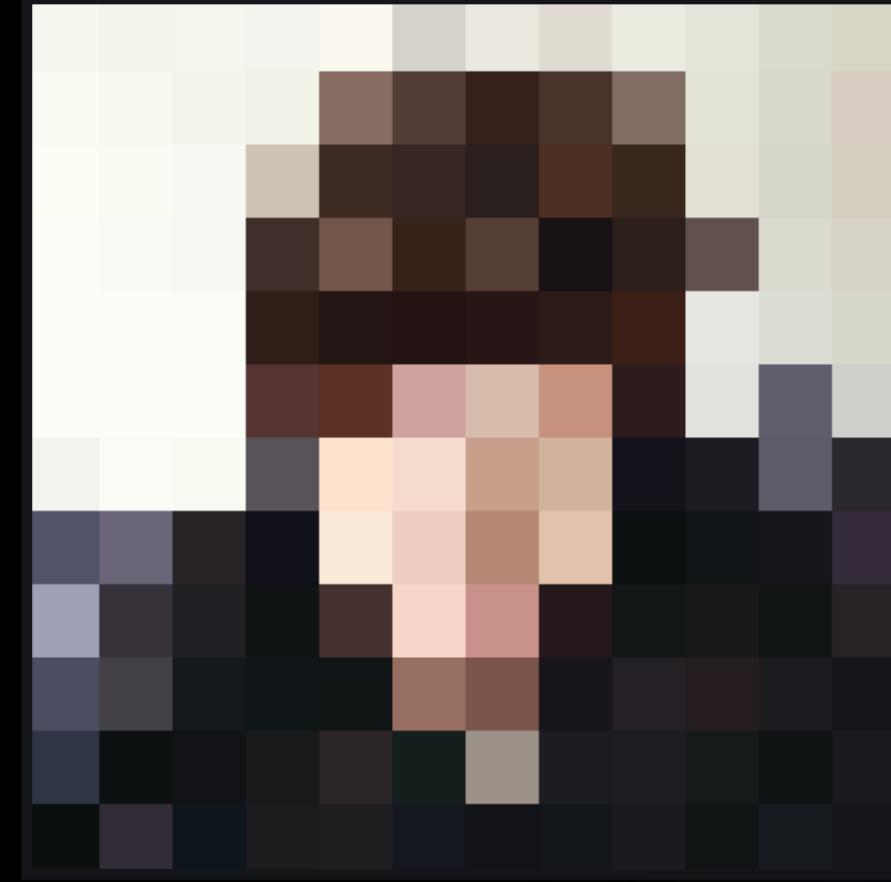


Building native Ruby extensions in Rust

Baltic Ruby # /// 2024-06-14

Guilherme Carreiro # karreiro.com



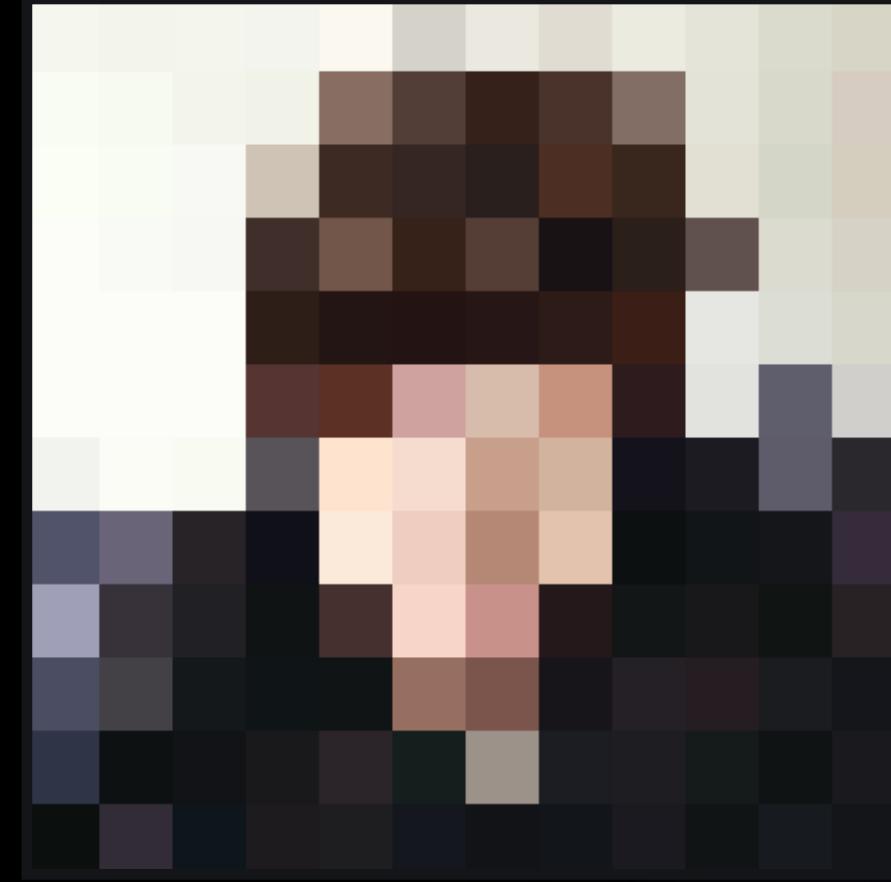


Guilherme Carreiro

@karreiro

karreiro.com/talks





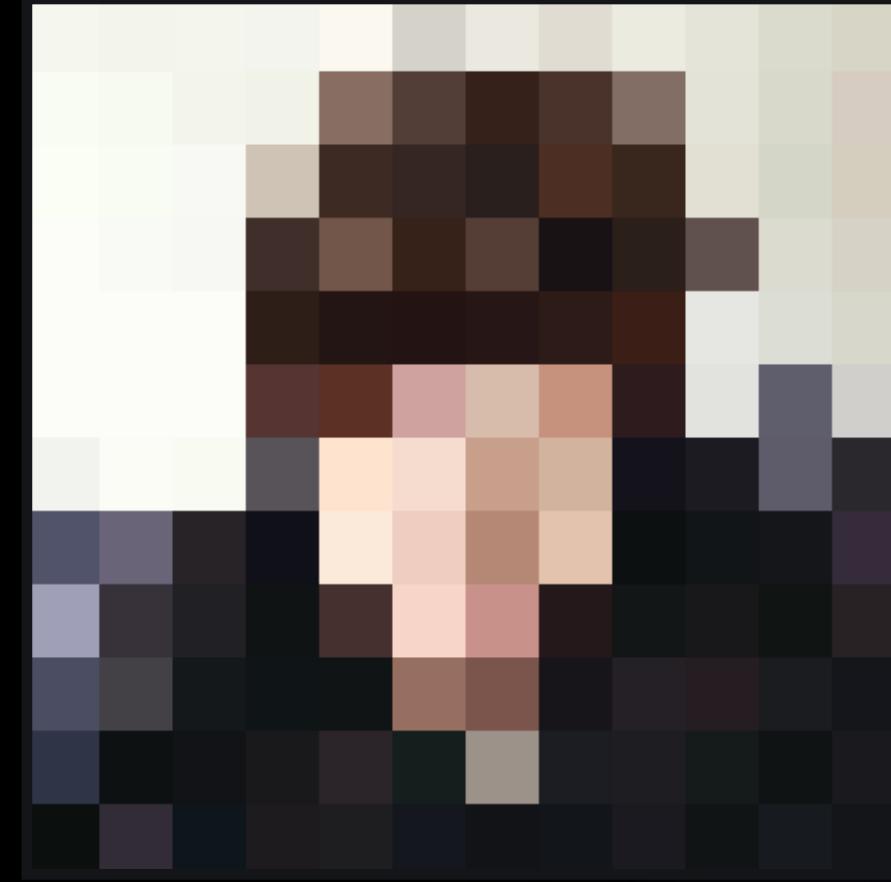
Guilherme Carreiro

@karreiro

karreiro.com/talks



Nova Europa, Campinas, Brasil



Guilherme Carreiro

@karreiro

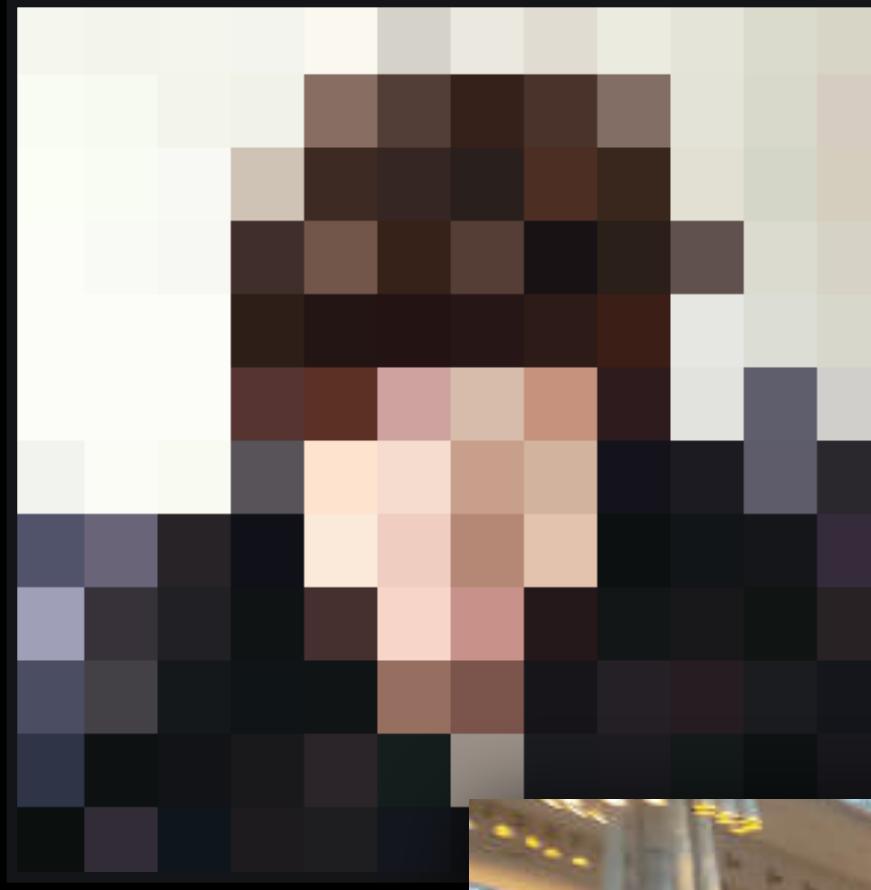
karreiro.com/talks



🇧🇷 Nova Europa, Campinas, Brasil



🇪🇸 Malasaña, Madrid, España



Guilherme Carreiro

