CREDO

Analysing ASTs for Fun and Profit

René Föhring, @rrrene

neopoly

@rrrene

@rrrene

read in pirate voice

@rrrene

read in pirate voice

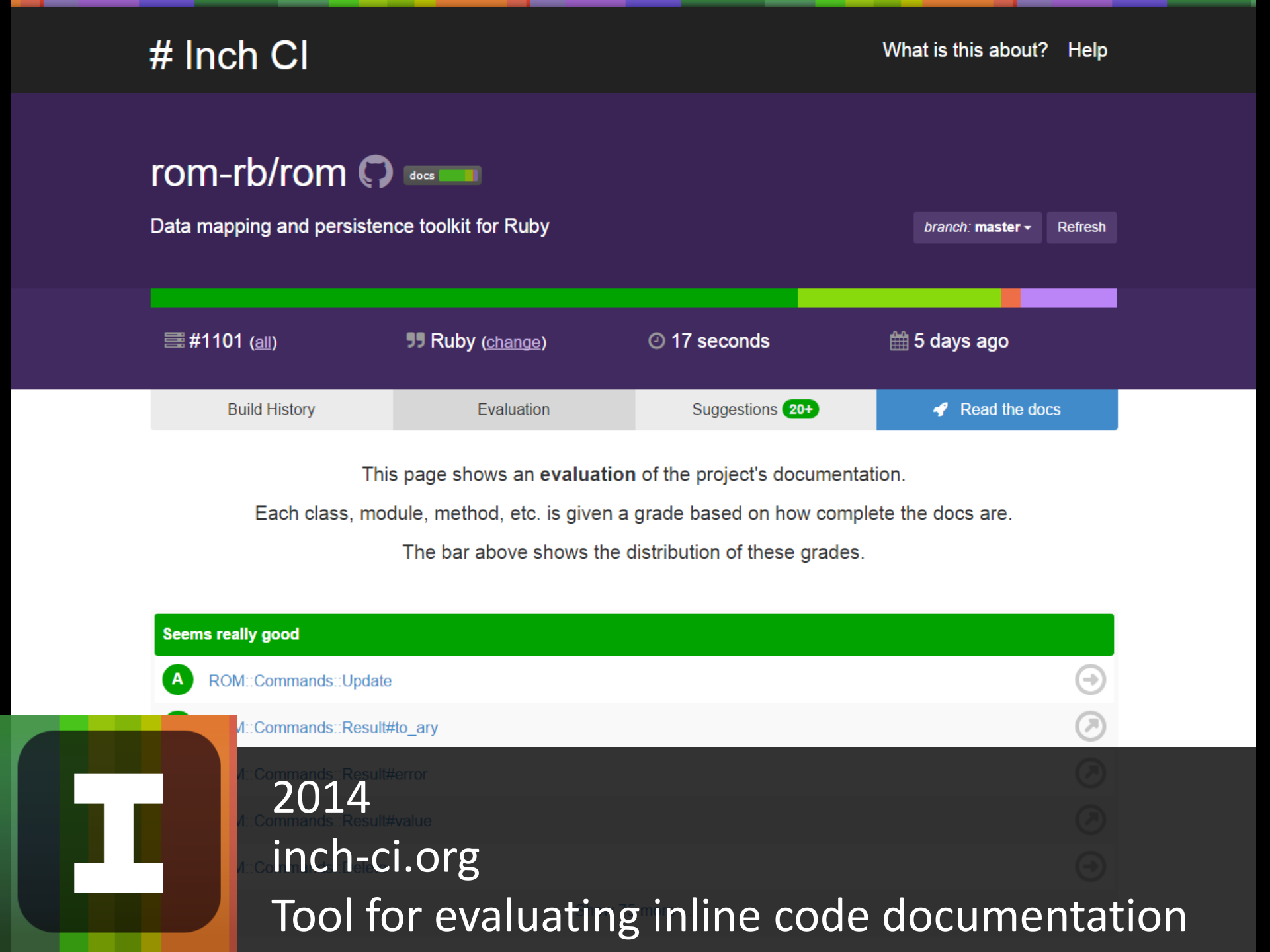
|> java |> ruby |> elixir

@rrrene

read in pirate voice

|> java |> ruby |> elixir

www.neopoly.com/jobs


Inch CI
What is this about? Help
rom-rb/rom
docs
Data mapping and persistence toolkit for Ruby
branch: master
Refresh
#1101 (all)
Ruby (change)
17 seconds
5 days ago
Build History
Evaluation
Suggestions 20+
Read the docs
This page shows an evaluation of the project's documentation.
Each class, module, method, etc. is given a grade based on how complete the docs are.
The bar above shows the distribution of these grades.
Seems really good
A
ROM::Commands::Update
M::Commands::Result#to_ary
M::Commands::Result#error
M::Commands::Result#value
2014
inch-ci.org
Tool for evaluating inline code documentation

elixirstatus

Announce your new project, blog post or version update.

Home

About

Sign in and post

Learning Elixir is one year old today!

13 Jul by @joekain

It's been one year since my first post on Learning Elixir.

I thought I would take this opportunity to look back over the last year and reflect and to look forward to the next year: http://learningelixir.joekain.com/learning-elixir-year-one-review/

The new, shiny Elixir Jobs

12 Jul by @rizafahmi

Elixir Jobs is the best place to find, list jobs and developer community space specifically for Elixir Programming Language.

Check out our new design: http://jobs.elixirdose.com/

Mongo.Ecto v0.1.0 released!

11 Jul by @michalmuskala

Mongo.Ecto is a MongoDB adapter for Ecto.

For detailed information for the Mongo.Ecto module check out the Github repo: https://github.com/michalmuskala/mongodb_ecto

2015

elixirstatus.com

Tool for community announcements

Docs

Community

Code

2014

inch-ci.org

2015

elixirstatus.com

2016

Credo

Today's topic!

