elixir

Concurrent Programming

Ruben Amortegui
@ramortegui
http://rubenamortegui.com
https://github.com/ramortegui

paradem

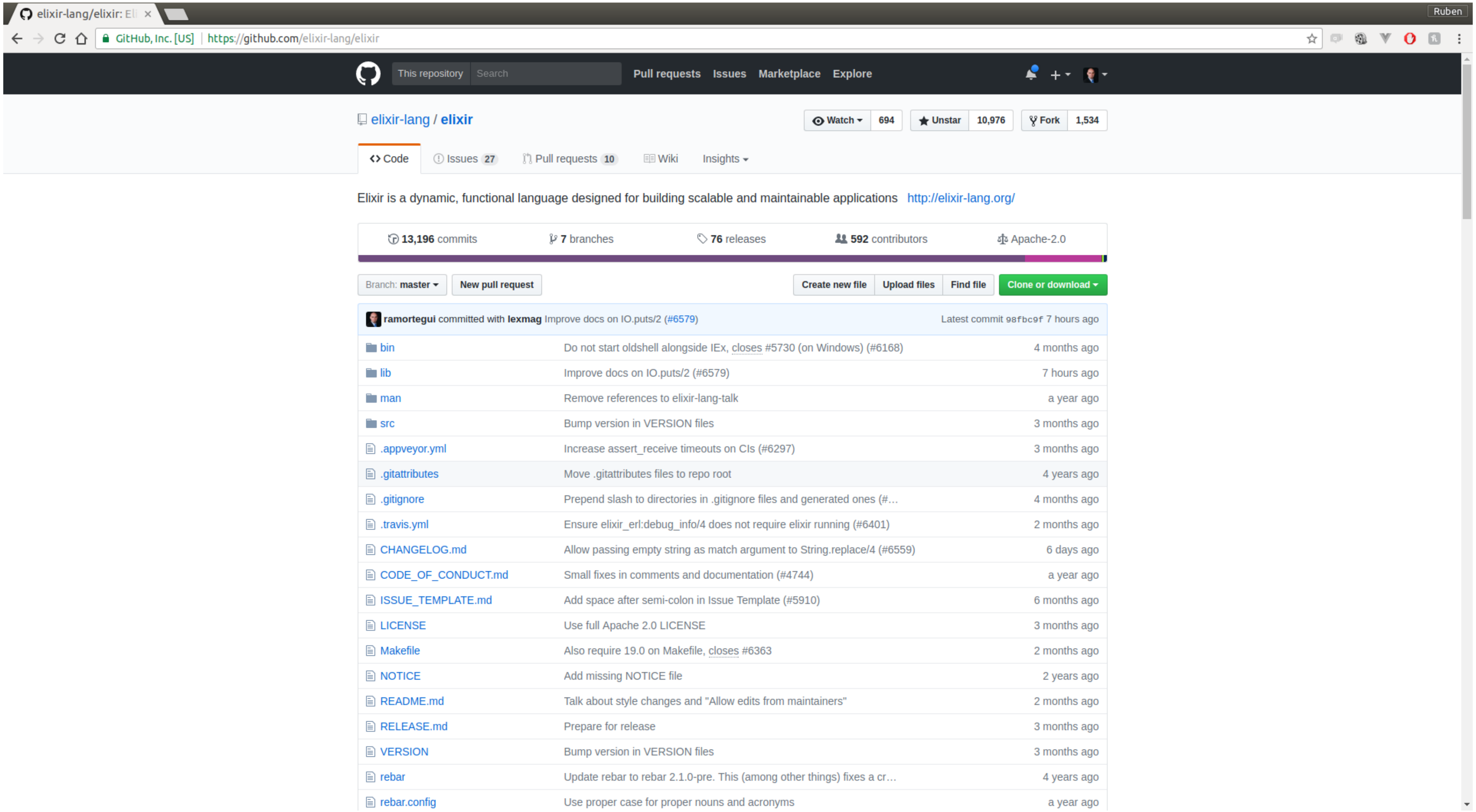
Lonestar ElixirConf

Austin, TX, February 10-11, 2018

- Videos https://www.youtube.com/channel/UCOy-_b9bqjokoWX9Hg5ZgUg/featured
  - Consistent, Distributed Elixir - Chris Keathley
  - Keynote - Aaron Patterson
  - Usercentered API Versioning - Niall Burkley

