

# CREDO: YOUR TEAM'S ROBOT MENTOR



# **DEEP-DIVE INTO**

- ANALYSIS**

- CHECKS**

- PLUGINS**

rrrene.org

Hi, I'm René!  
(@rrrene on  
social media)

rrrene\*

About



**Dr. René Föhring**

Head of Product Development

[@5MindsIT](#), Open Source Maintainer  
and Conference Speaker.

**LINTERS,  
RIGHT?**



**rrrene.org**

**rrrene\***

[About](#)

## **Linters: Why your team should tread lightly**

11 Mar 2019

The usage of linters, tools to analyse your code and tell you where it can be improved upon, is becoming more widespread in recent years. Many programmers nowadays start their careers in places where a CI system runs [ESLint](#), [Rubocop](#) or [Credo](#) against all their checkins. Overall, this is a good thing, because it enables discussing code quality issues within teams more openly and in a transparent way.

(you can imagine there's a "but")



rrrene.org

rrrene\*

[About](#)

## Linters: Why your team should tread lightly

11 Mar 2019

The usage of linters, tools to analyse your code and tell you where it can be improved upon, is becoming more widespread in recent years. Many programmers nowadays start their careers in places where a CI system runs [ESLint](#), [Rubocop](#) or [Credo](#) against all their checkins. Overall, this is a good thing, because it enables discussing code quality issues within teams more openly and in a transparent way.

(you can imagine there's a "but")

**Blogging:**  
It's like screaming  
into a pillow.

A silhouette of a person stands on a dark, grassy hill, looking up at a vast night sky. The Milky Way galaxy is visible as a bright, colorful band of stars stretching across the sky, with a purple and pink hue in the center. The sky is filled with numerous stars, and the overall scene conveys a sense of wonder and exploration.

# WHAT ARE WE LOOKING FOR?

**MASTERING  
COMPLEXITIES  
MANAGING  
EXPECTATIONS**



# **SENSE OF IMPROVEMENT PREVENTING ERRORS**

# SENSE OF **IMPROVEMENT PREVENTING** ERRORS

```
$ mix dogma
```

```
Inspecting 78 files.
```

```
.X...XXX.XX.XX.XXXXXXXXXXXXXXXXXX.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
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).
```



**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





twitter.com



**SmartLogic**  
@smartlogic



"I wanted Credo to be kind of a mentor, a digital mentor, a robot mentor for people because I was fortunate enough to have a human mentor which is a privilege that not everybody has in life and I think that really got me started." — @rrrene  
[smr.tl/2KfWtvV](https://smr.tl/2KfWtvV)

**René Föhrling**  
**on Credo**



twitter.com

# Mentoring: Helping others grow.



SmartLogic  
@smartlogic

"I wanted Credo to be kind of a mentor, a digital mentor, a robot mentor for people because I was fortunate enough to have a human mentor which is a privilege that not everybody has in life and I think that really got me started." — @rrrene  
[smr.tl/2KfWtvV](https://smr.tl/2KfWtvV)

René Föhrling  
on Credo



5Minds  
IT-SOLUTIONS

**CREDO WANTS TO BE YOUR  
TEAM'S ROBOT MENTOR.**

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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



```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `--a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `--a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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



```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `--a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

```
$ mix credo
```

## Code Readability

```
[R] ↗ Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

```
... (85 more, use `-a` to show them)
```

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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



```
$ mix credo lib/mix/tasks/my_task.ex:2:3
```

## Code Readability

[R] ↗ Large numbers should be written with underscores: 314\_159\_265

lib/mix/tasks/my\_task.ex:2:3 (Mix.Tasks.MyTask)

[R] → Modules should have a @moduledoc tag.

lib/mix/tasks/my\_task.ex:1:11 (Mix.Tasks.MyTask)

... (85 more, use ``-a`` to show them)

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

Analysis took 0.03 seconds (0.00s to load, 0.03s running checks)

21 mods/funs, found 2 code readability issues.

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

[R] Category: readability

➤ Priority: high

Large numbers should be written with underscores: 314\_159\_265

lib/mix/tasks/my\_task.ex:2:3 (Mix.Tasks.MyTask)

\_\_ CODE IN QUESTION

```
1 defmodule Mix.Tasks.MyTask do
2   314159265
   ^^^^^^^^^
3 end
4
```

\_\_ WHY IT MATTERS

## \_\_ WHY IT MATTERS

Numbers can contain underscores for readability purposes. These do not affect the value of the number, but can help read large numbers more easily.