Excursion:
What is static analysis?

```
11  def runApiCall(list, mode \\ nil) do
12    options = %{}
13    list |> IO.inspect
14    API.call(list, mode)
15  end
```

```
11  def runApiCall(list, mode \\ nil) do
12    options = %{}
13    list |> IO.inspect
14    API.call(list, mode)
15  end
```

```
$ mix compile
example.ex:11: warning: default arguments in run/2
are never used
example.ex:12: warning: variable options is unused
```

abstract syntax tree (AST)

{<operation>, <meta>, <arguments>}

abstract syntax tree (AST)

{<operation>, <meta>, <arguments>}

```
# 1 + 2
{:+, [line: 1], [1, 2]}
```

abstract syntax tree (AST)

```
{<operation>, <meta>, <arguments>}
# 1 + 2
{:+, [line: 1], [1, 2]}
# 42 |> inspect()
{:|>, [line: 1], [42, {:inspect, [line: 1], []}]}
```

abstract syntax tree (AST)

```
{<operation>, <meta>, <arguments>}
# 1 + 2
{:+, [line: 1], [1, 2]}
# 42 |> inspect()
{:|>, [line: 1], [42, {:inspect, [line: 1], []}]}
# try it yourself
Code.string_to_quoted!
```

Ruby - RuboCop

Python - pylint

JavaScript - ESLint

Elixir - ?

Ruby - RuboCop

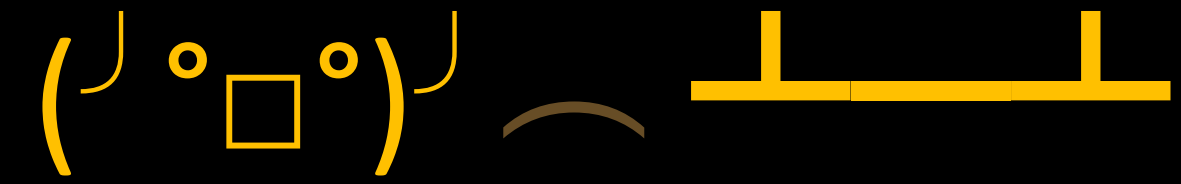
Python - pylint

JavaScript - ESLint

Elixir - Dogma

```
$ mix dogma
Inspecting 78 files.

.X....XXX.XX.XX.XXXXXXXXXXXXX.XXXXXXXXXXXXXXXXXXXXXXX.XXXXXXXXXXXXXXXXXXXXXXX

78 files, 624 errors!

== lib/plug/adapters/cowboy/conn.ex ==
92: CommentFormat: Comments should start with a single space
82: CommentFormat: Comments should start with a single space
121: FunctionArity: Arity of `parse_multipart` should be less than 4 (was 5).
117: FunctionArity: Arity of `parse_multipart` should be less than 4 (was 5).
38: FunctionArity: Arity of `send_file` should be less than 4 (was 6).
47: LineLength: Line length should not exceed 80 chars (was 92).
49: LineLength: Line length should not exceed 80 chars (was 97).
```

Dogma v0.1.5 analysing Plug v1.2.0-dev (output abbreviated)



But what is
the problem?

Elixir's recent rise from
totally unknown to
still definitely unknown but
mentioned in hushed tones

Clark Kampfe in "Elixir is not Ruby"
https://zeroclarkthirty.com/2015-11-01-elixir-is-not-ruby.html

What‘s the newcomer experience?

José Valim
@josevalim

Following

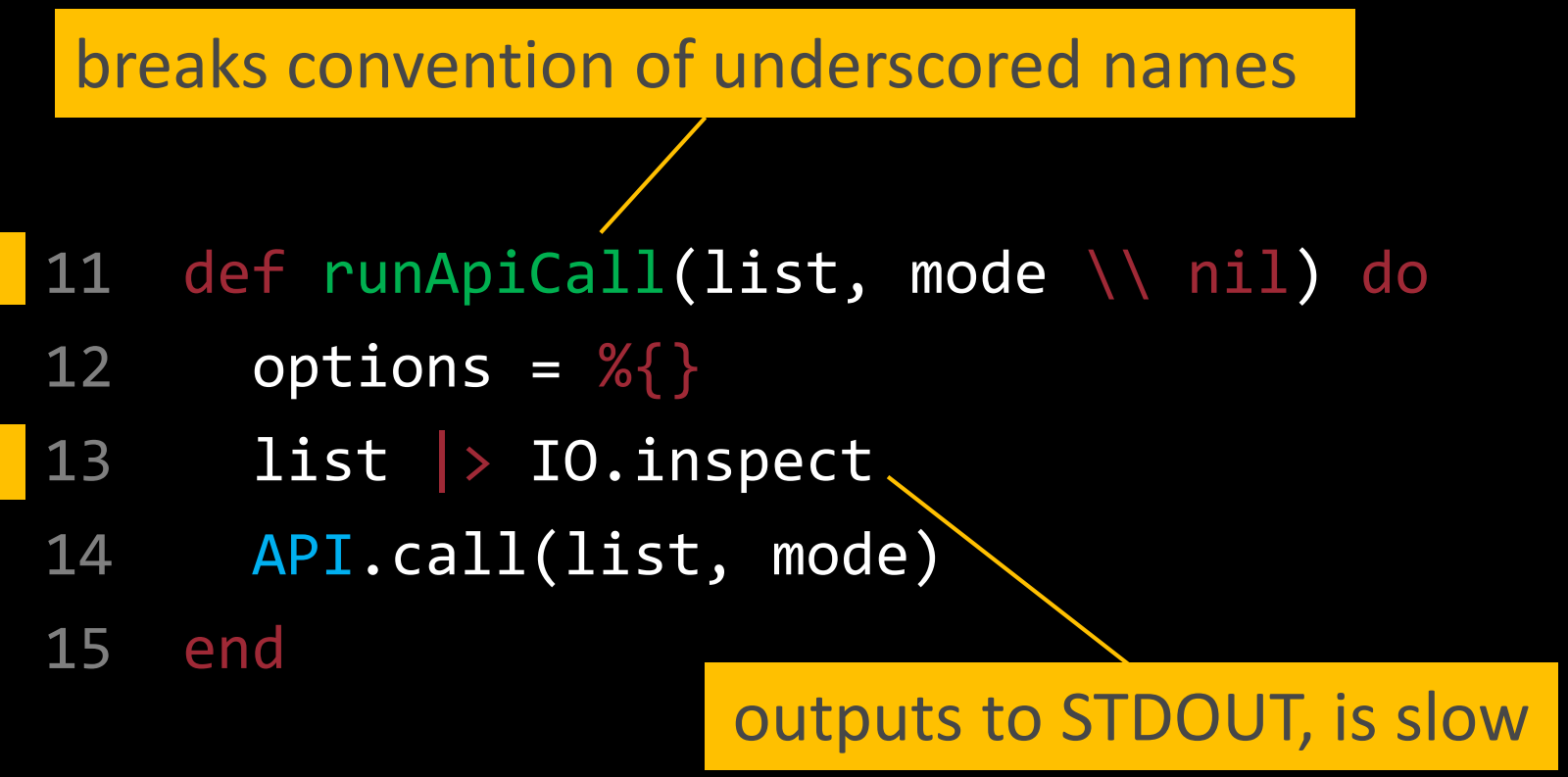
Maybe "you are wrong" is enough for talented singers and developers. It is really far from enough for the majority of us. We need guiding.