Elixir 1.6.4

- Code formater
- Dynamic supervisor
- Enhacements
- Bug Fixes


elixir-lang/elixir: El

GitHub, Inc. [US] | https://github.com/elixir-lang/elixir

This repository Search Pull requests Issues Marketplace Explore

elixir-lang / **elixir**

Watch 694 | Unstar 10,976 | Fork 1,534

Code | Issues 27 | Pull requests 10 | Wiki | Insights

Elixir is a dynamic, functional language designed for building scalable and maintainable applications http://elixir-lang.org/

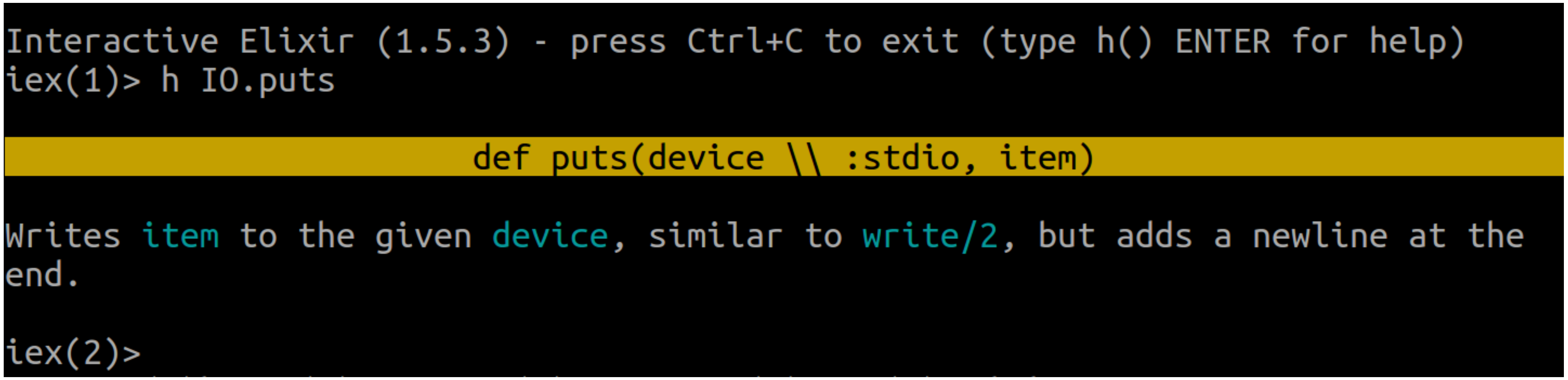
13,196 commits | **7** branches | **76** releases | **592** contributors | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

ramortegui committed with **lexmag** Improve docs on IO.puts/2 (#6579) — Latest commit `98fbc9f` 7 hours ago