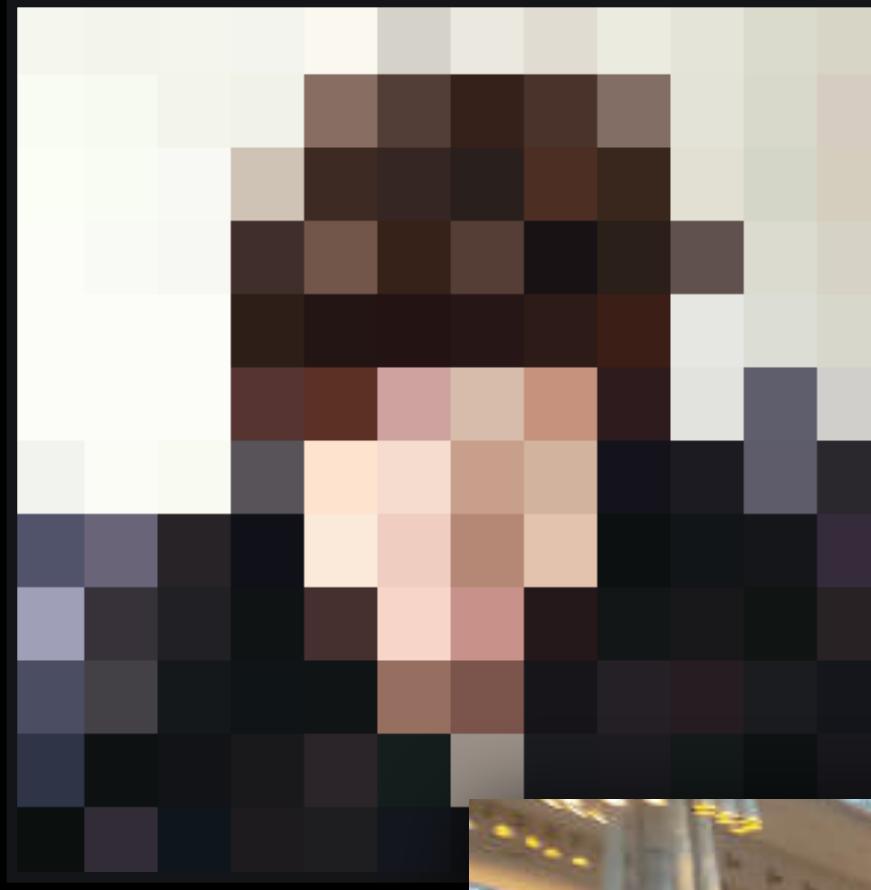
@karreiro

karreiro.com/talks



🇧🇷 Nova E

spaña



Guilherme Carreiro

@karreiro

karreiro.com/talks



Lógica de
Programação

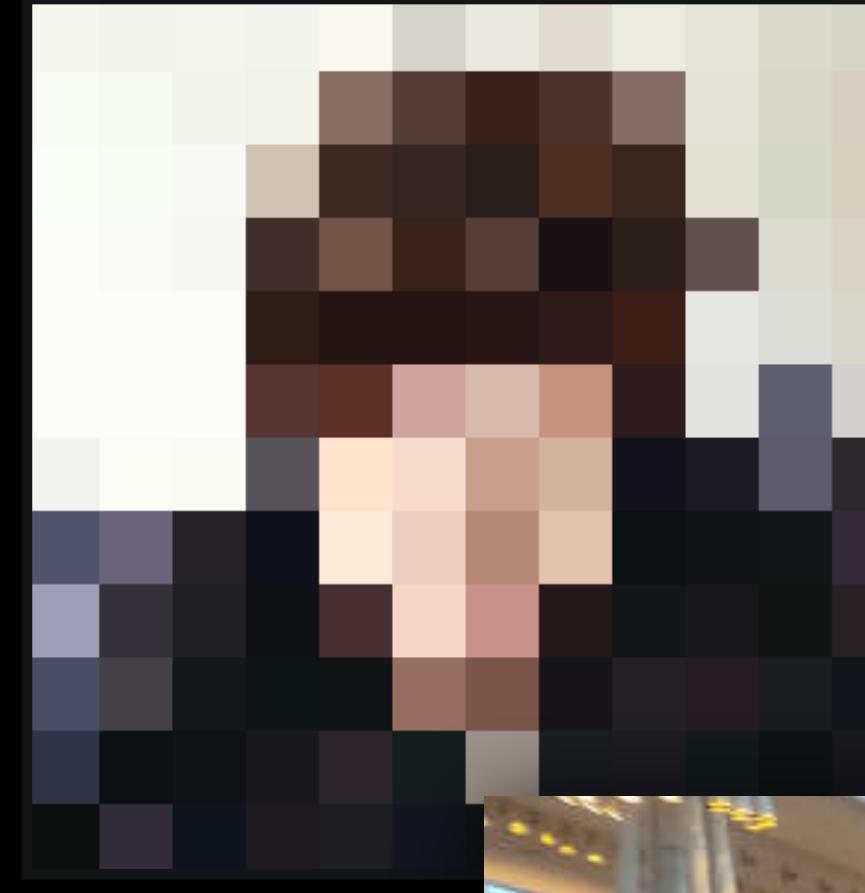
Aprendendo a Programar

Novo

The image shows the front cover of a book titled "Lógica de Programação" by Guilherme Carreiro. The cover features a large geometric pattern of orange and white triangles. Below the title, the subtitle "Aprendendo a Programar" is written. At the bottom, there is a small image of a 3D cube structure.