```
141592654 # how large is this number?
```

```
141_592_654 # ah, it's in the hundreds of millions!
```

Like all `Readability` issues, this one is not a technical concern. But you can improve the odds of others reading and liking your code by making it easier to follow.

## \_\_ CONFIGURATION OPTIONS

## \_\_ CONFIGURATION OPTIONS

To configure this check, use this tuple

```
{Credo.Check.Readability.LargeNumbers, <params>}
```

with <params> being false or any combination of these keywords:

<code>only_greater_than:</code>	The check only reports numbers greater than this. (defaults to 9999)
---------------------------------	---

# ANALYSIS



# ABSTRACT SYNTAX TREES (AST)

# AST consists mostly of 3 element tuples  
{\_operation, \_meta, \_arguments}



$$1 + 2$$

$1 + 2$ 

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

1 + 2

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

$1 + 2$ 

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

```
some_var  
|> inspect(label: "foo")
```

```
some_var  
|> inspect(label: "foo")
```

```
{:|>, [line: 1],  
 [  
   {:some_var, [line: 1], nil},  
   {:inspect, [line: 1], [[label: "foo"]]}  
 ]  
}
```

some\_var

|> inspect(label: "foo")

```
{:|>, [line: 1],  
 [  
   {:some_var, [line: 1], nil},  
   {:inspect, [line: 1], [[label: "foo"]]}  
 ]  
}
```



```
some_var
```

```
|> inspect(label: "foo")
```

```
{:|>, [line: 1],  
  [  
    {:some_var, [line: 1], nil},  
    {:inspect, [line: 1], [[label: "foo"]]}  
  ]  
}
```

```
some_var
```

```
|> inspect(label: "foo")
```

```
{:|>, [line: 1],  
 [  
   {:some_var, [line: 1], nil},  
   {:inspect, [line: 1], [[label: "foo"]]}  
 ]  
}
```

# Elixir literals that, when quoted, return themselves

<code>:sum</code>	# => Atoms
<code>1.0</code>	# => Numbers
<code>[1, 2]</code>	# => Lists
<code>"strings"</code>	# => Strings
<code>{key, value}</code>	# => Tuples with two elements

# more info: <https://elixir-lang.org/getting-started/meta/quote-and-unquote.html>

```
defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end
```

```
$ iex -S mix run -e "MyProject.add(1, 2)"
```

```
Erlang/OTP 22 [erts-10.4.4] [source] [64-bit]
```

```
Request to pry #PID<0.103.0> at MyProject.add/
```

```
defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end
```

```
7:      c = a + b
8:      require IEx
9:      IEx.pry()
10:    end
11:  end
```

```
Allow? [Yn] Y
```

```
Interactive Elixir (1.9.1) - press Ctrl+C to exit (type h() ENTER for help)
```

```
pry(1)>
```

```
$ iex -S mix run -e "MyProject.add(1, 2)"
```

```
Erlang/OTP 22 [erts-10.4.4] [source] [64-bit]
```

```
# snip ...
```

```
Interactive Elixir (1.9.1) - press Ctrl+C to exit (type h() ENTER for help)
```

```
pry(1)> Enum.map([a, b, c], &IO.inspect/1)
```

```
1
```

```
2
```

```
3
```

```
defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end
```

```

{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2]
          [
            do: {:__block__, [],
              [
                {::=, [line: 3],
                  [
                    {:c, [line: 3], nil},
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                  ]},
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]}]},
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                 [line: 4], []}
              ]}
            ]}
          ]}
        ]}
      ]}
    ]}
  ]}
]

```

```

defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end

```



```

{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2]
          [
            do: {:__block__, [],
              [
                {::=, [line: 3],
                  [
                    {:c, [line: 3], nil},
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
              ]},
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]}]},
                [{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]}

```

```

defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end

```

```
{:defmodule, [line: 1],
[
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2]
```

```
defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx; IEx.pry()
  end
end
```

```
{
  {:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
  [line: 4],
  []
}
```

```
{{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]}.
```

```
[line: 4], []}
```

```
]]
```

```
]
```

```
]]
```

```
]
```

```
]]
```

```
{:defmodule, [line: 1],
[
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
            {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
            [line: 4], []}
          ]}
        ]}
      ]}
  ]}
]}
```

```
Macro.prewalk(ast, fn leaf ->
  IO.inspect(leaf)
end)
```

# AST

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]},  
                  ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]},  
            ]},  
          ]},  
        ]},  
      ]},  
    ]},  
  ]}
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}}  
                ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [[:Adder]]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {:=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
             {:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
             [line: 4], []}
          ]}
        ]}
      ]}
    ]}
  ]}
]}
```

# AST

```
{:defmodule, [line: 1],  
 [  
   {:__aliases__, [line: 1], [:Adder]},  
   [  
     do: {:def, [line: 2],  
      [  
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
        [  
          do: {:__block__, [],  
            [  
              {:=, [line: 3],  
                [  
                  {:c, [line: 3], nil},  
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}  
                ]},  
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
               [line: 4], []}  
            ]}  
          ]}  
        ]}  
      ]}  
     ]  
   ]  
 ]  
}]
```

# AST

```
{:defmodule, [line: 1],
[
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                  [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]
```



```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
             {::., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
             [line: 4], []}
          ]}
        ]}
      ]}
    ]}
  ]}
]}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
              [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]}
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {::=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}  
                  ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                [{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

# AST

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                [[::=, [line: 3],  
                  [  
                    [:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}  
                  ]}],  
                [:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]]},  
                [{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]}]  
            ]  
          ]  
        ]  
      }  
    ]  
  ]  
}
```