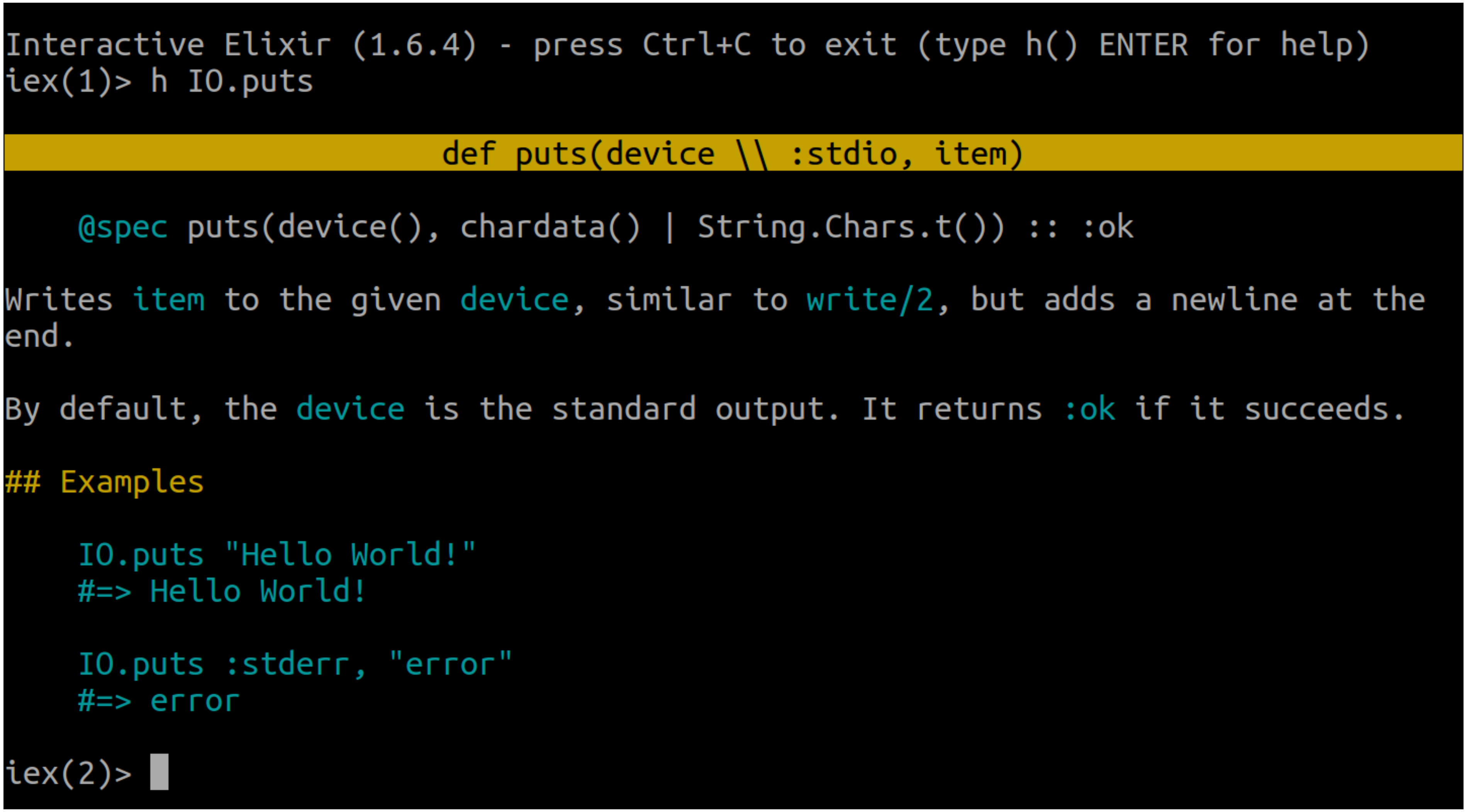
bin	Do not start oldshell alongside IEx, closes #5730 (on Windows) (#6168)	4 months ago
lib	Improve docs on IO.puts/2 (#6579)	7 hours ago
man	Remove references to elixir-lang-talk	a year ago
src	Bump version in VERSION files	3 months ago
.appveyor.yml	Increase assert_receive timeouts on CIs (#6297)	3 months ago
.gitattributes	Move .gitattributes files to repo root	4 years ago
.gitignore	Prepend slash to directories in .gitignore files and generated ones (#…	4 months ago
.travis.yml	Ensure elixir_erl:debug_info/4 does not require elixir running (#6401)	2 months ago
CHANGELOG.md	Allow passing empty string as match argument to String.replace/4 (#6559)	6 days ago
CODE_OF_CONDUCT.md	Small fixes in comments and documentation (#4744)	a year ago
ISSUE_TEMPLATE.md	Add space after semi-colon in Issue Template (#5910)	6 months ago
LICENSE	Use full Apache 2.0 LICENSE	3 months ago
Makefile	Also require 19.0 on Makefile, closes #6363	2 months ago
NOTICE	Add missing NOTICE file	2 years ago
README.md	Talk about style changes and "Allow edits from maintainers"	2 months ago
RELEASE.md	Prepare for release	3 months ago
VERSION	Bump version in VERSION files	3 months ago
rebar	Update rebar to rebar 2.1.0-pre. This (among other things) fixes a cr…	4 years ago
rebar.config	Use proper case for proper nouns and acronyms	a year ago

Help on IO.puts elixir 1.5.3



```
Interactive Elixir (1.5.3) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> h IO.puts

                        def puts(device \\ :stdio, item)

Writes item to the given device, similar to write/2, but adds a newline at the
end.

iex(2)>
```