España



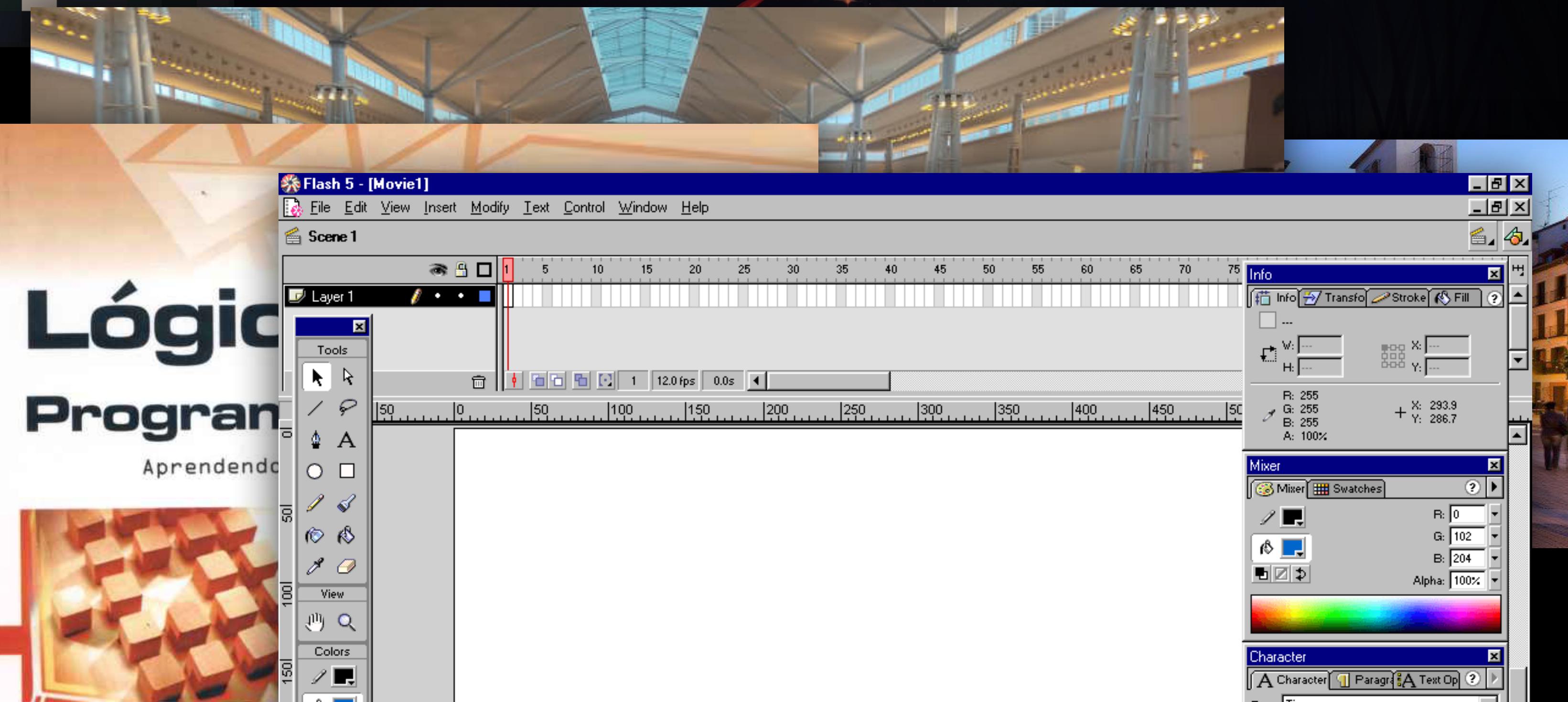
Guilherme Carreiro

@karreiro

karreiro.com/talks



Novo





Performance

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end

# No Rust!
```

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end
```

No Rust!

```
def compute2(iterations)
  result = 0

  iterations.times do |i|
    result += Ext.compute(i)
  end

  result
end
```

```
fn compute(input: i32) -> i32 {
    input * 2
}
```

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end
```

No Rust!

```
def compute2(iterations)
  result = 0

  iterations.times do |i|
    result += Ext.compute(i)
  end

  result
end
```

```
fn compute(input: i32) -> i32 {
    input * 2
}
```

```
def compute3(iterations)
  Ext.compute(iterations)
end
```

```
fn compute_all(iterations: i32) -> i32 {
    let mut result = 0;
    for i in 0..iterations {
        result += i * 2;
    }
    result
}
```

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end
```

No Rust!

```
def compute2(iterations)
  result = 0

  iterations.times do |i|
    result += Ext.compute(i)
  end

  result
end
```

```
fn compute(input: i32) -> i32 {
    input * 2
}
```

```
def compute3(iterations)
  Ext.compute(iterations)
end
```

```
fn compute_all(iterations: i32) -> i32 {
    let mut result = 0;
    for i in 0..iterations {
        result += i * 2;
    }
    result
}
```



Fastest

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end
```

No Rust!

```
def compute2(iterations)
  result = 0

  iterations.times do |i|
    result += Ext.compute(i)
  end

  result
end
```

```
fn compute(input: i32) -> i32 {
    input * 2
}
```

```
def compute3(iterations)
  Ext.compute(iterations)
end
```

```
fn compute_all(iterations: i32) -> i32 {
    let mut result = 0;
    for i in 0..iterations {
        result += i * 2;
    }
    result
}
```

2 7.70x slower

1 Fastest

Ruby vs Rust performance

```
def compute1(iterations)
  result = 0

  iterations.times do |i|
    result += i * 2
  end

  result
end
```

No Rust!

```
def compute2(iterations)
  result = 0

  iterations.times do |i|
    result += Ext.compute(i)
  end

  result
end
```

```
fn compute(input: i32) -> i32 {
  input * 2
}
```

2 7.70x slower

3 9.55x slower

```
def compute3(iterations)
  Ext.compute(iterations)
end
```

```
fn compute_all(iterations: i32) -> i32 {
  let mut result = 0;
  for i in 0..iterations {
    result += i * 2;
  }
  result
}
```

1 Fastest

Prototype

*.yaml

*.toml

toml

RELEASERS BLOG GEMS GUIDES KARREIRO

search for toml

Advanced Search →

DISPLAYING GEMS 1 - 30 OF 34 IN TOTAL

FILTER: NAME (12) DESCRIPTION (30) SUMMARY (22) UPDATED LAST WEEK (1) UPDATED LAST MONTH (2)

toml 0.3.0	Parse your TOML.	19,099,573 DOWNLOADS
toml-rb 3.0.1	Toml parser in ruby, for ruby.	38,741,002 DOWNLOADS
perfect_toml 0.9.0	A fast TOML parser gem fully compliant with TOML v1.0.0	160,818 DOWNLOADS
tomlrb 2.0.3	A racc based toml parser	65,488,191 DOWNLOADS
cfn-toml 1.0.12	Configuration file parsing for CFN Bash scripting	12,902 DOWNLOADS
toml-ruby 0.0.1	Library for parsing TOML infile format	3,731 DOWNLOADS
toml_empty 0.3.0	Parse your TOML (allow empty group).	2,587 DOWNLOADS
multi_toml 1.0.0		3,740

The Ruby Toolbox

toml

Search

Categories

Category results are hidden when using a custom project result order

Projects

Full Compact

Next page >

	PROJECT SCORE	DOWNLOADS	STARS
tomlrb	★ 1.58	★ 65,480,388	67
toml-rb	0.99	38,740,392	102
toml	0.54	19,099,511	151
qonfig	0.04	1,092,218	24
tomlib	0.03	421,740	49
tty-config	0.04	414,455	63
perfect_toml	0.02	159,050	53

toml

RELEASES BLOG GEMS GUIDES KARREIRO

search for toml

Advanced Search →

DISPLAYING GEMS 1 - 30 OF 34 IN TOTAL

FILTER: NAME (12) DESCRIPTION (30) SUMMARY (22) UPDATED LAST WEEK (1) UPDATED LAST MONTH (2)

toml 0.3.0	Parse your TOML.	19,099,573 DOWNLOADS
toml-rb 3.0.1	Toml parser in ruby, for ruby.	38,741,002 DOWNLOADS
perfect_toml 0.9.0	A fast TOML parser gem fully compliant with TOML v1.0.0	160,818 DOWNLOADS
tomlrb 2.0.3	A racc based toml parser	65,488,191 DOWNLOADS
cfn-toml 1.0.12	Configuration file parsing for CFN Bash scripting	12,902 DOWNLOADS
toml-ruby 0.0.1	Library for parsing TOML infile format	3,731 DOWNLOADS
toml_empty 0.3.0	Parse your TOML (allow empty group).	2,587 DOWNLOADS
multi_toml 1.0.0		3,740 DOWNLOADS

The Ruby Toolbox

toml

Search

Categories

Category results are hidden when using a custom project result order

Projects

Full Compact

Next page >

	PROJECT SCORE	DOWNLOADS	STARS
tomlrb	★ 1.58	★ 65,480,388	67
toml-rb	0.99	38,740,392	102
toml	0.54	19,099,511	151
qonfig	0.04	1,092,218	24
tomlib	0.03	421,740	49
tty-config	0.04	414,455	63
perfect_toml	0.02	159,050	53

\$



```
$ bundle gem fast_toml --ext=rust
Creating gem 'fast_toml'...
MIT License enabled in config
Code of conduct enabled in config
RuboCop enabled in config
Initializing git repo in /Users/karreiro/Projects/fast_toml
  create  fast_toml/Gemfile
  create  fast_toml/lib/fast_toml.rb
  create  fast_toml/lib/fast_toml/version.rb
  create  fast_toml/sig/fast_toml.rbs
  create  fast_toml/fast_toml.gemspec
  create  fast_toml/Rakefile
  create  fast_toml/README.md
  create  fast_toml/bin/console
  create  fast_toml/bin/setup
  create  fast_toml/.gitignore
  create  fast_toml/test/test_helper.rb
  create  fast_toml/test/test_fast_toml.rb
  create  fast_toml/.github/workflows/main.yml
  create  fast_toml/LICENSE.txt
  create  fast_toml/CODE_OF_CONDUCT.md
  create  fast_toml/.rubocop.yml
  create  fast_toml/Cargo.toml
  create  fast_toml/ext/fast_toml/Cargo.toml
  create  fast_toml/ext/fast_toml/extconf.rb
  create  fast_toml/ext/fast_toml/src/lib.rs

Gem 'fast_toml' was successfully created. For more information on making a RubyGem visit https://bundler.io/guides/creating\_gem.html
$ █
```

```
📁 ~/Projects/fast_toml
> 📂 bin
› 📂 ext
  > 📂 fast_toml
    > 📂 src
      └── ⓘ lib.rs
      [T] Cargo.toml
      ⓘ extconf.rb
> 📂 lib
> 📂 test
  M CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  [T] Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📄 LICENSE.txt
  M README.md
  ⓘ Rakefile
  ⓘ fast_toml.gemspec
  (7 hidden items)
```

```
1 use magnus::{function, prelude::*, Error, Ruby};
1
2 fn hello(subject: String) -> String {
3     format!("Hello from Rust, {}", subject)
4 }
5
6 #[magnus::init]
7 fn init(ruby: &Ruby) -> Result<(), Error> {
8     let module = ruby.define_module("FastToml")?;
9     module.define_singleton_method("hello", function!(hello, 1))?;
10    Ok(())
11 }
```

NORMAL ➤ ↗ main ➤ ⓘ ext/.../src/lib.rs
⌚ ▶ 1·nvim

Neo-tree

📁 ~/Projects/fast_toml

> 📂 bin

⌄ 📂 ext

⌄ 📂 fast_toml

⌄ 📂 src

└ lib.rs

[T] Cargo.toml

↳ extconf.rb

⌄ 📂 lib

⌄ 📂 fast_toml

└ fast_toml.rb

> 📂 test

Ⓜ CODE_OF_CONDUCT.md

🔒 Cargo.lock

[T] Cargo.toml

* Gemfile

🔒 Gemfile.lock

📄 LICENSE.txt

Ⓜ README.md

↳ Rakefile

↳ fast_toml.gemspec

(7 hidden items)

~/Projects/fast_toml ➔

⌚ ▶ 1·nvim