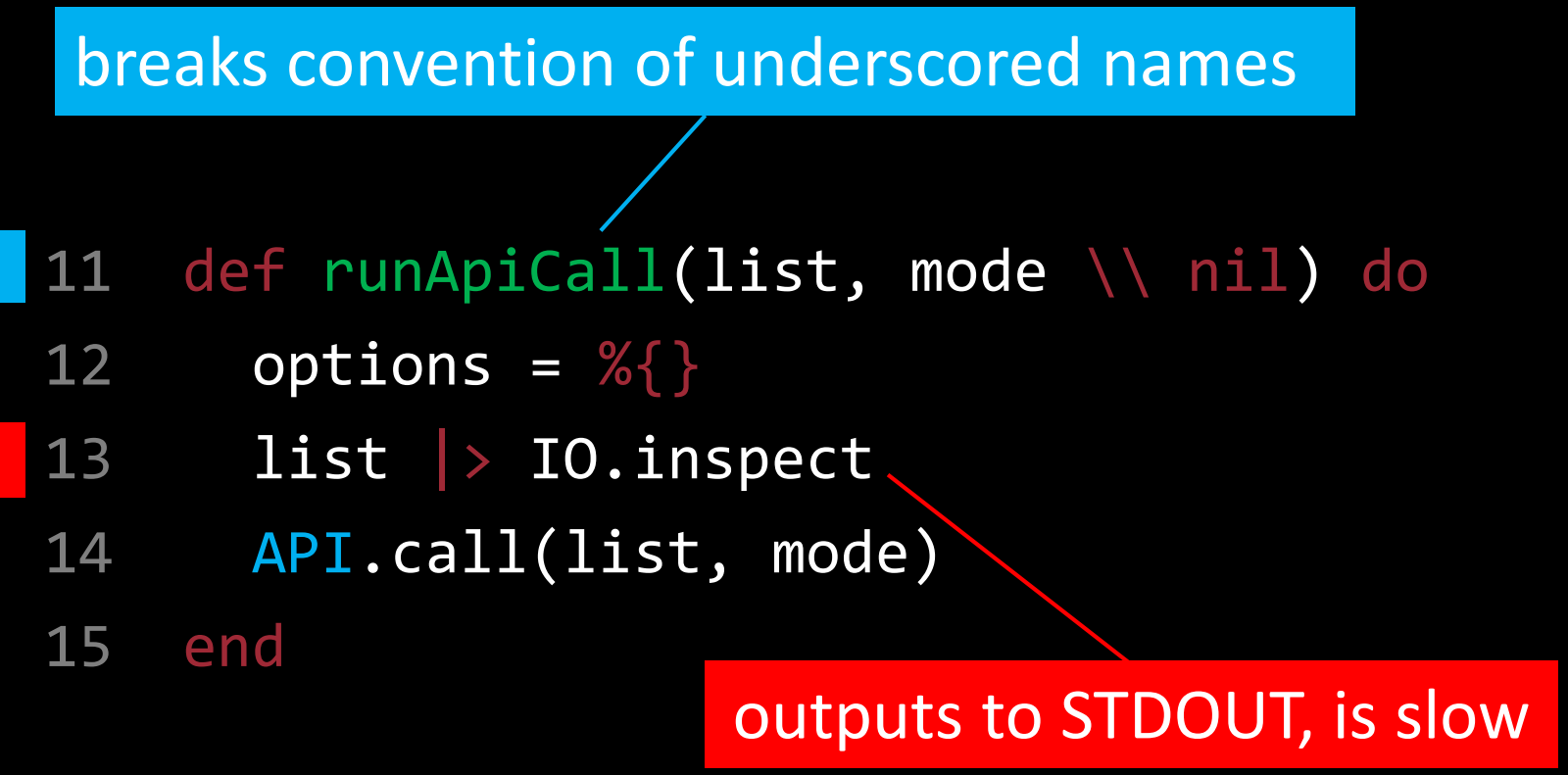
RETWEETS 17 LIKES 32

3:00 PM - 1 Nov 2015

Let‘s create
a mentoring tool.

```
11  def runApiCall(list, mode \\ nil) do
12    options = %{}
13    list |> IO.inspect
14    API.call(list, mode)
15  end
```


breaks convention of underscored names
11 def runApiCall(list, mode \\ nil) do
12 options = %{}
13 list |> IO.inspect
14 API.call(list, mode)
15 end
outputs to STDOUT, is slow


breaks convention of underscored names
11 def runApiCall(list, mode \\ nil) do
12 options = %{}
13 list |> IO.inspect
14 API.call(list, mode)
15 end
outputs to STDOUT, is slow

First, we need to categorize issues.