Help IO.puts elixir 1.6.4



```
Interactive Elixir (1.6.4) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> h IO.puts

                        def puts(device \\ :stdio, item)

    @spec puts(device(), chardata() | String.Chars.t()) :: :ok

Writes item to the given device, similar to write/2, but adds a newline at the
end.

By default, the device is the standard output. It returns :ok if it succeeds.

## Examples

    IO.puts "Hello World!"
    #=> Hello World!

    IO.puts :stderr, "error"
    #=> error

iex(2)> █
```

Concurrent Programming in Elixir

- **Agenda**
  - Concepts
  - Code samples and implementations
  - Summary

Erlang

- Erlang is a programming language and runtime system for building massively scalable soft real-time systems with requirements on high availability.

  https://github.com/erlang/otp

Elixir

- Elixir is a dynamic, functional language designed for building scalable and maintainable applications.
  - https://elixir-lang.org/

OTP

- OTP is a set of Erlang libraries, which consists of the Erlang runtime system, a number of ready-to-use components mainly written in Erlang, and a set of design principles for Erlang programs.

  https://github.com/erlang/otp

OTP

- OTP stands for Open Telecomunication Platform.
- Is a general purpose tool for developing and managing large systems.
- Provides tools, libraries, conventions and defines a structure for your application.

OTP

- Features included in OTP:
  - Erlang interpreter and compiler
  - Standard libraries
  - Dialyzer, a static analysis tool
  - Mnesia, a distributed database
  - Erlang Term Storage (ETS)
  - A debugger
  - An event tracer
  - A release management tool (hot swap)

Concurrent Programming

- Programs that can handle several threads of execution at the same time.
  - A CPU is processing one thread(or job) at a time.
  - Swaps between jobs a such rate that gives the illusion of running at the same time.

http://erlang.org/doc/getting_started/conc_prog.html

Actor Concurrency Model

- An actor is a computational entity that, in response to a message it receives, can concurrently:
  - Send a finite number of messages to other actors.
  - Create a finite number of new actors.
  - Designate the behaviour to be used for the next message it receives.

Actor Concurrency Model in Erlang

- Each actor is a process.
- Each process performs a specific task.
- To tell a process to do something, you need to send it a message. The process can reply by sending back another message.
- The kinds of messages can act on are specific to the process itself.
- Other than that, processes don’t share any information with other processes.