```
③ lib.rs      x | ↗ fast_toml.rb  x
8 # frozen_string_literal: true
7
6 require_relative 'fast_toml/version'
5 require_relative 'fast_toml/fast_toml'
4
3 module FastToml
2   class Error < StandardError; end
1   # Your code goes here...
9 end
```

```
$ bin/console
/Users/karreiro/Projects/fast_toml/lib/fast_toml.rb:9:in `require_relative': cannot load such
file -- /Users/karreiro/Projects/fast_toml/lib/fast_toml/fast_toml (LoadError)
    from /Users/karreiro/Projects/fast_toml/lib/fast_toml.rb:9:in `rescue in <top (requir
ed)>'
    from /Users/karreiro/Projects/fast_toml/lib/fast_toml.rb:5:in `<top (required)>'
    from <internal:/Users/karreiro/.rbenv/versions/3.1.4/lib/ruby/site_ruby/3.1.0/rubygem
s/core_ext/kernel_require.rb>:37:in `require'
        from <internal:/Users/karreiro/.rbenv/versions/3.1.4/lib/ruby/site_ruby/3.1.0/rubygem
s/core_ext/kernel_require.rb>:37:in `require'
        from bin/console:7:in `<main>'
/Users/karreiro/Projects/fast_toml/lib/fast_toml.rb:7:in `require_relative': cannot load such
file -- /Users/karreiro/Projects/fast_toml/lib/fast_toml/3.1/fast_toml (LoadError)
    from /Users/karreiro/Projects/fast_toml/lib/fast_toml.rb:7:in `<top (required)>'
    from <internal:/Users/karreiro/.rbenv/versions/3.1.4/lib/ruby/site_ruby/3.1.0/rubygem
s/core_ext/kernel_require.rb>:37:in `require'
        from <internal:/Users/karreiro/.rbenv/versions/3.1.4/lib/ruby/site_ruby/3.1.0/rubygem
s/core_ext/kernel_require.rb>:37:in `require'
        from bin/console:7:in `<main>'
```

\$

② ▶ 1 · zsh

```
$ tree lib
lib
└── fast_toml
    ├── fast_toml.bundle
    └── version.rb
    fast_toml.rb
```

2 directories, 3 files

\$ █



How it works

```
extern "C" {
    fn rb_define_global_function(
        name: *const u8,
        func: extern "C" fn() -> usize,
        arity: isize
    );
    fn rb_str_new(
        ptr: *const u8,
        len: isize
    ) -> usize;
}

extern "C" fn hello_from_rust() -> usize {
    unsafe { rb_str_new("Hello, world!".as_ptr(), 12) }
}

#[no_mangle]
unsafe extern "C" fn Init_simplest_rust_extension() {
    rb_define_global_function(
        "hello_from_rust\0".as_ptr(),
        hello_from_rust,
        0
    );
}
```

```
$ rustc \
    --crate-type=cdylib \
    simplest_rust_extension.rs \
    -o simplest_rust_extension.bundle \
    -C link-arg="-Wl,-undefined,dynamic_lookup"

$ ruby \
    -r ./simplest_rust_extension \
    -e "puts hello_from_rust"
#=> "Hello, world!"
```

How it works

```
$ ruby -e "puts hello_from_rust"  
#=> "Hello, world!"
```

How it works

```
$ ruby \
  -r ./simplest_rust_extension \
  -e "puts hello_from_rust"
#=> "Hello, world!"
```

How it works

```
$ rustc \
  --crate-type=cdylib \
  simplest_rust_extension.rs \
-o simplest_rust_extension.bundle \
-C link-arg="-Wl,-undefined,dynamic_lookup"

$ ruby \
  -r ./simplest_rust_extension \
  -e "puts hello_from_rust"
#=> "Hello, world!"
```

How it works

```
#[no_mangle]
unsafe extern "C" fn Init_simplest_rust_extension() {
    rb_define_global_function(
        "hello_from_rust\0".as_ptr(),
        hello_from_rust,
        0
    );
}
```

```
$ rustc \
    --crate-type=cdylib \
    simplest_rust_extension.rs \
    -o simplest_rust_extension.bundle \
    -C link-arg="-Wl,-undefined,dynamic_lookup"

$ ruby \
    -r ./simplest_rust_extension \
    -e "puts hello_from_rust"
#=> "Hello, world!"
```

How it works

```
extern "C" fn hello_from_rust() -> usize {
    unsafe { rb_str_new("Hello, world!".as_ptr(), 12) }
}

#[no_mangle]
unsafe extern "C" fn Init_simplest_rust_extension() {
    rb_define_global_function(
        "hello_from_rust\0".as_ptr(),
        hello_from_rust,
        0
    );
}
```

```
$ rustc \
    --crate-type=cdylib \
    simplest_rust_extension.rs \
    -o simplest_rust_extension.bundle \
    -C link-arg="-Wl,-undefined,dynamic_lookup"

$ ruby \
    -r ./simplest_rust_extension \
    -e "puts hello_from_rust"
#=> "Hello, world!"
```

How it works

```
extern "C" {
    fn rb_define_global_function(
        name: *const u8,
        func: extern "C" fn() -> usize,
        arity: isize
    );
    fn rb_str_new(
        ptr: *const u8,
        len: isize
    ) -> usize;
}

extern "C" fn hello_from_rust() -> usize {
    unsafe { rb_str_new("Hello, world!".as_ptr(), 12) }
}

#[no_mangle]
unsafe extern "C" fn Init_simplest_rust_extension() {
    rb_define_global_function(
        "hello_from_rust\0".as_ptr(),
        hello_from_rust,
        0
    );
}
```

```
$ rustc \
    --crate-type=cdylib \
    simplest_rust_extension.rs \
    -o simplest_rust_extension.bundle \
    -C link-arg="-Wl,-undefined,dynamic_lookup"

$ ruby \
    -r ./simplest_rust_extension \
    -e "puts hello_from_rust"
#=> "Hello, world!"
```

Magnus and rb-sys

High-level Ruby bindings for Rust

- Memory management
- Error handling
- Type conversions
- Performance optimizations

Magnus and rb-sys

High-level Ruby bindings for Rust

- Memory management
- Error handling
- Type conversions
- Performance optimizations

Magnus and rb-sys

Low-level Rust bindings to the Ruby C API

- Provides core utilities for creating extensions
- Runtime support
- Safety mechanisms
- Basic interoperability
- Raw/unsafe bindings to libruby

TOML prototype

TOML prototype

```
fn parse(str: String) -> RubyValue {  
    let rust_value = toml.parse(str);  
    rust_value.to_ruby()  
}
```

```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
  > 📂 fast_toml
    > 📂 src
      📜 lib.rs
      [T] Cargo.toml
      📜 extconf.rb
> 📂 lib
> 📂 rakelib
> 📂 test
  📜 CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  [T] Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📄 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
1 use magnus::{function, prelude::*, Error, IntoValue, RArray, RHash, Ruby}
1
2 #[magnus::init]
3 fn init(ruby: &Ruby) -> Result<(), Error> {
4     let module = ruby.define_module("FastToml")?;
5     module.define_singleton_method("parse", function!(parse, 1))?;
6     Ok(())
7 }
8
9 fn parse(str: String) -> magnus::Value {
10     let toml_value: toml::Value = toml::from_str(&str).unwrap();
11
12     to_ruby_value(toml_value)
13 }
14
15 fn to_ruby_value(value: toml::Value) -> magnus::Value {
16     match value {
17         toml::Value::String(string) => string.into_value(),
18         toml::Value::Integer(integer) => integer.into_value(),
19         toml::Value::Float(float) => float.into_value(),
20         toml::Value::Boolean(boolean) => boolean.into_value(),
21         toml::Value::Datetime(date) => date.to_string().into_value(),
22         toml::Value::Array(rust_array) => {
23             let ruby_array = RArray::new();
24
25             for value in rust_array {
26                 let value = to_ruby_value(value);
27                 ruby_array.push(value).unwrap();
28             }
29         }
30     }
31 }
```

```
> └─ lib
> └─ rakelib
> └─ test
  └─ CODE_OF_CONDUCT.md
  └─ Cargo.lock
  └─ Cargo.toml
  * Gemfile
  └─ Gemfile.lock
  └─ LICENSE.txt
  └─ README.md
  └─ Rakefile
  └─ fast_toml.gemspec
(7 hidden items)
```

```
29
28 fn to_ruby_value(value: toml::Value) -> magnus::Value {
27     match value {
26         toml::Value::String(string) => string.into_value(),
25         toml::Value::Integer(integer) => integer.into_value(),
24         toml::Value::Float(float) => float.into_value(),
23         toml::Value::Boolean(boolean) => boolean.into_value(),
22         toml::Value::Datetime(date) => date.to_string().into_value(),
21         toml::Value::Array(rust_array) => {
20             let ruby_array = RArray::new();
19
18             for value in rust_array {
17                 let value = to_ruby_value(value);
16                 ruby_array.push(value).unwrap();
15             }
14
13             ruby_array.into_value()
12         }
11         toml::Value::Table(rust_map) => {
10             let ruby_hash = RHash::new();
9
8             for (key, value) in rust_map {
7                 let value = to_ruby_value(value);
6                 ruby_hash.aset(key, value).unwrap();
5             }
4
3             ruby_hash.into_value()
2         }
1     }
44 }
```

Benchmarks