readability

readability

software design

readability

software design

refactoring opportunities

readability

software design

refactoring opportunities

consistency checks

readability

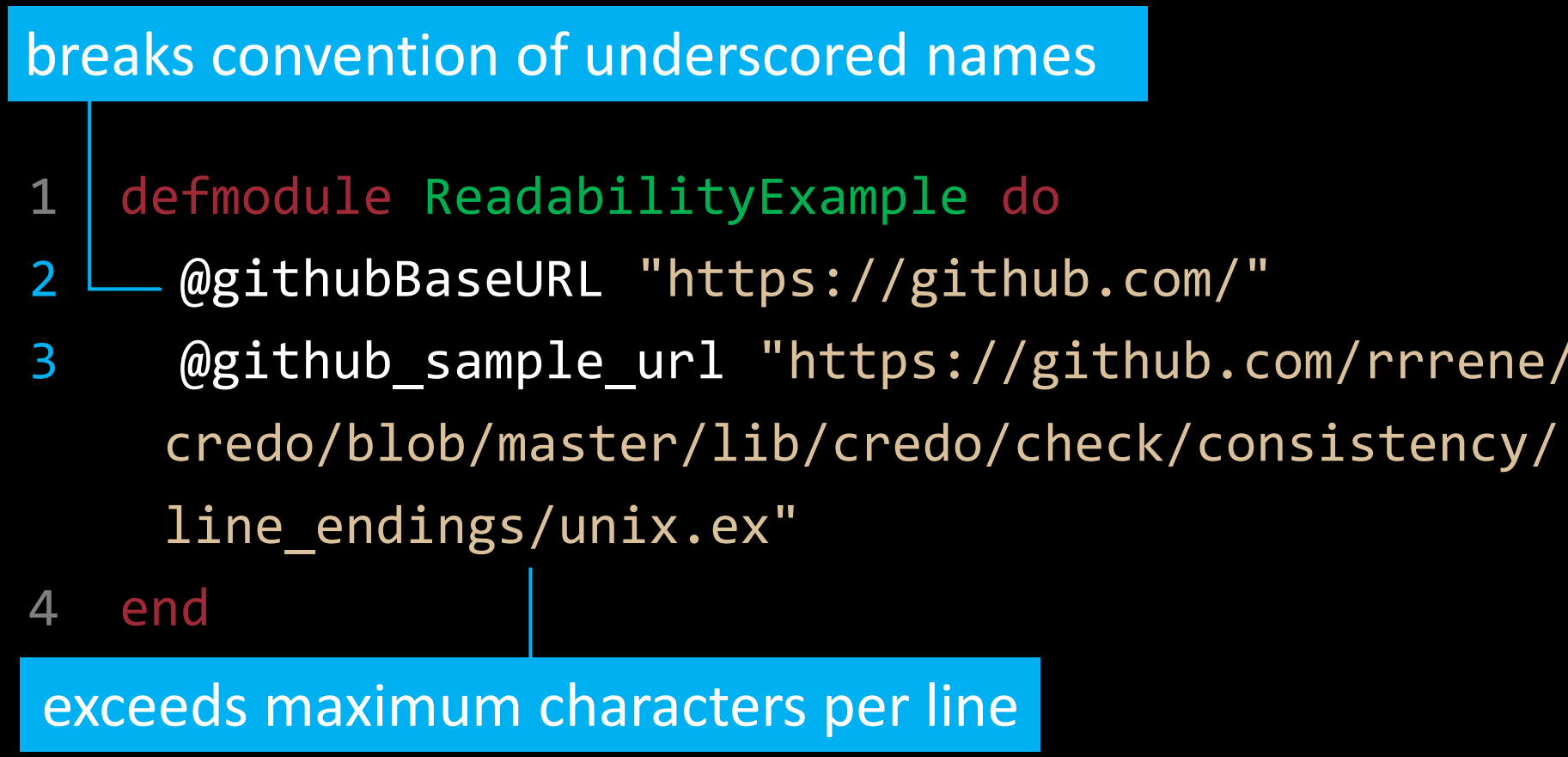
software design

refactoring opportunities

consistency checks

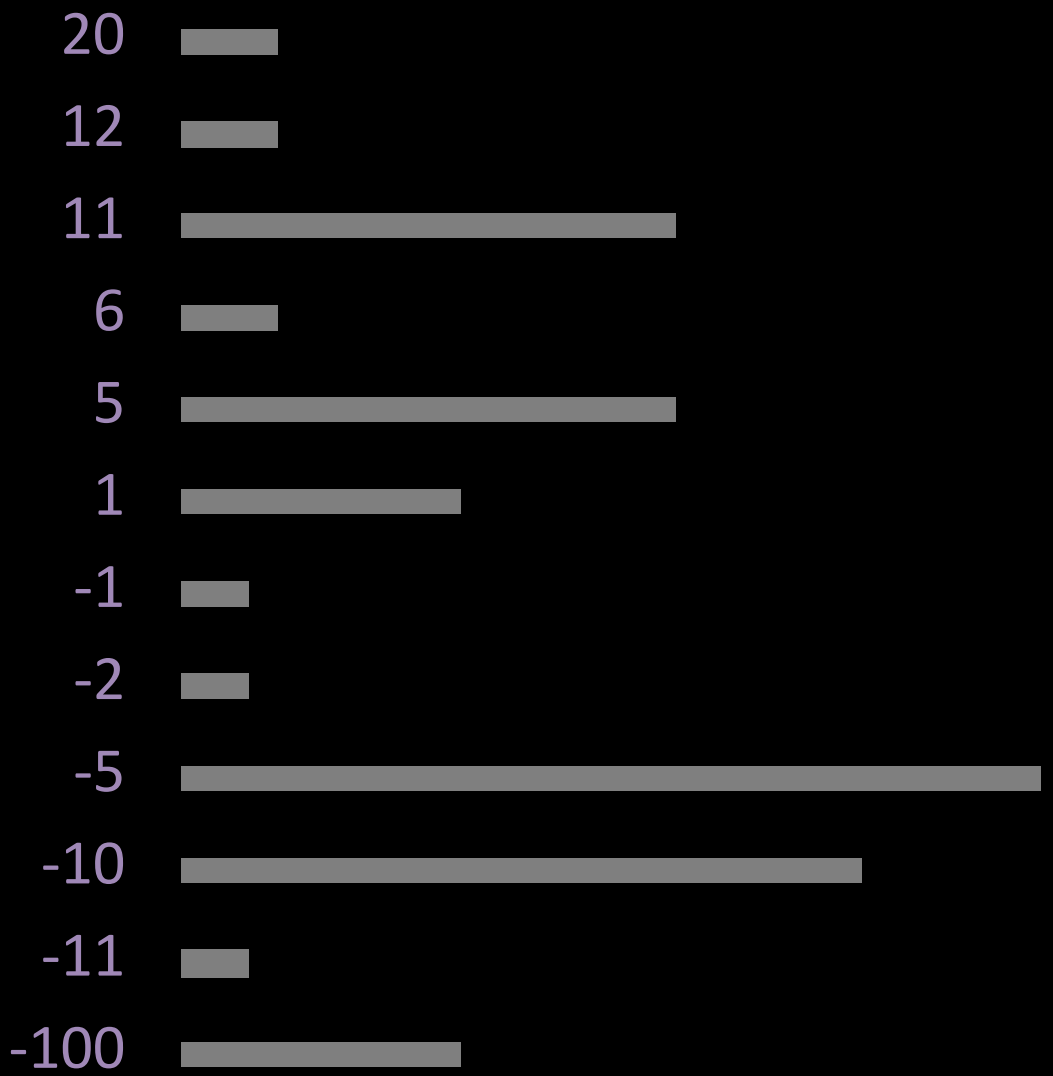
warnings

```elixir
defmodule ReadabilityExample do
  @githubBaseURL "https://github.com/"
  @github_sample_url "https://github.com/rrrene/credo/blob/master/lib/credo/check/consistency/line_endings/unix.ex"
end
```