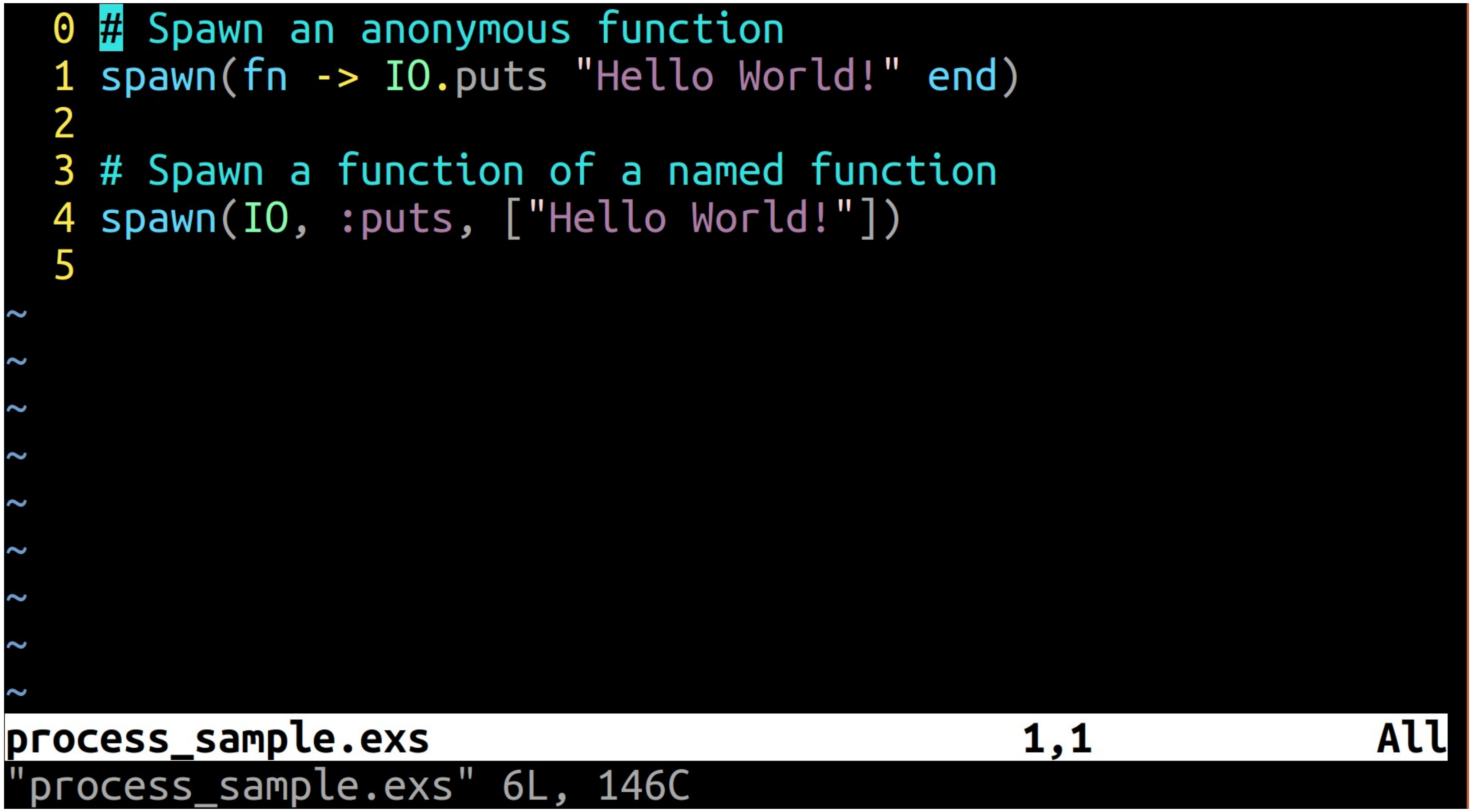
Process

- Fundamental part of concurrency
- Are light weight
- Doesn’t share memory

Processes

- Spawn
  - ‘spawn’: spawns a process and returns his id.
- Send messages
  - ‘send’: sends message to the pid
- Receive
  - ‘receive’ receives and patern match the message
- Process
  - Library to deal with processes

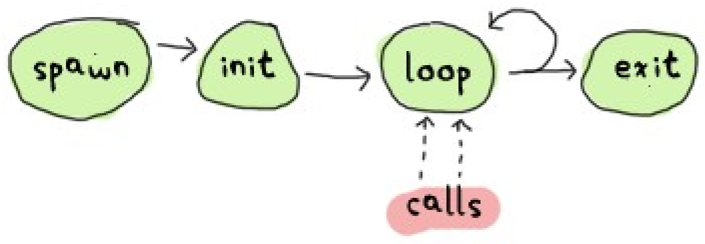
Process Sample



```
# Spawn an anonymous function
spawn(fn -> IO.puts "Hello World!" end)

# Spawn a function of a named function
spawn(IO, :puts, ["Hello World!"])

~
~
~
~
~
~
~
~
~
process_sample.exs                                    1,1              All
"process_sample.exs" 6L, 146C
```