```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
> 📂 lib
> 📂 rakelib
  └── 📜 benchmark.rake
> 📂 test
  📜 CODE_OF_CONDUCT.md
  📜 Cargo.lock
  [T] Cargo.toml
  * Gemfile
  📜 Gemfile.lock
  📜 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
24   require 'benchmark'
23   require 'benchmark/ips'
22   require 'toml'
21   require 'toml-rb'
20   require 'perfect_toml'
19   require 'fast_toml'
18   require 'tomlrb'

17
16 TOML_INPUT = <<~TOML
15   [package]
14     name = "fast_toml"
13     version = "0.1.0"
12     edition = "2021"
11     authors = ["Guilherme Carreiro <karreiro@gmail.com>"]
10     license = "MIT"
9      publish = false

8
7   [lib]
6     crate-type = ["cdylib"]

5
4   [dependencies]
3     magnus = "0.6.4"
2     toml = "0.8.13"
1   TOML

27
1   namespace :benchmark do
2     desc 'Benchmark different TOML parsers (ips)'
3     task :ips do
4       Benchmark.ips do |x|
5         x.report('fast_toml') { FastToml.parse(TOML_INPUT) }
```

```
* Gemfile
❷ Gemfile.lock
❸ LICENSE.txt
❹ README.md
❺ Rakefile
❻ fast_toml.gemspec
(7 hidden items)
```

```
27
1 namespace :benchmark do
2   desc 'Benchmark different TOML parsers (ips)'
3   task :ips do
4     Benchmark.ips do |x|
5       x.report('fast_toml')    { FastToml.parse(TOML_INPUT) }
6       x.report('toml')        { TOML::Parser.new(TOML_INPUT).parsed }
7       x.report('toml-rb')     { TomlRB.parse(TOML_INPUT) }
8       x.report('perfect_toml') { PerfectTOML.parse(TOML_INPUT) }
9       x.report('tomlrb')      { Tomlrb.parse(TOML_INPUT) }
10
11     x.compare!
12   end
13 end
14
15 desc 'Benchmark different TOML parsers (ms)'
16 task :ms do
17   n = 5_000
18
19   Benchmark.bm(20) do |x|
20     x.report('fast_toml:')    { n.times { FastToml.parse(TOML_INPUT) } }
21     x.report('toml:')        { n.times { TOML::Parser.new(TOML_INPUT).parsed } }
22     x.report('toml-rb:')     { n.times { TomlRB.parse(TOML_INPUT) } }
23     x.report('perfect_toml:') { n.times { PerfectTOML.parse(TOML_INPUT) } }
24     x.report('tomlrb:')      { n.times { Tomlrb.parse(TOML_INPUT) } }
25   end
26 end
27 end
```

```
NORMAL ➤ ↗ main ➤ ⌂ rakelib/benchmark.rake
① ▶ 1·nvim
```

```
$ rake benchmark:ips
ruby 3.3.0 (2023-12-25 revision 5124f9ac75) [arm64-darwin23]
Warming up -----
```

fast_toml	12.107k	i/100ms
toml	60.000	i/100ms
toml-rb	60.000	i/100ms
perfect_toml	4.136k	i/100ms
tomlrb	730.000	i/100ms

```
Calculating -----
```

fast_toml	120.849k	(± 0.7%) i/s -	605.350k	in	5.009348s
toml	609.228	(± 1.0%) i/s -	3.060k	in	5.023174s
toml-rb	610.812	(± 1.5%) i/s -	3.060k	in	5.010792s
perfect_toml	41.351k	(± 0.2%) i/s -	206.800k	in	5.001107s
tomlrb	7.250k	(± 0.6%) i/s -	36.500k	in	5.034708s

Comparison:

fast_toml:	120849.5	i/s
perfect_toml:	41351.0	i/s - 2.92x slower
tomlrb:	7249.9	i/s - 16.67x slower
toml-rb:	610.8	i/s - 197.85x slower
toml:	609.2	i/s - 198.36x slower

```
❷ Cargo.lock          2     publish = false
❸ Cargo.toml          1
❹ Gemfile              20    [lib]
❺ Gemfile.lock         1      crate-type = ["cdylib"]
❻ LICENSE.txt          2
❼ README.md            3      [dependencies]
⽿ Rakefile             4      magnus = "0.6.4"
⽾ fast_toml.gemspec   5      toml = "0.8.13"
⽿ TOML                6
⽿
⽿
⽿ namespace :benchmark do
⽿   desc 'Benchmark different TOML parsers (ips)'
⽿   task :ips do
⽿     Benchmark.ips do |x|
⽿       x.report('fast_toml')    { FastToml.parse(TOML_INPUT) }
⽿       x.report('toml')        { TOML::Parser.new(TOML_INPUT).parsed }
⽿       x.report('toml-rb')     { TomlRB.parse(TOML_INPUT) }
⽿       x.report('perfect_toml') { PerfectTOML.parse(TOML_INPUT) }
⽿       x.report('tomlrb')      { Tomlrb.parse(TOML_INPUT) }
⽿
⽿     x.compare!
```

NORMAL ➔ ↵ main ➔ 📄 rakelib/benchmark.rake

```
$ bundle exec rake benchmark:ms
              user      system      total      real
fast_toml:      0.041813  0.000127  0.041940 ( 0.042178)
toml:          9.060186  0.016067  9.076253 ( 9.096280)
toml-rb:        8.939877  0.007595  8.947472 ( 8.963423)
perfect_toml:   0.103443  0.000128  0.103571 ( 0.103828)
tomlrb:         0.531844  0.000733  0.532577 ( 0.533547)
```

\$ █

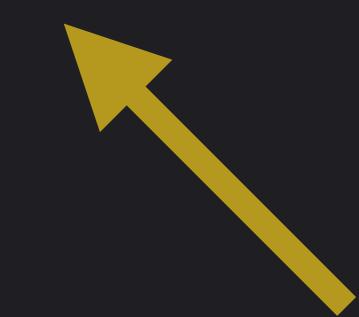
⌚ ▶ 1·ruby

Error handling

thread '`<unnamed>`' panicked at ext/fast_toml/src/lib.rs:11:56

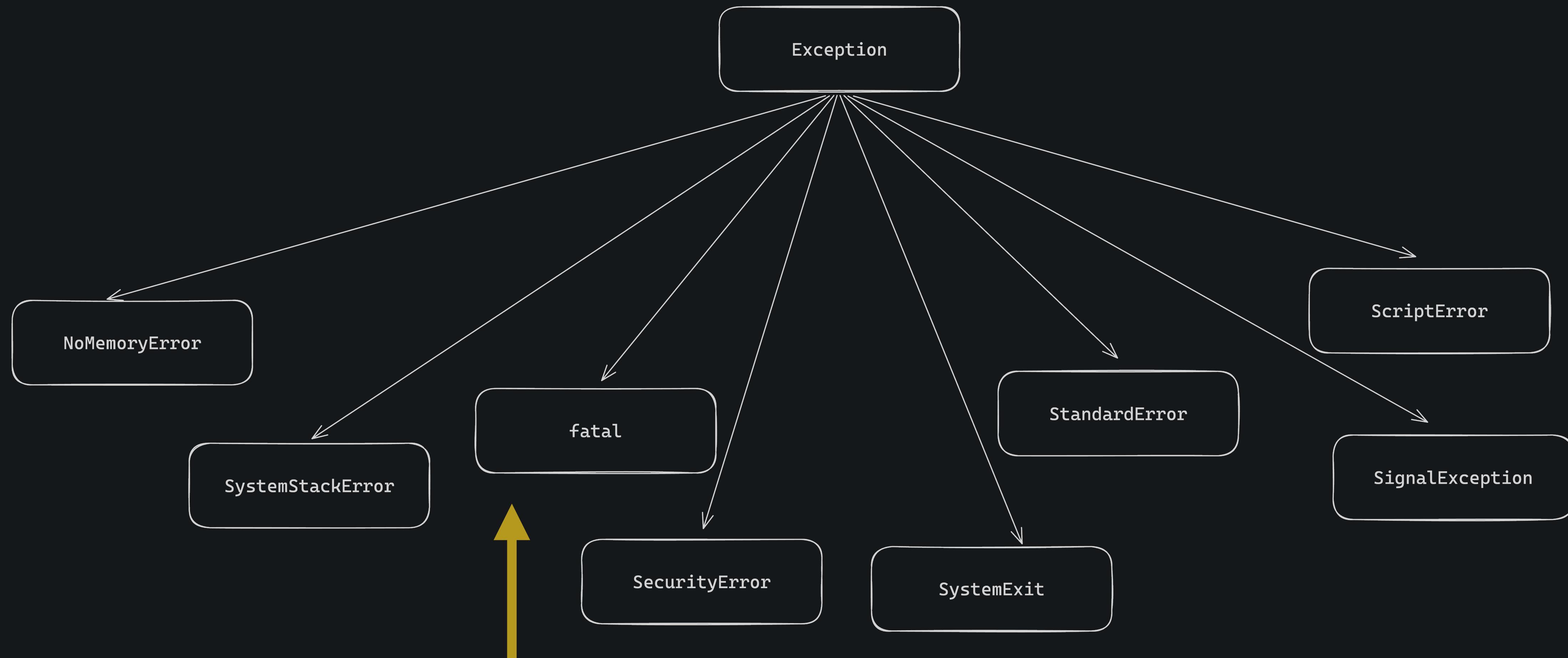
```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
  > 📂 fast_toml
    > 📂 src
      📜 lib.rs
      [T] Cargo.toml
      📜 extconf.rb
> 📂 lib
> 📂 rakelib
> 📂 test
  📜 CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  [T] Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📄 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
1 use magnus::{function, prelude::*, Error, IntoValue, RArray, RHash, Ruby}
1
2 #[magnus::init]
3 fn init(ruby: &Ruby) -> Result<(), Error> {
4     let module = ruby.define_module("FastToml")?;
5     module.define_singleton_method("parse", function!(parse, 1))?;
6     Ok(())
7 }
8
9 fn parse(str: String) -> magnus::Value {
10     let toml_value: toml::Value = toml::from_str(&str).unwrap();
11
12     to_ruby_value(toml_value)
13 }
14
15 fn to_ruby_value(value: toml::Value) -> magnus::Value {
16     match value {
17         toml::Value::String(string) => string.into_value(),
18         toml::Value::Integer(integer) => integer.into_value(),
19         toml::Value::Float(float) => float.into_value(),
20         toml::Value::Boolean(boolean) => boolean.into_value(),
21         toml::Value::Datetime(date) => date.to_string().into_value(),
22         toml::Value::Array(rust_array) => {
23             let ruby_array = RArray::new();
24
25             for value in rust_array {
26                 let value = to_ruby_value(value);
27                 ruby_array.push(value).unwrap();
28             }
29         }
30     }
31 }
```



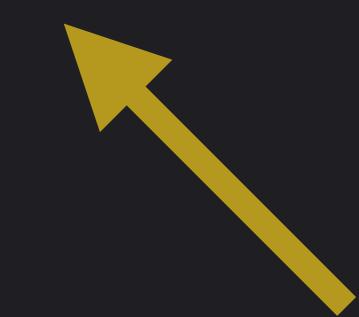
\$