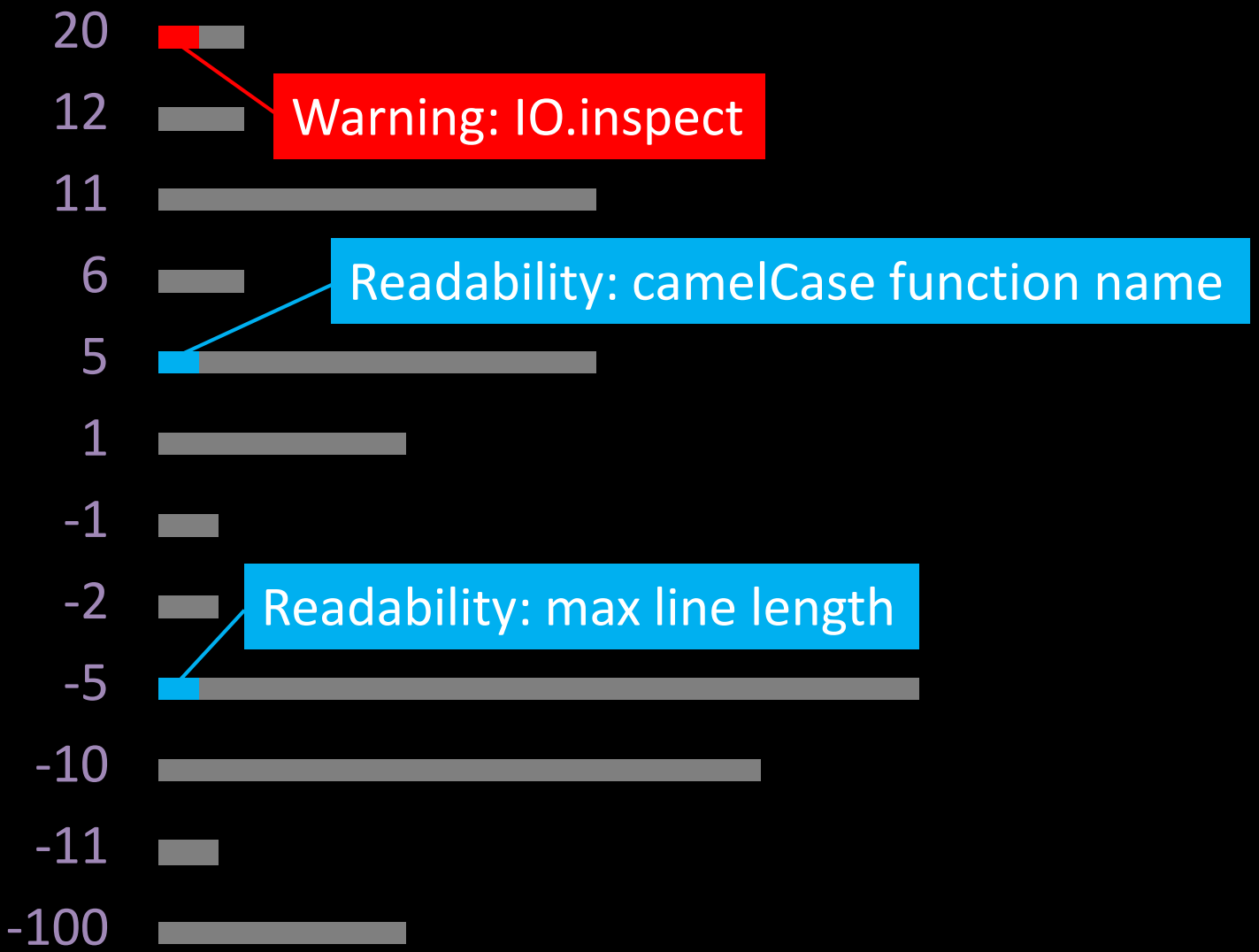

breaks convention of underscored names
1 defmodule ReadabilityExample do
2 @githubBaseURL "https://github.com/"
3 @github_sample_url "https://github.com/rrrene/credo/blob/master/lib/credo/check/consistency/line_endings/unix.ex"
4 end
exceeds maximum characters per line

Next, issues need to be prioritized.

