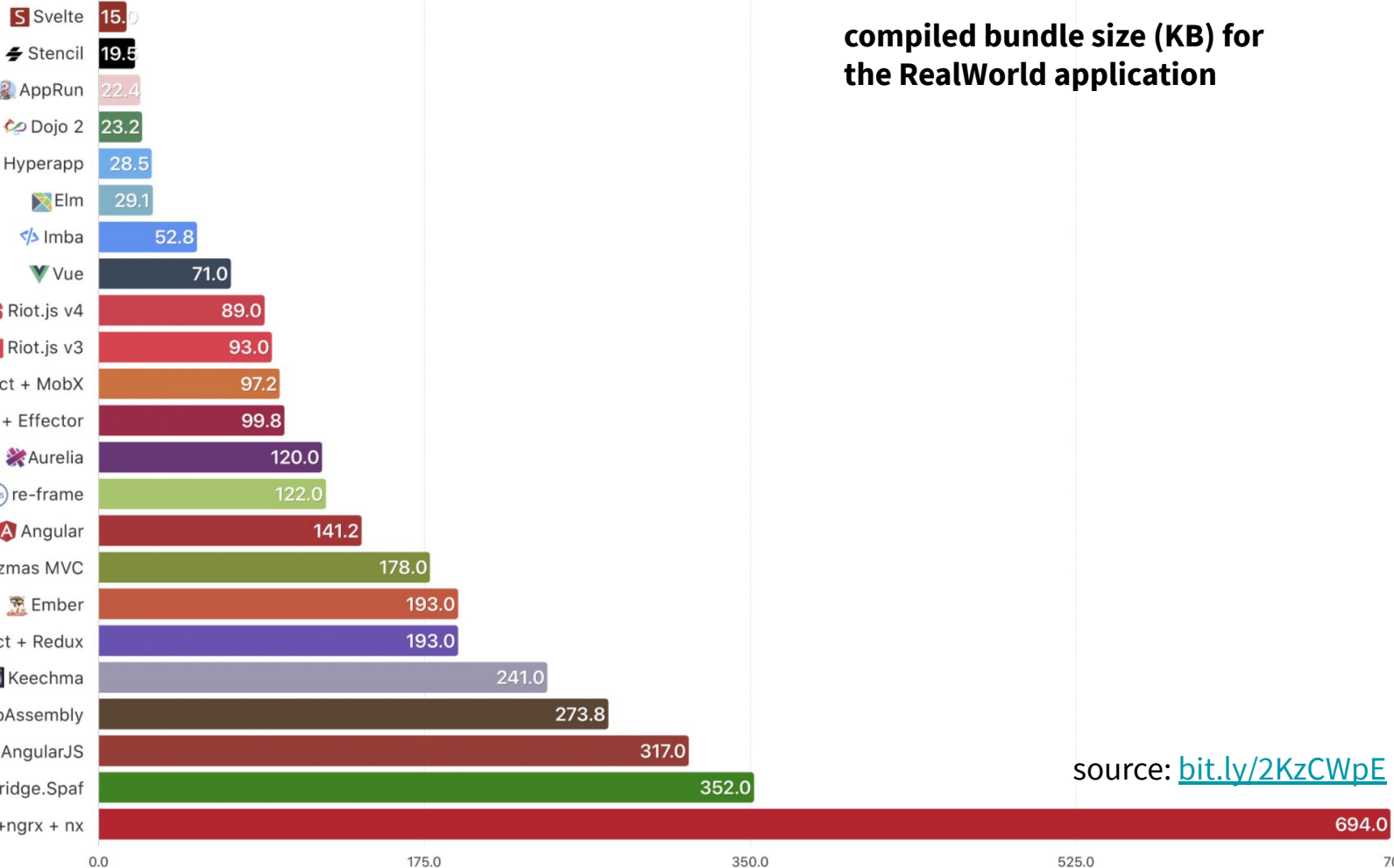
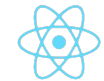


The Computer Language Benchmarks Game

source: bit.ly/2Y986w3



compiled bundle size (KB) for the RealWorld application



source: bit.ly/2KzCWpE

Elm

Rust



helpful compiler

This `user` record does not have a `phoenNumber` field:

```
9|      user.phoenNumber
      ^^^^^^^^^^^^^^^
```

```

error[E0499]: cannot borrow `foo.bar1` as mutable more than once at a time
--> src/test/compile-fail/borrowck/borrowck-borrow-from-owned-ptr.rs:29:22
    |
28 |         let bar1 = &mut foo.bar1;
    |                        ----- first mutable borrow occurs here
29 |         let _bar2 = &mut foo.bar1;
    |                        ^^^^^^^ second mutable borrow occurs here
30 |         *bar1;
31 |     }
    |     - first borrow ends here

```

\$ rustc --explain E0200

Unsafe traits must have unsafe implementations. This error occurs when an implementation for an unsafe trait isn't marked as unsafe. This may be resolved by marking the unsafe implementation as unsafe.

```
struct Foo;
```

```
unsafe trait Bar { }
```

```
// this won't compile because Bar is unsafe and impl isn't unsafe
```

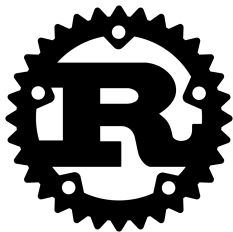
```
impl Bar for Foo { }
```

```
// this will compile
```

```
unsafe impl Bar for Foo { }
```

“If it compiles, it usually works.”

Rust



rust-lang.org

doc.rust-lang.org/book

users.rust-lang.org

Elm



elm-lang.org

guide.elm-lang.org

discourse.elm-lang.org