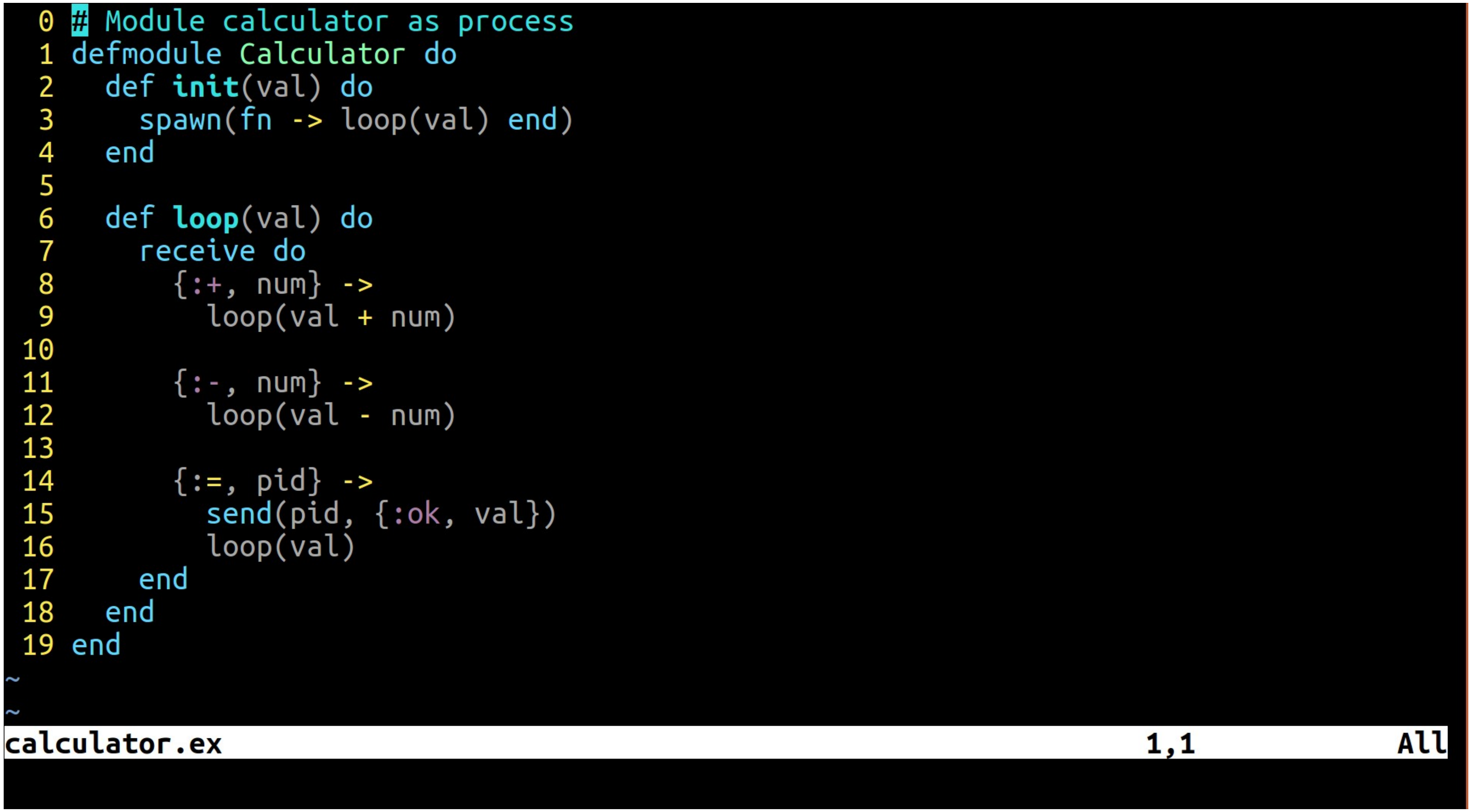
Process

- How to maintain state?
  - Tail recursion
- Have a mailbox queue to process messages



Taken from: http://learnyousomeerlang.com/event-handlers

Sample of process maintaing state



```
# Module calculator as process
defmodule Calculator do
  def init(val) do
    spawn(fn -> loop(val) end)
  end

  def loop(val) do
    receive do
      {:+, num} ->
        loop(val + num)

      {:-, num} ->
        loop(val - num)

      {:=, pid} ->
        send(pid, {:ok, val})
        loop(val)
    end
  end
end
~
~
calculator.ex                                                        1,1           All
```

HOME INSTALL GUIDES LEARNING DOC

elixir

```
defprotocol String.Inspect
  only: [BitString, List,

defimpl String.Inspect, fo
  def inspect(false), do:
  def inspect(true),  do:
  def inspect(nil),   do:
  def inspect(:""),   do:

  def inspect(atom) do
```

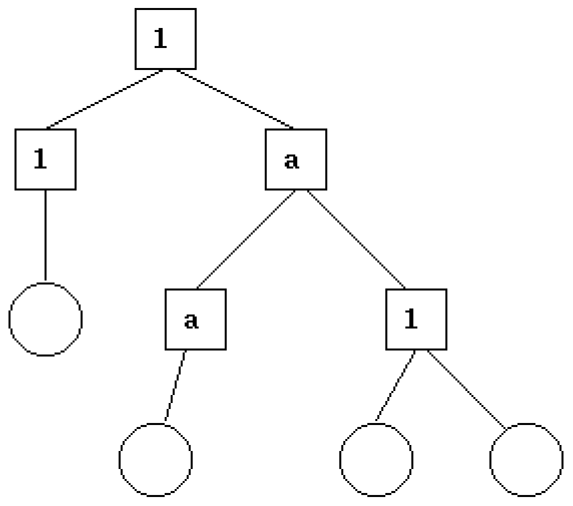
Elixir is a dynamic, functional language designed for building scalable and maintainable applications.