# AST

```
{:defmodule, [line: 1],  
 [  
   {:__aliases__, [line: 1], [:Adder]},  
   [  
     do: {:def, [line: 2],  
      [  
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
        [  
          do: {:__block__, [],  
            [  
              {:=, [line: 3],  
                [  
                  {:c, [line: 3], nil},  
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}}  
                ]}},  
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
              [line: 4], []}  
            ]}}  
          ]  
        ]}  
      ]}  
    ]  
  ]  
}
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {::=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {::+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}}  
                ]},  
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
              [line: 4], []}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]  
}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {:=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
             {::., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
             [line: 4], []}
          ]}
        ]}
      ]}
  ]}
]}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                 [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]
```



```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                 [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {::=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}  
                  ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                [{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]},  
                  ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]}]},
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
               [line: 4], []}
            ]}
          ]}
        ]}
      ]}
    ]}
  ]}
]
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}}  
                ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]}]},  
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],
[
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
              [{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
                [line: 4], []}
            ]}
          ]}
        ]}
      ]}
  ]}
]
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]},  
                  ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                {[:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                [line: 4], []}  
              ]}  
            ]}  
          ]}  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
      [
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
        [
          do: {:__block__, [],
            [
              {::=, [line: 3],
                [
                  {:c, [line: 3], nil},
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}
                ]},
            {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},
            {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},
            [line: 4], []}
          ]}
        ]}
      ]}
  ]}
]}
```



```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            do: {:__block__, [],  
              [  
                {:=, [line: 3],  
                  [  
                    {:c, [line: 3], nil},  
                    {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}}  
                ]},  
                {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
                {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
                 [line: 4], []}  
              ]}  
            ]}  
          ]  
        ]}  
      ]  
    ]  
  ]}
```

```
{:defmodule, [line: 1],  
 [  
   {:__aliases__, [line: 1], [:Adder]},  
   [  
     do: {:def, [line: 2],  
      [  
        {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
        [  
          do: {:__block__, [],  
            [  
              {:=, [line: 3],  
                [  
                  {:c, [line: 3], nil},  
                  {:+, [line: 3], [{:a, [line: 3], nil}, {:b, [line: 3], nil}]}  
                ]},  
              {:require, [line: 4], [{:__aliases__, [line: 4], [:IEx]]},  
              {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
               [line: 4], []}  
            ]}  
          ]}  
        ]}  
      ]}  
    ]  
  ]  
}
```

```
{:defmodule, [line: 1],  
  [  
    {:__aliases__, [line: 1], [:Adder]},  
    [  
      do: {:def, [line: 2],  
        [  
          {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},  
          [  
            {  
              {:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]},  
              [line: 4],  
              []  
            }  
          ]  
        }  
      ]  
    ]  
  ]  
}
```

```
Macro.prewalk(ast, fn leaf ->  
  IO.inspect(leaf)  
end)
```

`{{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]}, :pry]}.`

`[line: 4], []`

```
{:defmodule, [line: 1],
 [
  {:__aliases__, [line: 1], [:Adder]},
  [
    do: {:def, [line: 2],
```

```
Macro.prewalk(ast, fn
  {:., _, [{:__aliases__, _, [:IEx]], :pry}} ->
    # do something
  leaf -> leaf
end)
```

```
    [
      {:add, [line: 2], [{:a, [line: 2], nil}, {:b, [line: 2], nil}]},
      [
        {
          {:., [line: 4], [{:__aliases__, [line: 4], [:IEx]], :pry}},
          [line: 4],
          []
        }
      ]
    ]
  }
}
```

```
    {{:., [line: 4], [{:__aliases__, [line: 4], [:IEx]], :pry}}.
```

```
    [line: 4], []
```

```
  ]}
}
```

```
]
```

```
]}
]
```

```
]
```

```
]}
}
```

# CHECKS



```
defmodule Credo.Check.Warning.IExPry do
  @explanation [check: "..."]

  use Credo.Check, base_priority: :high

  def run(source_file, params \\ []) do
    Credo.Code.prewalk(source_file,
                        &traverse(&1, &2, source_file, params))
  end
end
```

```
defmodule Credo.Check.Warning.IExPry do
  @explanation [check: "..."]

  use Credo.Check, base_priority: :high

  def run(source_file, params \\ []) do
    Credo.Code.prewalk(source_file,
                        &traverse(&1, &2, source_file, params))
  end
end
```

```
defmodule Credo.Check.Warning.IExPry do
  @explanation [check: "..."]

  use Credo.Check, base_priority: :high

  def run(source_file, params \\ []) do
    Credo.Code.prewalk(source_file,
                        &traverse(&1, &2, source_file, params))
  end
end
```



```
defmodule Credo.Check.Warning.IExPry do
  @explanation [check: "..."]

  use Credo.Check, base_priority: :high

  def run(source_file, params \\ []) do
    Credo.Code.prewalk(source_file,
                        &traverse(&1, &2, source_file, params))
  end
end
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp traverse(  
  {  
    {:., _, [{:__aliases__, _, [:IEx]}, :pry]},  
    meta,  
    _arguments  
  } = ast,  
  issues,  
  source_file,  
  params  
) do  
  {ast, [issue_for(meta[:line], source_file, params) | issues]}  
end  
  
defp traverse(ast, issues, _source_file, _params), do: {ast, issues}
```

```
defp issue_for(line_no, source_file, params) do
  issue_meta = IssueMeta.for(source_file, params)

  format_issue(
    issue_meta,
    message: "There should be no calls to IEx.pry/0.",
    trigger: "IEx.pry",
    line_no: line_no
  )
end
```



```
defp issue_for(line_no, source_file, params) do
  issue_meta = IssueMeta.for(source_file, params)

  format_issue(
    issue_meta,
    message: "There should be no calls to IEx.pry/0.",
    trigger: "IEx.pry",
    line_no: line_no
  )
end
```

```
defp issue_for(line_no, source_file, params) do
  issue_meta = IssueMeta.for(source_file, params)

  format_issue(
    issue_meta,
    message: "There should be no calls to IEx.pry/0.",
    trigger: "IEx.pry",
    line_no: line_no
  )
end
```

```
defp issue_for(line_no, source_file, params) do
  issue_meta = IssueMeta.for(source_file, params)

  format_issue(
    issue_meta,
    message: "There should be no calls to IEx.pry/0.",
    trigger: "IEx.pry",
    line_no: line_no
  )
end
```

```
$ mix credo
```