```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
  > 📂 fast_toml
    > 📂 src
      📜 lib.rs
      [T] Cargo.toml
      📜 extconf.rb
> 📂 lib
> 📂 rakelib
> 📂 test
  📜 CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  [T] Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📄 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
1 use magnus::{function, prelude::*, Error, IntoValue, RArray, RHash, Ruby}
1
2 #[magnus::init]
3 fn init(ruby: &Ruby) -> Result<(), Error> {
4     let module = ruby.define_module("FastToml")?;
5     module.define_singleton_method("parse", function!(parse, 1))?;
6     Ok(())
7 }
8
9 fn parse(str: String) -> magnus::Value {
10     let toml_value: toml::Value = toml::from_str(&str).unwrap();
11
12     to_ruby_value(toml_value)
13 }
14
15 fn to_ruby_value(value: toml::Value) -> magnus::Value {
16     match value {
17         toml::Value::String(string) => string.into_value(),
18         toml::Value::Integer(integer) => integer.into_value(),
19         toml::Value::Float(float) => float.into_value(),
20         toml::Value::Boolean(boolean) => boolean.into_value(),
21         toml::Value::Datetime(date) => date.to_string().into_value(),
22         toml::Value::Array(rust_array) => {
23             let ruby_array = RArray::new();
24
25             for value in rust_array {
26                 let value = to_ruby_value(value);
27                 ruby_array.push(value).unwrap();
28             }
29         }
30     }
31 }
```



```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
  > 📂 fast_toml
    > 📂 src
      📜 lib.rs
      📜 Cargo.toml
      📜 extconf.rb
> 📂 lib
> 📂 rakelib
> 📂 test
  📜 CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  📜 Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📜 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
1 use magnus::exception::runtime_error;
1 use magnus::function;
2 use magnus::prelude::*;
3 use magnus::value::{InnerValue, Lazy};
4 use magnus::RModule;
5 use magnus::{IntoValue, RArray, RHash, Ruby};
6
7 #[magnus::init]
8 fn init(ruby: &Ruby) -> Result<(), magnus::Error> {
9     let module = ruby.define_module("FastToml")?;
10    module.define_singleton_method("parse", function!(parse, 1))?;
11    Ok(())
12 }
13
14 fn parse(str: String) -> Result<magnus::Value, magnus::Error> {
15     let toml_value = toml::from_str(&str);
16
17     match toml_value {
18         Ok(toml_value) => {
19             let ruby_value = to_ruby_value(toml_value);
20             Ok(ruby_value)
21         }
22         Err(err) => {
23             let ruby_err = to_ruby_err(err.message());
24             Err(ruby_err)
25         }
26     }
27 }
28
29 // ...
```

```
> lib
> rakelib
> test
M CODE_OF_CONDUCT.md
L Cargo.lock
T Cargo.toml
* Gemfile
L Gemfile.lock
L LICENSE.txt
M README.md
R Rakefile
R fast_toml.gemspec
(7 hidden items)
```

```
4 }
3
2 // ...
1
32 fn to_ruby_err(err_msg: &str) -> magnus::Error {
1     let ruby_api = Ruby::get();
2
3     if let Ok(ruby_api) = ruby_api {
4         let module = fast_toml_module(&ruby_api);
5
6         let err_class = module
7             .define_error("Error", ruby_api.exception_standard_error())
8             .expect("Failed to define or get FastToml::Error");
9
10        return magnus::Error::new(err_class, err_msg.to_string());
11    }
12
13    magnus::Error::new(runtime_error(), "FastToml runtime err")
14}
15
16 fn fast_toml_module(ruby_api: &Ruby) -> RModule {
17     Lazy::new(|ruby| {
18         ruby.define_module("FastToml")
19             .expect("Failed to define or get FastToml")
20     })
21     .get_inner_with(ruby_api)
22 }
23
24 // ...
25
```

```
NORMAL ➤ ↵ main ➤ ⌂ ext/.../src/lib.rs > f to_ruby_err f to_ruby_err
```

```
📁 ~/Projects/fast_toml
> 📂 bin
> 📂 ext
  > 📂 fast_toml
    > 📂 src
      📜 lib.rs
      📜 Cargo.toml
      📜 extconf.rb
> 📂 lib
> 📂 rakelib
> 📂 test
  📜 CODE_OF_CONDUCT.md
  🔒 Cargo.lock
  📜 Cargo.toml
  * Gemfile
  🔒 Gemfile.lock
  📜 LICENSE.txt
  📜 README.md
  📜 Rakefile
  📜 fast_toml.gemspec
(7 hidden items)
```

```
1 use magnus::exception::runtime_error;
1 use magnus::function;
2 use magnus::prelude::*;
3 use magnus::value::{InnerValue, Lazy};
4 use magnus::RModule;
5 use magnus::{IntoValue, RArray, RHash, Ruby};
6
7 #[magnus::init]
8 fn init(ruby: &Ruby) -> Result<(), magnus::Error> {
9     let module = ruby.define_module("FastToml")?;
10    module.define_singleton_method("parse", function!(parse, 1))?;
11    Ok(())
12 }
13
14 fn parse(str: String) -> Result<magnus::Value, magnus::Error> {
15     let toml_value = toml::from_str(&str);
16
17     match toml_value {
18         Ok(toml_value) => {
19             let ruby_value = to_ruby_value(toml_value);
20             Ok(ruby_value)
21         }
22         Err(err) => {
23             let ruby_err = to_ruby_err(err.message());
24             Err(ruby_err)
25         }
26     }
27 }
28
29 // ...
```

```
$ bin/console
irb(main):001:0> FastToml.parse("[")
(irb):1:in `parse': invalid table header (FastToml::Error)
  from (irb):1:in `<main>'
  from bin/console:10:in `<main>'
irb(main):002:0> █
```

Memory & GC

Memory leaks

```
use std::mem::ManuallyDrop;
use std::ptr;

fn main() {
    let data = Box::new(42);
    let raw = Box::into_raw(data);

    unsafe {
        ManuallyDrop::new(ptr::read(raw));
    }
}
```

Information

- About
- News
- Tool Suite
- Supported Platforms
- The Developers

Source Code

- Current Releases
- Release Archive
- Variants / Patches
- Code Repository
- Valkyrie / GUIs

Documentation

- Table of Contents
- Quick Start
- FAQ
- User Manual
- Download Manual
- Research Papers
- Books

Contact

- Mailing Lists and IRC
- Bug Reports
- Feature Requests
- Contact Summary
- Commercial Support

How to Help

- Contributing
- Project Suggestions

Gallery

- Projects / Users
- Press / Media
- Awards
- Surveys
- Artwork / Clothing

Valgrind



Current release: [valgrind-3.23.0](#)

Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. You can also use Valgrind to build new tools.

The Valgrind distribution currently includes seven production-quality tools: a memory error detector, two thread error detectors, a cache and branch-prediction profiler, a call-graph generating cache and branch-prediction profiler, and two different heap profilers. It also includes an experimental SimPoint basic block vector generator. It runs on the following platforms: X86/Linux, AMD64/Linux, ARM/Linux, ARM64/Linux, PPC32/Linux, PPC64/Linux, PPC64LE/Linux, S390X/Linux, MIPS32/Linux, MIPS64/Linux, X86/Solaris, AMD64/Solaris, ARM/Android (2.3.x and later), ARM64/Android, X86/Android (4.0 and later), MIPS32/Android, X86/FreeBSD, AMD64/FreeBSD, ARM64/FreeBSD, X86/Darwin and AMD64/Darwin (Mac OS X 10.12).

Valgrind is [Open Source / Free Software](#), and is freely available under the [GNU General Public License, version 2](#).

Recent News

- 26 April 2024: valgrind-3.23.0 is available. This release supports: X86/Linux, AMD64/Linux, ARM32/Linux, ARM64/Linux, PPC32/Linux, PPC64BE/Linux, PPC64LE/Linux, S390X/Linux, MIPS32/Linux, MIPS64/Linux, ARM/Android, ARM64/Android, MIPS32/Android, X86/Android, X86/Solaris, AMD64/Solaris, X86/FreeBSD, AMD64/FreeBSD, ARM64/FreeBSD, X86/MacOSX 10.12 and AMD64/MacOSX 10.12. For more details

Memory leaks