Elixir leverages the Erlang VM, known for running low-latency, distributed and fault-tolerant systems, while also being successfully used in web development and the embedded software domain.

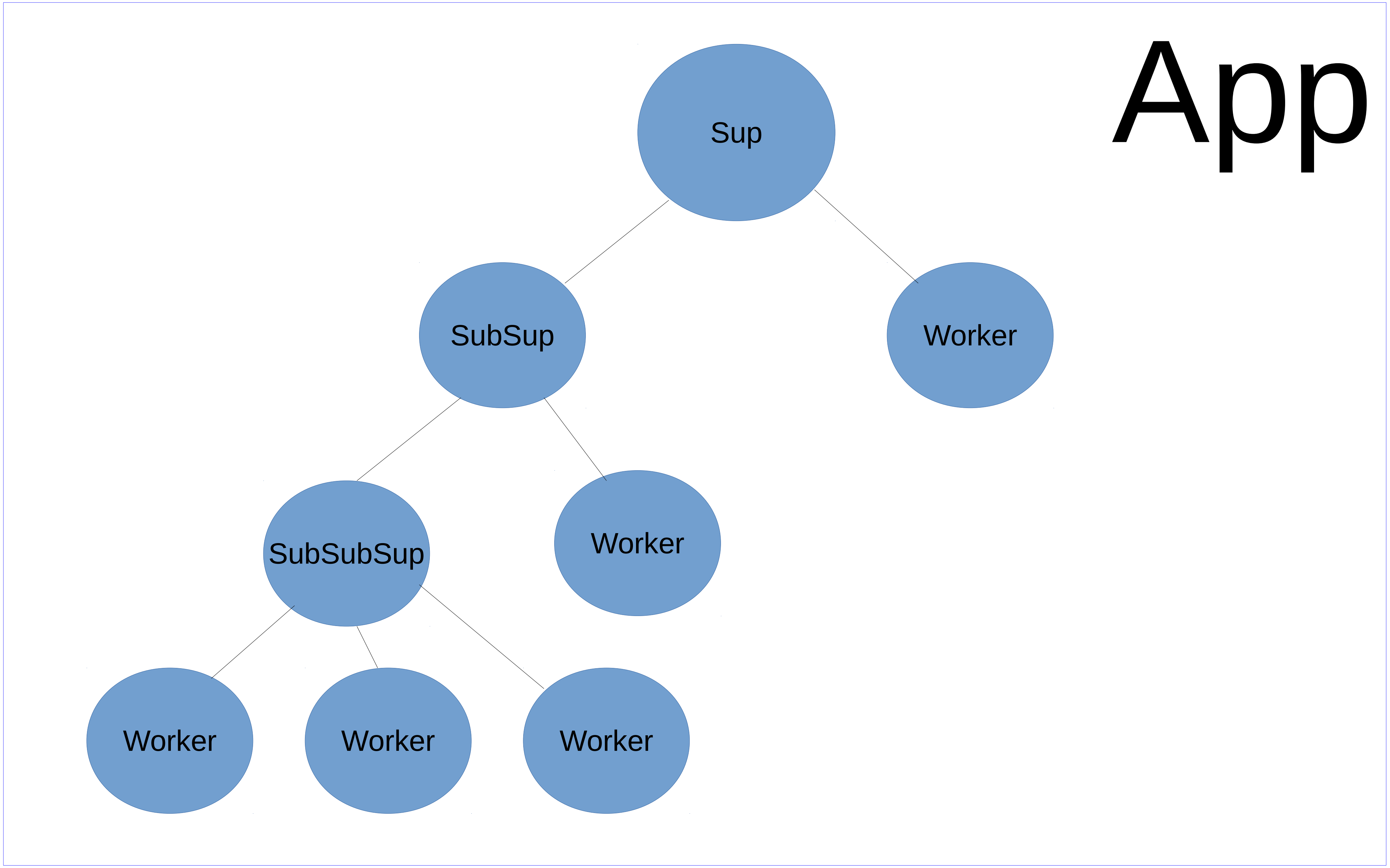
Suprevision Trees

http://erlang.org/doc/design_principles/des_princ.html

- The square boxes represent supervisors.
- The circles represent workers.
- 1 and a: represents strategy