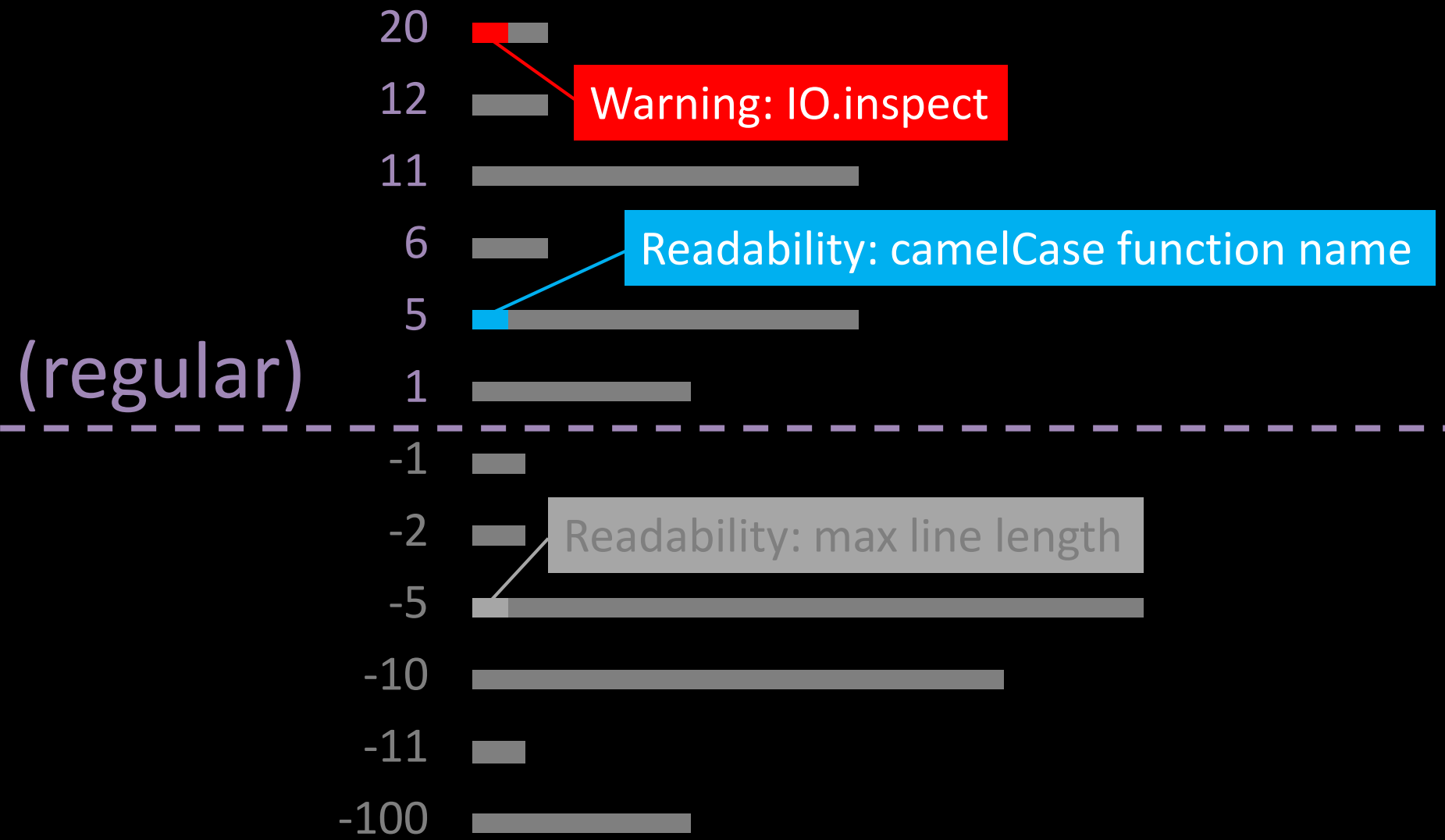
issues ordered by priority



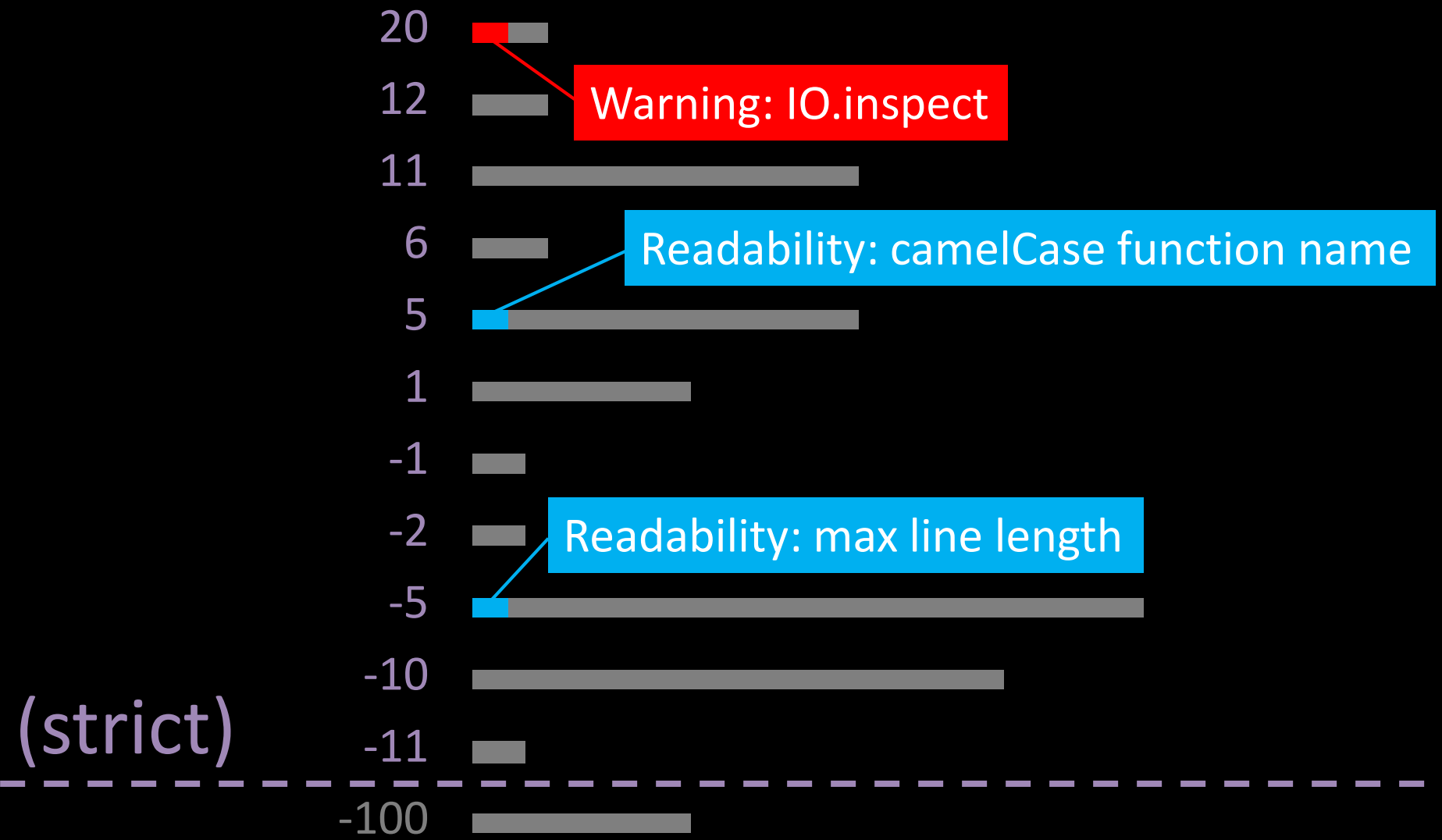
issues ordered by priority



issues ordered by priority



issues ordered by priority

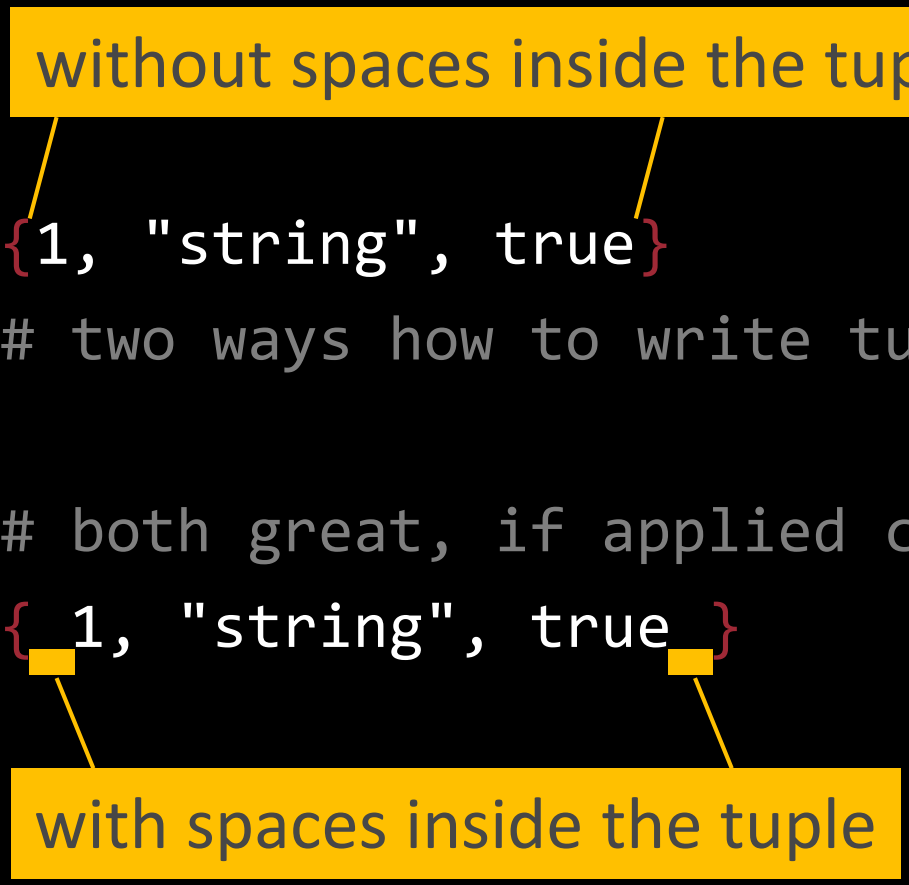


readability
software design
refactoring opportunities
consistency checks
warnings

```
{1, "string", true}
# two ways how to write tuples

# both great, if applied consistently
{ 1, "string", true }
```

without spaces inside the tuple

```
{1, "string", true}
# two ways how to write tuples

# both great, if applied consistently
{ 1, "string", true }
```


without spaces inside the tuple
{1, "string", true}
two ways how to write tuples
both great, if applied consistently
{ 1, "string", true }
with spaces inside the tuple

```elixir
defmodule BadHTTPHeaderError do
  defexception [:message]
end

defmodule UserRequestError do
  defexception [:message]
end
```

```elixir
defmodule BadHTTPHeaderError do
  defexception [:message]
end

defmodule UserRequestError do
  defexception [:message]
end
```

```elixir
defmodule BadHTTPHeaderException do
  defexception [:message]
end

defmodule UserRequestException do
  defexception [:message]
end
```

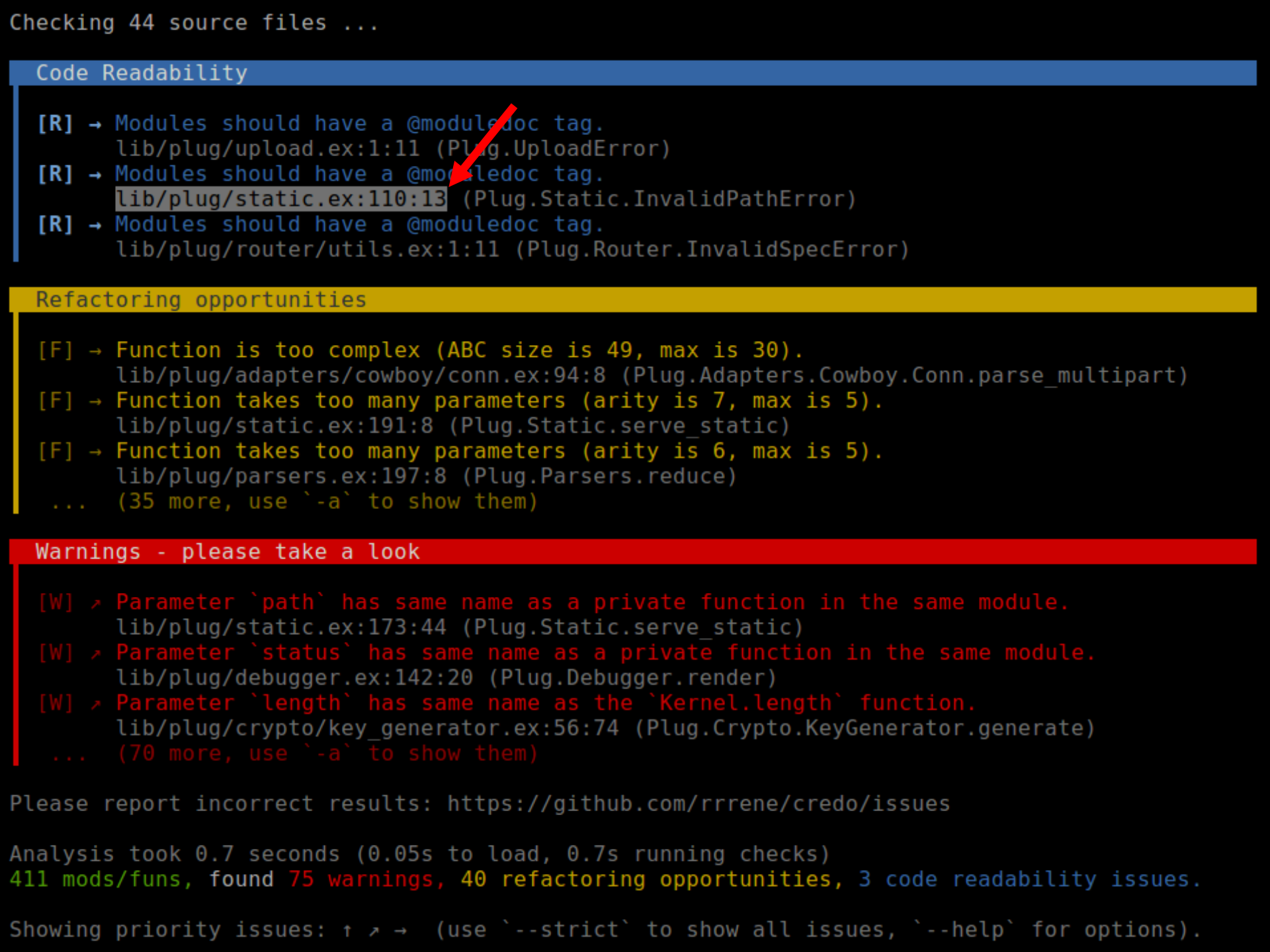
consistent suffix „Exception"

```elixir
defmodule InvalidHTTPHeader do
  defexception [:message]
end

defmodule InvalidUserRequest do
  defexception [:message]
end
```

```elixir
defmodule InvalidHeader do
  defexception [:message]
end

defmodule UserRequestFailed do
  defexception [:message]
end
```

```
$ # add credo to mix.exs
$ mix deps.get
$ mix credo
```

```
Checking 44 source files ...

  Code Readability