```
use std::mem::ManuallyDrop;
use std::ptr;

fn main() {
    let data = Box::new(42);
    let raw = Box::into_raw(data);

    unsafe {
        ManuallyDrop::new(ptr::read(raw));
    }
}
```

```
$ rustc mem.rs -o mem

$ valgrind --leak-check=full ./mem

==8190== Memcheck, a memory error detector
==8190== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
...
==8190== HEAP SUMMARY:
==8190==     in use at exit: 4 bytes in 1 blocks
==8190==   total heap usage: 11 allocs, 10 frees, 2,161 by
==8190==   4 bytes in 1 blocks are definitely lost in loss
==8190==       at 0x4848899: malloc (vg_replace_malloc.c:381)
==8190==       by 0x1105F5: alloc::alloc::alloc (in mem)
==8190==       by 0x1106DD: alloc::alloc::Global::alloc_impl
==8190==       by 0x110563: alloc::alloc::exchange_malloc (in
==8190==       by 0x110880: mem::main (in mem)
==8190==       by 0x110462: core::ops::function::FnOnce::call
==8190==       by 0x110285: std::sys_common::backtrace::__rust
==8190==       by 0x1102D8: std::rt::lang_start::{closure}
==8190==       by 0x126F42: std::rt::lang_start_internal (fur
==8190==       by 0x1102B6: std::rt::lang_start (in mem)
==8190==       by 0x1108E4: main (in mem)
==8190==

==8190== LEAK SUMMARY:
==8190==   definitely lost: 4 bytes in 1 blocks
==8190==   indirectly lost: 0 bytes in 0 blocks
==8190==       possibly lost: 0 bytes in 0 blocks
==8190==   still reachable: 0 bytes in 0 blocks
==8190==       suppressed: 0 bytes in 0 blocks
```

Memory leaks

```
$ valgrind --leak-check=full ruby main.rb

==55460== Memcheck, a memory error detector
==55460== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
...
==55460== LEAK SUMMARY:
==55460==   definitely lost: 2,614,221 bytes in 6,283 blocks
==55460==   indirectly lost: 1,027,215 bytes in 11,411 blocks
==55460==     possibly lost: 1,592 bytes in 7 blocks
==55460==   still reachable: 1,371,172 bytes in 908 blocks
==55460==           suppressed: 0 bytes in 0 blocks
```

A screenshot of a GitHub repository page for `Shopify / ruby_memcheck`. The repository is public and has 210 stars, 267 watchers, and 12 forks.

The main navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, and Insights.

The repository name `ruby_memcheck` is displayed with a green Shopify icon. The repository is marked as Public.

On the left, there's a sidebar with a dropdown for the main branch, a file browser icon, and a tag icon. It also shows a list of recent commits:

- peterzhu2118 Merge pull request #36 from Shopify/d... · 57f99a0 · 3 weeks ago
- .github/workflows Test on Ruby 3.3 · last month
- exe Add running on a Ruby script · 10 months ago
- lib Bump version to 3.0.0 · last month
- suppressions Suppress Regexp created in date__... · 10 months ago
- test Use only RUBY_FREE_AT_EXIT on s... · last month
- .gitignore Centralize ruby version definition to .r... · 2 months ago
- .rubocop.yml Centralize ruby version definition to .r... · 2 months ago

On the right, there's an **About** section with the following text:
Use Valgrind memcheck on your native gem without going crazy

Tags: ruby, valgrind

Links: Readme, MIT license, Code of conduct, Security policy, Activity, Custom properties, 210 stars, 267 watching

Publish it!

Publish it!

Compiling native extensions. This could take a while...

A screenshot of a web browser window showing the RubyGems.org website for the wasmtime gem. The page title is "All versions of wasmtime". The header includes a search bar, navigation links for Releases, Blog, Gems, Guides, and Karreiro, and a user profile icon.

All versions of wasmtime

256 VERSIONS SINCE NOVEMBER 01, 2023:

- 20.0.0** - May 31, 2024 aarch64-linux (95.7 MB)
- 20.0.0** - May 31, 2024 arm64-darwin (15.1 MB)
- 20.0.0** - May 31, 2024 x86_64-linux-musl (97.5 MB)
- 20.0.0** - May 31, 2024 x86_64-darwin (16.3 MB)
- 20.0.0** - May 31, 2024 aarch64-linux-musl (95.3 MB)
- 20.0.0** - May 31, 2024 x86_64-linux (97.5 MB)
- 20.0.0** - May 31, 2024 x64-mingw-ucrt (108 MB)
- 20.0.0** - May 31, 2024 (39.3 MB)
- 19.0.2** - May 15, 2024 aarch64-linux-musl (92.6 MB)
- 19.0.2** - May 15, 2024 arm64-darwin (14.5 MB)
- 19.0.2** - May 15, 2024 aarch64-linux (92.9 MB)
- 19.0.2** - May 15, 2024 x64-mingw-ucrt (106 MB)
- 19.0.2** - May 15, 2024 x86_64-linux-musl (95.8 MB)
- 19.0.2** - May 15, 2024 x86_64-linux (95.7 MB)
- 19.0.2** - May 15, 2024 x86_64-darwin (15.8 MB)
- 19.0.2** - May 15, 2024 (39 MB)
- 19.0.0** - May 13, 2024 x64-mingw-ucrt (106 MB)
- 19.0.0** - May 13, 2024 arm64-darwin (14.5 MB)

A screenshot of a web browser window showing the RubyGems.org website. The address bar at the top displays "rubygems.org/gems/wasmtime/versions". The page has a red header with navigation links for "RELEASES", "BLOG", "GEMS", "GUIDES", and "KARREIRO". A search bar with the placeholder "Search Gems..." and a magnifying glass icon is also present. On the right side of the header, there is a user profile picture and a dropdown arrow.

All versions of wasmtime

256 VERSIONS SINCE NOVEMBER 01, 2023:

- 20.0.0** - May 31, 2024 aarch64-linux (95.7 MB)
- 20.0.0** - May 31, 2024 arm64-darwin (15.1 MB)
- 20.0.0** - May 31, 2024 x86_64-linux-musl (97.5 MB)
- 20.0.0** - May 31, 2024 x86_64-darwin (16.3 MB)
- 20.0.0** - May 31, 2024 aarch64-linux-musl (95.3 MB)
- 20.0.0** - May 31, 2024 x86_64-linux (97.5 MB)
- 20.0.0** - May 31, 2024 x64-mingw-ucrt (108 MB)
- 20.0.0** - May 31, 2024 (39.3 MB)

- 19.0.2** - May 15, 2024 aarch64-linux-musl (92.6 MB)
- 19.0.2** - May 15, 2024 arm64-darwin (14.5 MB)
- 19.0.2** - May 15, 2024 aarch64-linux (92.9 MB)
- 19.0.2** - May 15, 2024 x64-mingw-ucrt (106 MB)
- 19.0.2** - May 15, 2024 x86_64-linux-musl (95.8 MB)
- 19.0.2** - May 15, 2024 x86_64-linux (95.7 MB)
- 19.0.2** - May 15, 2024 x86_64-darwin (15.8 MB)
- 19.0.2** - May 15, 2024 (39 MB)

- 19.0.0** - May 13, 2024 x64-mingw-ucrt (106 MB)
- 19.0.0** - May 13, 2024 arm64-darwin (14.5 MB)

rubygems.org/gems/wasmtime/versions

All versions of wasmtime | RubyGems.org | your community gem host

Search Gems... 

RELEASES BLOG GEMS GUIDES KARREIRO  

All versions of wasmtime

256 VERSIONS SINCE NOVEMBER 01, 2023:

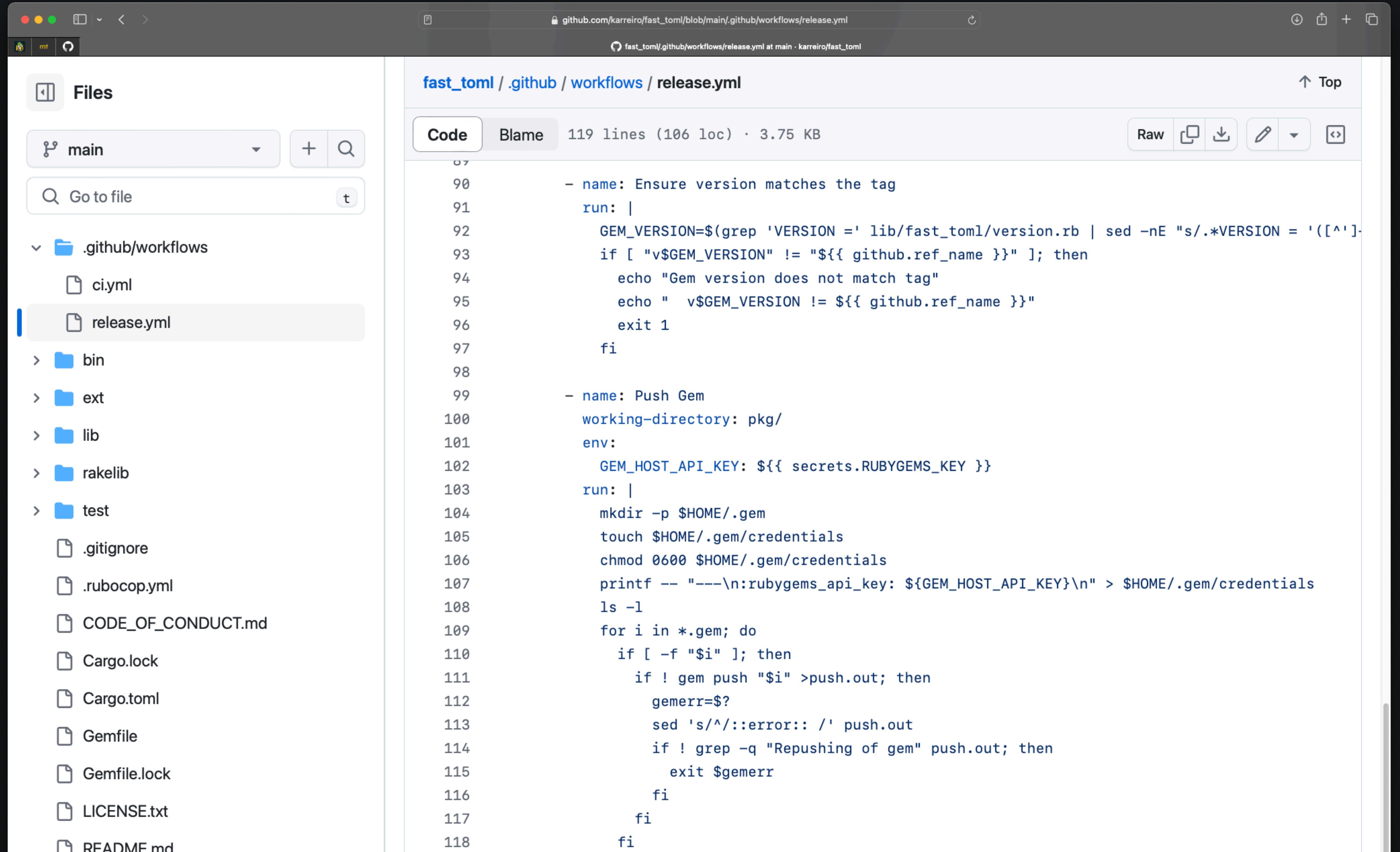
- 20.0.0** - May 31, 2024 aarch64-linux (95.7 MB)
- 20.0.0** - May 31, 2024 arm64-darwin (15.1 MB)
- 20.0.0** - May 31, 2024 x86_64-linux-musl (97.5 MB)
- 20.0.0** - May 31, 2024 x86_64-darwin (16.3 MB)
- 20.0.0** - May 31, 2024 aarch64-linux-musl (95.3 MB)
- 20.0.0** - May 31, 2024 x86_64-linux (97.5 MB)
- 20.0.0** - May 31, 2024 x64-mingw-ucrt (108 MB)
- 20.0.0** - May 31, 2024 (39.3 MB)

- 19.0.2** - May 15, 2024 aarch64-linux-musl (92.6 MB)
- 19.0.2** - May 15, 2024 arm64-darwin (14.5 MB)
- 19.0.2** - May 15, 2024 aarch64-linux (92.9 MB)
- 19.0.2** - May 15, 2024 x64-mingw-ucrt (106 MB)
- 19.0.2** - May 15, 2024 x86_64-linux-musl (95.8 MB)
- 19.0.2** - May 15, 2024 x86_64-linux (95.7 MB)
- 19.0.2** - May 15, 2024 x86_64-darwin (15.8 MB)
- 19.0.2** - May 15, 2024 (39 MB)

- 19.0.0** - May 13, 2024 x64-mingw-ucrt (106 MB)
- 19.0.0** - May 13, 2024 arm64-darwin (14.5 MB)