### Code Readability

```
[R] → Modules should have a @moduledoc tag.  
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

### Warnings - please take a look

```
[W] ↗ There should be no calls to IEx.pry/0.  
lib/my_project.ex:9:5 (MyProject.add)
```

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

Analysis took 0.03 seconds (0.02s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues, 1 warning.

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

```
defmodule MyProject do
  def add(a, b) do
    c = a + b
    require IEx
    # credo:disable-for-next-line Credo.Check.Warning.IExPry
    IEx.pry()
  end
end
```

# CUSTOM CHECK

[https://github.com/rrrene/credo\\_sample\\_project](https://github.com/rrrene/credo_sample_project)

```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```

```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```



```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```

```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```

```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```

```
defmodule MyProject.Checks.MyIExPry do
  # ...

  @default_params [excluded: []]

  def run(source_file, params \\ []) do
    excluded = Params.get(params, :excluded, @default_params)
    excluded_for_source_file? = Enum.any?(excluded, &(source_file.filename =~ &1))

    if excluded_for_source_file? do
      []
    else
      Credo.Code.prewalk(source_file, &traverse(&1, &2, source_file, params))
    end
  end

  # ...
end
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      requires: [".../lib/my_project/checks/my_iex_pry.ex"],  
      checks: [  
        {Credo.Check.Warning.IExPry, false},  
        {MyProject.Checks.MyIExPry, []}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      requires: ["/lib/my_project/checks/my_iex_pry.ex"],  
      checks: [  
        {Credo.Check.Warning.IExPry, false},  
        {MyProject.Checks.MyIExPry, []}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      requires: ["/lib/my_project/checks/my_iex_pry.ex"],  
      checks: [  
        {Credo.Check.Warning.IExPry, false},  
        {MyProject.Checks.MyIExPry, []}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      requires: [".../lib/my_project/checks/my_iex_pry.ex"],  
      checks: [  
        {Credo.Check.Warning.IExPry, false},  
        {MyProject.Checks.MyIExPry, []}  
      ]  
    }  
  ]  
}
```



```
$ mix credo
```

## Code Readability

```
[R] → Modules should have a @moduledoc tag.  
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

## Warnings - please take a look

```
[W] ↗ There should be no calls to IEx.pry/0.  
lib/my_project.ex:9:5 (MyProject.add)
```

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

Analysis took 0.03 seconds (0.02s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues, 1 warning.

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

```
%{  
  configs: [  
    %{  
      name: "default",  
      requires: [".lib/my_project/checks/my_iex_pry.ex"],  
      checks: [  
        {Credo.Check.Warning.IExPry, false},  
        {MyProject.Checks.MyIExPry, [excluded: ["lib/my_project.ex"]]}  
      ]  
    }  
  ]  
}
```

```
$ mix credo
```

## Code Readability

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

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

Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues.

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

# PLUGINS



# **CONTROLLING THE EXECUTION PIPELINE**

```
# Ecto.Changeset
```

```
changeset= cast(%User{}, params, [:name, :email, :age])
```

```
changeset
```

```
|> validate_required([:name, :email])
```

```
|> validate_format(:email, ~r/@/)
```

```
|> validate_inclusion(:age, 18..100)
```

```
|> unique_constraint(:email)
```

```
# Plug.Conn
```

```
conn
```

```
|> put_resp_content_type("text/plain")
```

```
|> put_resp_header("Server-Timing", "total;dur=#{timing_in_ms}")
```

```
|> send_resp(200, body)
```

```
# Credo.Execution
```

```
exec
```

```
|> parse_cli_options()
```

```
|> initialize_plugins()
```

```
|> validate_cli_options()
```

```
|> convert_cli_options_to_config()
```

```
|> determine_command()
```

```
|> set_default_command()
```

```
|> resolve_config()
```

```
|> validate_config()
```

```
|> run_command()
```

```
|> halt_execution()
```

```
# Credo.Execution
```

```
exec
```

```
|> parse_cli_options()  
|> initialize_plugins()  
|> validate_cli_options()  
|> convert_cli_options_to_config()  
|> determine_command()  
|> set_default_command()  
|> resolve_config()  
|> validate_config()  
|> run_command()  
|> halt_execution()
```

```
exec # e.g. SuggestCommand
```

```
|> load_and_validate_source_files()  
|> prepare_analysis()  
|> print_before_analysis()  
|> run_analysis()  
|> filter_issues()  
|> print_after_analysis()
```