  [R] → Modules should have a @moduledoc tag.
        lib/plug/upload.ex:1:11 (Plug.UploadError)
  [R] → Modules should have a @moduledoc tag.
        lib/plug/static.ex:110:13 (Plug.Static.InvalidPathError)
  [R] → Modules should have a @moduledoc tag.
        lib/plug/router/utils.ex:1:11 (Plug.Router.InvalidSpecError)

  Refactoring opportunities

  [F] → Function is too complex (ABC size is 49, max is 30).
        lib/plug/adapters/cowboy/conn.ex:94:8 (Plug.Adapters.Cowboy.Conn.parse_multipart)
  [F] → Function takes too many parameters (arity is 7, max is 5).
        lib/plug/static.ex:191:8 (Plug.Static.serve_static)
  [F] → Function takes too many parameters (arity is 6, max is 5).
        lib/plug/parsers.ex:197:8 (Plug.Parsers.reduce)
    ...  (35 more, use `-a` to show them)

  Warnings - please take a look

  [W] ↗ Parameter `path` has same name as a private function in the same module.
        lib/plug/static.ex:173:44 (Plug.Static.serve_static)
  [W] ↗ Parameter `status` has same name as a private function in the same module.
        lib/plug/debugger.ex:142:20 (Plug.Debugger.render)
  [W] ↗ Parameter `length` has same name as the `Kernel.length` function.
        lib/plug/crypto/key_generator.ex:56:74 (Plug.Crypto.KeyGenerator.generate)
    ...  (70 more, use `-a` to show them)

Please report incorrect results: https://github.com/rrrene/credo/issues

Analysis took 0.7 seconds (0.05s to load, 0.7s running checks)
411 mods/funs, found 75 warnings, 40 refactoring opportunities, 3 code readability issues.

Showing priority issues: ↑ ↗ →  (use `--strict` to show all issues, `--help` for options).
```