```
$ gem install wasmtime
Fetching wasmtime-20.0.0-arm64-darwin.gem
Successfully installed wasmtime-20.0.0-arm64-darwin
Parsing documentation for wasmtime-20.0.0-arm64-darwin
Installing ri documentation for wasmtime-20.0.0-arm64-darwin
Done installing documentation for wasmtime after 0 seconds
1 gem installed
$ █
```

⌚ ▶ 1 · zsh



A screenshot of a web browser window showing the RubyGems.org website. The address bar at the top displays the URL `rubygems.org/gems/fast_toml/versions`. The page has a red header bar with navigation links for **RELEASES**, **BLOG**, **GEMS**, **GUIDES**, and **KARREIRO**. On the left, there's a search bar labeled "Search Gems..." with a magnifying glass icon. A user profile picture is visible on the right.

All versions of fast_toml

10 VERSIONS SINCE MAY 26, 2024:

- 0.3.0** - June 06, 2024 x86_64-linux-musl (3.18 MB)
- 0.3.0** - June 06, 2024 x86_64-darwin (1020 KB)
- 0.3.0** - June 06, 2024 x86_64-linux (3.19 MB)
- 0.3.0** - June 06, 2024 aarch64-linux (3.15 MB)
- 0.3.0** - June 06, 2024 aarch64-linux-musl (3.13 MB)
- 0.3.0** - June 06, 2024 arm64-darwin (996 KB)
- 0.3.0** - June 06, 2024 (13.5 KB)
- 0.3.0** - June 06, 2024 x64-mingw-ucrt (5.2 MB)
- 0.2.0** - June 03, 2024 (13.5 KB)
- 0.1.0** - May 26, 2024 (11.5 KB)

A screenshot of a GitHub repository page for "karreiro / fast_toml".

The repository name is "fast_toml" and it is marked as "Private".

Key navigation links include: Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings.

On the right side, there are buttons for Unwatch (1), Fork (0), and Star (0).

The main content area shows the "Code" tab selected, displaying a list of recent commits:

Author	Commit Message	Date
karreiro	Update README.md	51148f0 · 2 hours ago
	.github/workflows	2 hours ago
	bin	last week
	ext/fast_toml	2 hours ago
	lib	2 hours ago
	rakelib	2 hours ago
	test	last week
	.gitignore	last week
	.rubocop.yml	last week
	CODE_OF_CONDUCT.md	last week
	Cargo.lock	2 hours ago

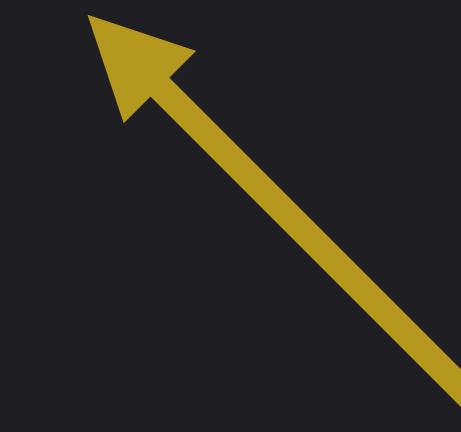
The "About" section indicates no description, website, or topics provided. It also lists the Readme, MIT license, Code of conduct, Activity, stars, watching, and forks counts.

The "Releases" section shows no releases published, with a link to "Create a new release".

The "Packages" section shows no packages published.

```
> lib
> rakelib
> test
  CODE_OF_CONDUCT.md
  Cargo.lock
  Cargo.toml
  Gemfile
  Gemfile.lock
  LICENSE.txt
  README.md
  Rakefile
  fast_toml.gemspec
(7 hidden items)
```

```
29
28 fn to_ruby_value(value: toml::Value) -> magnus::Value {
27     match value {
26         toml::Value::String(string) => string.into_value(),
25         toml::Value::Integer(integer) => integer.into_value(),
24         toml::Value::Float(float) => float.into_value(),
23         toml::Value::Boolean(boolean) => boolean.into_value(),
22         toml::Value::Datetime(date) => date.to_string().into_value(),
21         toml::Value::Array(rust_array) => {
20             let ruby_array = RArray::new();
29
19
18         for value in rust_array {
17             let value = to_ruby_value(value);
16             ruby_array.push(value).unwrap();
15         }
14
13         ruby_array.into_value()
12     }
11     toml::Value::Table(rust_map) => {
10         let ruby_hash = RHash::new();
9
8         for (key, value) in rust_map {
7             let value = to_ruby_value(value);
6             ruby_hash.aset(key, value).unwrap();
5         }
4
3         ruby_hash.into_value()
2     }
1
44 }
```







Thank you

karreiro.com/talks