```
%Credo.Execution{  
  argv: [...],  
  cli_options: %Credo.CLI.Options{...},  
  checks: [...],  
  issues: [...],  
  plugins: [...],  
  commands: %{...},  
  config_files: [...],  
  halted: false,  
  pipeline_map: %{...},  
  ...  
}
```

```
%Credo.Execution{
  argv: [...],
  cli_options: %Credo.CLI.Options{...},
  checks: [...],
  issues: [...],
  plugins: [...],
  commands: %{...},
  config_files: [...],
  halted: false,
  pipeline_map: %{...},
  ...
}
```

```
%Credo.Execution{
  argv: [...],
  cli_options: %Credo.CLI.Options{...},
  checks: [...],
  issues: [...],
  plugins: [...],
  commands: %{...},
  config_files: [...],
  halted: false,
  pipeline_map: %{...},
  ...
}
```

```
%Credo.Execution{  
  pipeline_map: %{Credo.Execution => [  
    parse_cli_options: [...],  
    initialize_plugins: [...],  
    validate_cli_options: [...],  
    convert_cli_options_to_config: [...],  
    determine_command: [...],  
    set_default_command: [...],  
    resolve_config: [...],  
    validate_config: [...],  
    run_command: [...],  
    halt_execution: [...]  
  ], Credo.CLI.Command.Suggest.SuggestCommand => [...]}},  
  ...  
}
```

```
%Credo.Execution{  
  pipeline_map: %{Credo.Execution => [  
    parse_cli_options: [...],  
    initialize_plugins: [...],  
    validate_cli_options: [...],  
    convert_cli_options_to_config: [...],  
    determine_command: [...],  
    set_default_command: [...],  
    resolve_config: [...],  
    validate_config: [...],  
    run_command: [...],  
    halt_execution: [...]  
  ], Credo.CLI.Command.Suggest.SuggestCommand => [...]}},  
  ...  
}
```

```
%Credo.Execution{  
  pipeline_map: %{Credo.Execution => [  
    parse_cli_options: [...],  
    initialize_plugins: [...],  
    validate_cli_options: [...],  
    convert_cli_options_to_config: [...],  
    determine_command: [...],  
    set_default_command: [...],  
    resolve_config: [...],      # named groups of tasks  
    validate_config: [...],    # a bit like Plug pipelines  
    run_command: [...],  
    halt_execution: [...]  
  ], Credo.CLI.Command.Suggest.SuggestCommand => [...]}  
  ...  
}
```

```
%Credo.Execution{  
  pipeline_map: %{Credo.Execution => [  
    ...  
    resolve_config: [  
  
      {Credo.Execution.Task.UseColors, []},  
      {Credo.Execution.Task.RequireRequires, []}  
  
    ],  
    ...  
  ]},  
  ...  
}
```

```
%Credo.Execution{  
  pipeline_map: %{Credo.Execution => [  
    ...  
    resolve_config: [  
      {MyPrependedTask, []},  
      {Credo.Execution.Task.UseColors, []},  
      {Credo.Execution.Task.RequireRequires, []}  
      {MyAppendedTask, []}  
    ],  
    ...  
  ]},  
  ...  
}
```



# MINIMAL PLUGIN

[https://github.com/rrrene/credo\\_demo\\_plugin](https://github.com/rrrene/credo_demo_plugin)

CREDO  
SOURCE CODE

PLUGIN  
SOURCE CODE

USER PROJECT  
SOURCE CODE

# MINIMAL PLUGIN

[https://github.com/rrrene/credo\\_demo\\_plugin](https://github.com/rrrene/credo_demo_plugin)

```
# mix.exs
{:credo_demo_plugin, "~> 0.1.0"},

# .credo.exs
%{
  configs: [
    %{
      name: "default",
      plugins: [
        {CredoDemoPlugin, []}
      ]
    }
  ]
}
```

```
defmodule CredoDemoPlugin do
  def init(exec) do
    # This would be an MVP: minimal viable plugin ^_^
    exec
  end
end
```

```

defmodule CredoDemoPlugin do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> register_command("demo",
                        CredoDemoPlugin.DemoCommand)
    |> register_cli_switch(:demo_name, :string, :D)
    |> prepend_task(:set_default_command,
                    CredoDemoPlugin.SetDemoAsDefaultCommand)
  end
end

```

```

defmodule CredoDemoPlugin do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> register_command("demo",
                        CredoDemoPlugin.DemoCommand)
    |> register_cli_switch(:demo_name, :string, :D)
    |> prepend_task(:set_default_command,
                    CredoDemoPlugin.SetDemoAsDefaultCommand)
  end
end
    
```

```

defmodule CredoDemoPlugin do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> register_command("demo",
                        CredoDemoPlugin.DemoCommand)
    |> register_cli_switch(:demo_name, :string, :D)
    |> prepend_task(:set_default_command,
                    CredoDemoPlugin.SetDemoAsDefaultCommand)
  end
end

```

```

defmodule CredoDemoPlugin do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> register_command("demo",
                        CredoDemoPlugin.DemoCommand)
    |> register_cli_switch(:demo_name, :string, :D)
    |> prepend_task(:set_default_command,
                    CredoDemoPlugin.SetDemoAsDefaultCommand)
  end
end

```