Checking 44 source files ...
Code Readability
[R] → Modules should have a @moduledoc tag.
lib/plug/upload.ex:1:11 (Plug.UploadError)
[R] → Modules should have a @moduledoc tag.
lib/plug/static.ex:110:13 (Plug.Static.InvalidPathError)
[R] → Modules should have a @moduledoc tag.
lib/plug/router/utils.ex:1:11 (Plug.Router.InvalidSpecError)
Refactoring opportunities
[F] → Function is too complex (ABC size is 49, max is 30).
lib/plug/adapters/cowboy/conn.ex:94:8 (Plug.Adapters.Cowboy.Conn.parse_multipart)
[F] → Function takes too many parameters (arity is 7, max is 5).
lib/plug/static.ex:191:8 (Plug.Static.serve_static)
[F] → Function takes too many parameters (arity is 6, max is 5).
lib/plug/parsers.ex:197:8 (Plug.Parsers.reduce)
... (35 more, use `-a` to show them)
Warnings - please take a look
[W] ↗ Parameter `path` has same name as a private function in the same module.
lib/plug/static.ex:173:44 (Plug.Static.serve_static)
[W] ↗ Parameter `status` has same name as a private function in the same module.
lib/plug/debugger.ex:142:20 (Plug.Debugger.render)
[W] ↗ Parameter `length` has same name as the `Kernel.length` function.
lib/plug/crypto/key_generator.ex:56:74 (Plug.Crypto.KeyGenerator.generate)
... (70 more, use `-a` to show them)
Please report incorrect results: https://github.com/rrrene/credo/issues
Analysis took 0.7 seconds (0.05s to load, 0.7s running checks)
411 mods/funs, found 75 warnings, 40 refactoring opportunities, 3 code readability issues.
Showing priority issues: ↑ ↗ → (use `--strict` to show all issues, `--help` for options).

```
$ mix credo lib/plug/static.ex:90:13
```

```
Plug.Static.InvalidPathError

  [R] Category: readability
   →  Priority: normal

      Modules should have a @moduledoc tag.
      lib/plug/static.ex:110:13 (Plug.Static.InvalidPathError)

    __ CODE IN QUESTION

      defmodule InvalidPathError do
                ^^^^^^^^^^^^^^^^

    __ WHY IT MATTERS

      Every module should contain comprehensive documentation.

      Many times a sentence or two in plain english, explaining why the module
      exists, will suffice. Documenting your train of thought this way will help
      both your co-workers and your future-self.

      Other times you will want to elaborate even further and show some
      examples of how the module's functions can and should be used.

      In some cases however, you might not want to document things about a module,
      e.g. it is part of a private API inside your project. Since Elixir prefers
      explicitness over implicit behaviour, you should "tag" these modules with

          @moduledoc false

      to make it clear that there is no intention in documenting it.
```

Where do we
go from here?

Explanations

Explanations
Style Guide

Explanations

Style Guide

Editor Support

Explanations

Style Guide

Editor Support

Custom Configuration

Explanations
Style Guide
Editor Support
Custom Configuration
@lint attributes

Explanations

Style Guide

Editor Support

Custom Configuration

@lint attributes

Custom Checks are coming!

Explanations

Style Guide

Editor Support

Custom Configuration

@lint attributes

Custom Checks are coming!

Creating a Toolbox, not a Linter!

final thoughts

Questions & Alchemists

APPENDIX

Credo – A static code analysis tool with a focus on code consistency and teaching.
https://hex.pm/packages/credo

Dogma – A code style linter for Elixir, powered by shame.
https://hex.pm/packages/dogma

Clark Kampfe in “Elixir is not Ruby”
https://zeroclarkthirty.com/2015-11-01-elixir-is-not-ruby.html

Inch CI – Lint your docs
http://inch-ci.org

ElixirStatus – community announcements
http://elixirstatus.com

HexFaktor – monitor your deps
http://beta.hexfaktor.org