```

defmodule CredoDemoPlugin do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> register_command("demo",
                        CredoDemoPlugin.DemoCommand)
    |> register_cli_switch(:demo_name, :string, :D)
    |> prepend_task(:set_default_command,
                    CredoDemoPlugin.SetDemoAsDefaultCommand)
  end
end

```

**LET'S WRITE  
AN ACTUAL PLUGIN**

```
$ mix credo
```

## Code Readability

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

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

Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues.

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

```
%{  
  configs: [  
    %{  
      name: "default",  
      checks: [  
        {Credo.Check.Readability.ModuleDoc,  
         [excluded: [~r"_task\.ex$"]]}  
      ]  
    }  
  ]  
}
```

# **FILTER CHECKS PLUGIN**

[https://github.com/rrrene/credo\\_check\\_filter\\_config](https://github.com/rrrene/credo_check_filter_config)

```

defmodule CredoCheckFilterConfig do
  import Credo.Plugin

  def init(exec) do
    exec
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :determine_issues,
      CredoCheckFilterConfig.Task.FilterChecks
    )
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :print_issues,
      CredoCheckFilterConfig.Task.ReportFilteredChecks
    )
  end
end
end

```

```

defmodule CredoCheckFilterConfig do
  import Credo.Plugin

  def init(exec) do
    exec
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :determine_issues,
      CredoCheckFilterConfig.Task.FilterChecks
    )
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :print_issues,
      CredoCheckFilterConfig.Task.ReportFilteredChecks
    )
  end
end
end

```

```

defmodule CredoCheckFilterConfig do
  import Credo.Plugin

  def init(exec) do
    exec
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :determine_issues,
      CredoCheckFilterConfig.Task.FilterChecks
    )
    |> append_task(
      Credo.CLI.Command.Suggest.SuggestCommand,
      :print_issues,
      CredoCheckFilterConfig.Task.ReportFilteredChecks
    )
  end
end

```



```

defmodule CredoCheckFilterConfig.Task.FilterChecks do

  # ...

  def call(exec, _opts) do
    all_issues = Execution.get_issues(exec)
    checks = get_check_tuples_for_raised_issues(exec, all_issues)

    {ignored_issues, issues} = Enum.split_with(all_issues,
                                              &ignored_by_excluded_param?(&1, checks))

    exec
    |> Execution.put_assign(
      "credo_check_filter_config.ignored_issue_count",
      Enum.count(ignored_issues)
    )
    |> Execution.set_issues(issues) # overwrite issues for this execution
  end
end

```

```

defmodule CredoCheckFilterConfig.Task.ReportFilteredChecks do
  use Credo.Execution.Task

  def call(exec, _opts) do
    ignored_issue_count =
      Credo.Execution.get_assign(exec, "credo_check_filter_config.ignored_issue_count")

    case ignored_issue_count do
      0 -> nil
      1 -> Credo.CLI.Output.UI.puts([:faint, "1 issue was filtered."])
      count -> Credo.CLI.Output.UI.puts([:faint, "#{count} issues were filtered."])
    end

    exec
  end
end

```

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoCheckFilterConfig, []}  
      ],  
      checks: [  
        {Credo.Check.Readability.ModuleDoc,  
          [excluded: [~r"_task\\.ex$"]]}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoCheckFilterConfig, []}  
      ],  
      checks: [  
        {Credo.Check.Readability.ModuleDoc,  
          [excluded: [~r"_task\\.ex$"]]}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoCheckFilterConfig, []}  
      ],  
      checks: [  
        {Credo.Check.Readability.ModuleDoc,  
          [excluded: [~r"_task\\.ex$"]]}  
      ]  
    }  
  ]  
}
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoCheckFilterConfig, []}  
      ],  
      checks: [  
        {Credo.Check.Readability.ModuleDoc,  
          [excluded: [~r"_task\.ex$"]]}  
      ]  
    }  
  ]  
}
```

```
$ mix credo
```

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

```
Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)
```

```
344 mods/funs, found no issues.
```

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

```
1 issue was filtered.
```

```
$ mix credo
```

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

```
Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)
```

```
344 mods/funs, found no issues.
```

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

```
1 issue was filtered.
```



**LET'S INTEGRATE  
AN ACTUAL THIRD-PARTY  
TOOL**

# INCH IN A NUTSHELL

---

# INCH IN A NUTSHELL

---

**static analysis tool**  
for inline code documentation

**identifies documentation**  
in modules and functions

**assigns a relative priority**  
based on a pairwise comparison

**assigns a weighted score**  
based on an biased framework

# INCH IN A NUTSHELL

---

**static analysis tool**  
for inline code documentation

**identifies documentation**  
in modules and functions

**assigns a relative priority**  
based on a pairwise comparison

**assigns a weighted score**  
based on an biased framework

**focuses the user**  
on certain aspects

**is able to explain**  
how it determines scores

**no demand for all items**  
to be documented

**does not obsess**  
about numbers

```
$ mix inch
```

```
# Proper documentation present
```

```
|
```

```
| [B] → Phoenix.Token.sign/4 (lib/phoenix/token.ex:94)
```

```
| [B] → Mix.Phoenix.inflect/1 (lib/mix/phoenix.ex:57)
```

```
| [B] ↓ Phoenix.Endpoint.Supervisor.server?/2 (lib/phoenix/endpoint/supervisor.ex:103)
```

```
# Undocumented
```

```
|
```

```
| [U] → Phoenix.Param.Any (lib/phoenix/param.ex:82)
```


```
| [U] → Phoenix.Param.Map (lib/phoenix/param.ex:75)
```

```
| [U] ↓ Mix.Phoenix.Schema.valid?/1 (lib/mix/phoenix/schema.ex:41)
```

You might want to look at these files:

```
| lib/phoenix/controller.ex
```

```
| lib/phoenix/test/conn_test.ex
```

Grade distribution (undocumented, C, B, A): 

```
$ mix inch
```

```
# Proper docs grades instead of numbers
```

```
|  
| [B] → Phoenix.Token.sign/4 (lib/phoenix/token.ex:94)  
| [B] → Mix.Phoenix.inflect/1 (lib/mix/phoenix.ex:57)  
| [B] ↓ Phoenix.Endpoint.Supervisor.server?/2 (lib/phoenix/endpoint/supervisor.ex:103)
```

```
# Undocumented
```

```
|  
| [U] → Phoenix.Param.Any (lib/phoenix/param.ex:82)  
| [U] → Phoenix.Param.Map (lib/phoenix/param.ex:75)  
| [U] ↓ Mix.Phoenix.Schema.valid?/1 (lib/mix/phoenix/schema.ex:41)
```

You might want to look at these files:

```
| lib/phoenix/controller.ex  
| lib/phoenix/test/conn_test.ex
```

Grade distribution (undocumented, C, B, A): ■ \_ ■ \_

```
$ mix inch
```

```
# Proper documentation present
```

```
|
```

```
| [B] → Phoenix.Token.sign/4 (lib/phoenix/token.ex:94)
```

```
| [B] → Mix.Phoenix.inflect/1 (lib/mix/phoenix.ex:57)
```

```
| [B] ↓ Phoenix.Endpoint.Supervisor.server?/2 (lib/phoenix/endpoint/supervisor.ex:103)
```

```
# Undocumented
```

special grade for undocumented objects



```
|
```

```
| [U] → Phoenix.Param.Any (lib/phoenix/param.ex:82)
```


```
| [U] → Phoenix.Param.Map (lib/phoenix/param.ex:75)
```

```
| [U] ↓ Mix.Phoenix.Schema.valid?/1 (lib/mix/phoenix/schema.ex:41)
```

You might want to look at these files:

```
| lib/phoenix/controller.ex
```

```
| lib/phoenix/test/conn_test.ex
```

Grade distribution (undocumented, C, B, A): 

```
$ mix inch
```

```
# Proper documentation present
```

```
|
```

```
| [B] → Phoenix.Token.sign/4 (lib/phoenix/token.ex:94)
```

```
| [B] → Mix.Phoenix.inflect/1 (lib/mix/phoenix.ex:57)
```

```
| [B] ↓ Phoenix.Endpoint.Supervisor.server?/2 (lib/phoenix/endpoint/supervisor.ex:103)
```

```
# Undocumented
```

```
|
```

```
| [U] → Phoenix.Param.Any (lib/phoenix/param.ex:82)
```

```
| [U] → Phoenix.Param.Map (lib/phoenix/param.ex:75)
```


```
| [U] ↓ Mix.Phoenix.Schema.valid?/1 (lib/mix/phoenix/schema.ex:41)
```

You might want to look at these files:

```
| lib/phoenix/controller.ex
```

```
| lib/phoenix/test/conn_test.ex
```

no overall grade!

Grade distribution (undocumented, C, B, A): 



# USING INCH IN CREDO

[https://github.com/rrrene/credo\\_inch](https://github.com/rrrene/credo_inch)

```
$ mix credo
```

## Code Readability

```
[R] → Modules should have a @moduledoc tag.
```

```
lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

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

Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues.

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

```

defmodule CredoInch do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> prepend_task(:initialize_plugins,
                  CredoInch.EnsureProjectIsCompiled)
  end
end

```

```

defmodule CredoInch do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> prepend_task(:initialize_plugins,
                  CredoInch.EnsureProjectIsCompiled)
  end
end
    
```

```

defmodule CredoInch do
  @config_file File.read!(".credo.exs")

  import Credo.Plugin

  def init(exec) do
    exec
    |> register_default_config(@config_file)
    |> prepend_task(:initialize_plugins,
                  CredoInch.EnsureProjectIsCompiled)
  end
end
    
```

```
defmodule CredoInch.EnsureProjectIsCompiled do
  @moduledoc false

  use Credo.Execution.Task

  def call(exec, _opts) do
    Mix.Task.run("compile")

    exec
  end
end
```

```
%{
  configs: [
    %{
      name: "default",
      checks: [
        {Credo.Check.Readability.ModuleDoc, false},
        {CredoInch.Checks.Readability.Doc, []}
      ]
    }
  ]
}
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```



## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])

    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```

## PLUGIN SOURCE CODE

# early draft,  
# subject to change

```
defmodule CredoInch.Checks.Readability.Doc do
  @default_params [min_priority: 1]

  use Credo.Check, category: :readability, run_on_all: true

  def run(source_files, exec, params \\ []) do
    min_priority = Params.get(params, :min_priority, @default_params)

    InchEx.run([])
    |> Enum.filter(&filter_by_min_priority(&1, min_priority))
    |> Enum.map(&inch_result_to_issue/1)
    |> Enum.reject(&is_nil/1)
    |> Enum.each(&append_issue_via_issue_service(&1, exec))

    exec
  end
end
```



```
defp filter_by_min_priority(%{"priority" => priority}, min_priority) do
  priority >= min_priority
end
```

```
defp inch_result_to_issue(inch_result, source_files, params) do
  name = inch_result["name"]
  grade = inch_result["grade"]
  trigger = inch_result["trigger"]
  location = String.split(inch_result["location"], ":")
  filename = Enum.at(location, 0)
  source_file = Enum.find(source_files, &(&1.filename == filename))

  line_no =
    location
    |> Enum.at(1)
    |> String.to_integer()

  issue_for(name, grade, line_no, trigger, source_file, params)
end
```

```
defp issue_for(_name, _grade, _line_no, _trigger, nil, _params), do: nil

defp issue_for(name, grade, line_no, trigger, source_file, params) do
  issue_meta = IssueMeta.for(source_file, params)

  format_issue(issue_meta,
    message: "The documentation for `#{name}` should be improved (grade is '#{grade}']",
    trigger: trigger,
    line_no: line_no
  )
end
```

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoInch, []}  
      ]  
    }  
  ]  
}
```

```
$ mix credo
```

## Code Readability

```
[R] → The documentation for `HtmlSanitizeEx.strip_tags/1` should be improved  
      (current grade is 'U')  
      lib/mix/tasks/my_task.ex:1:11 (Mix.Tasks.MyTask)
```

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

Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)

344 mods/funs, found 1 code readability issues.

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

```
%{  
  configs: [  
    %{  
      name: "default",  
      plugins: [  
        {CredoCheckFilterConfig, []},  
        {CredoInch, []}  
      ],  
      checks: [  
        {CredoInch.Checks.Readability.Doc,  
          [excluded: [~r"_task\.ex$"]]}  
      ]  
    }  
  ]  
}
```

```
$ mix credo
```

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

```
Analysis took 0.04 seconds (0.03s to load, 0.01s running 1 check)
```

```
344 mods/funs, found no issues.
```

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

```
1 issue was filtered.
```

# MORE TOOLS





# JSON OUTPUT

## SHELL

```
$ mix credo info
```

System:

Credo: 1.1.4

Elixir: 1.9.1

Erlang: 21

```
$ mix credo info --format json
```

```
{  
  "system": {  
    "credo": "1.1.4",  
    "elixir": "1.9.1",  
    "erlang": "21"  
  }  
}
```

```
$ mix credo lib/mix/tasks/my_task.ex:2:3
```

```
[R] Category: readability
```

```
➤ Priority: high
```

```
Large numbers should be written with underscores: 314_159_265
```

```
lib/mix/tasks/my_task.ex:2:3 (Mix.Tasks.MyTask)
```

```
-- CODE IN QUESTION
```

```
1 defmodule Mix.Tasks.MyTask do
```

```
2   314159265
```

```
   ^^^^^^^^
```

```
3 end
```

```
4
```

```
-- WHY IT MATTERS
```

Numbers can contain underscores for readability purposes.

These do not affect the value of the number, but can help read large numbers more easily.

```
141592654 # how large is this number?
```

```
{
  "explanations": [
    {
      "category": "readability",
      "check": "Elixir.Credo.Check.Readability.ModuleDoc",
      "filename": "lib/mix/tasks/my_task.ex",
      "line_no": 2,
      "column": 3,
      "explanation_for_issue": "Numbers can contain underscores for readability purposes.\nThese do not affect the value of the number, but can help read large numbers\nmore easily.\n\n    141592654 # how large is this number?\n\n    ...",
      "message": "Large numbers should be written with underscores: 314_159_265",
      "priority": 2,
      "related_code": [...
```

# EXIT STATUSES

```
consistency: 1
design:       2
readability: 4
refactor:    8
warning:     16
```

## SHELL

```
$ mix credo
```

```
...
```

```
$ echo $?
```

```
12
```

## EXIT STATUSES

```
consistency: 1  
design:      2  
readability: 4  
refactor:   8  
warning:    16
```

## SHELL

```
$ mix credo --mute-exit-status
```

```
...
```

```
$ echo $?
```

```
0
```

## EXIT STATUSES

consistency:	1
design:	2
readability:	4
refactor:	8
warning:	16



# **MULTIPLE CONFIGURATIONS**

## CONFIG

```
# .credo.exs
%{
  configs: [
    %{
      name: "default",
      checks: [...]
    },
    %{
      name: "CI",
      strict: true
    }
  ]
}
```

## SHELL

```
$ mix credo --config-name CI
```

# ROADMAP FOR 2.0

# Homework for Credo 2.0

- Separating backend and frontend
- Rethinking check execution
- Rethinking issue creation
- Providing plugin test utilities
- Shaping the newcomer experience



# CREDO: YOUR TEAM'S ROBOT MENTOR