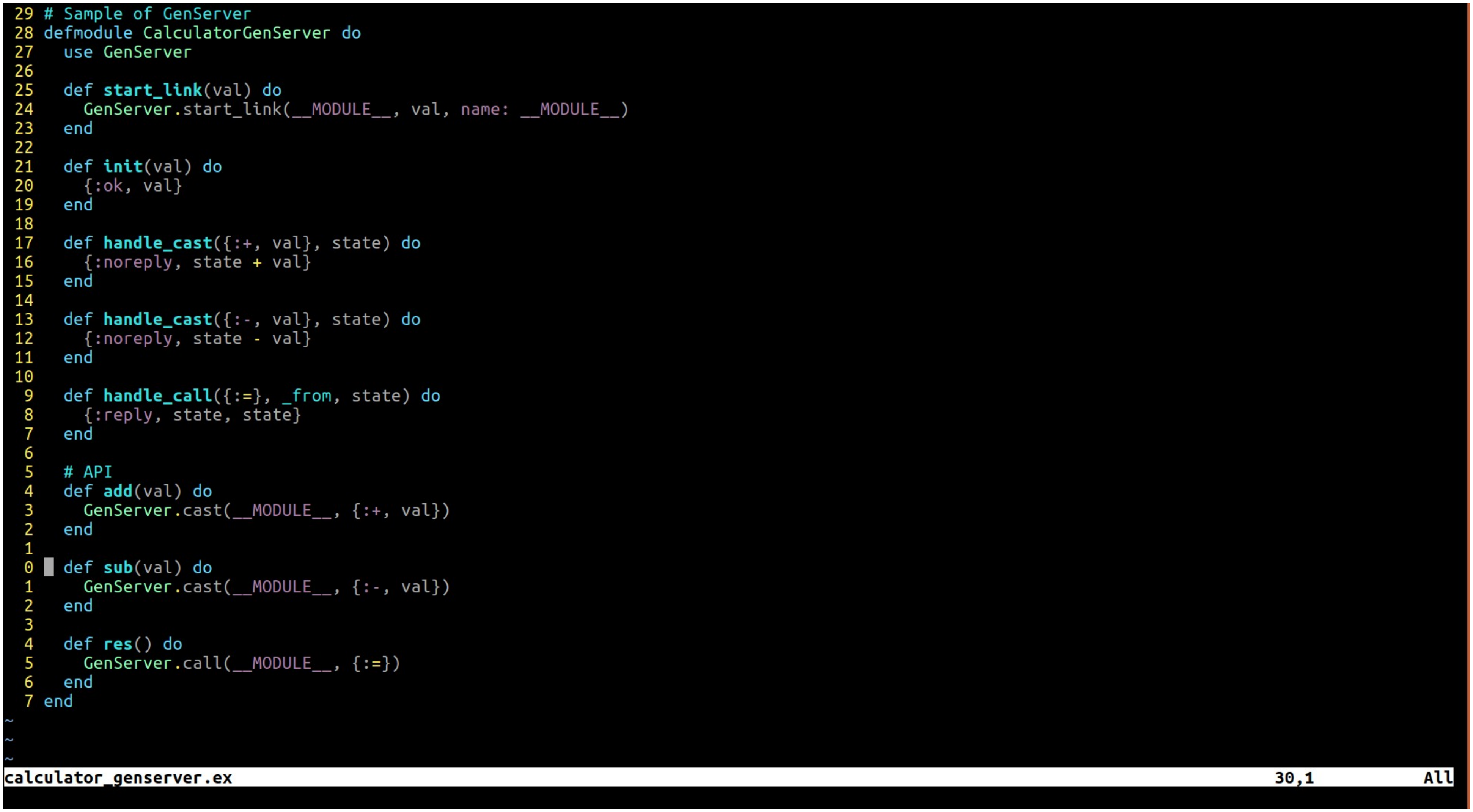
OTP Application



Behaviours

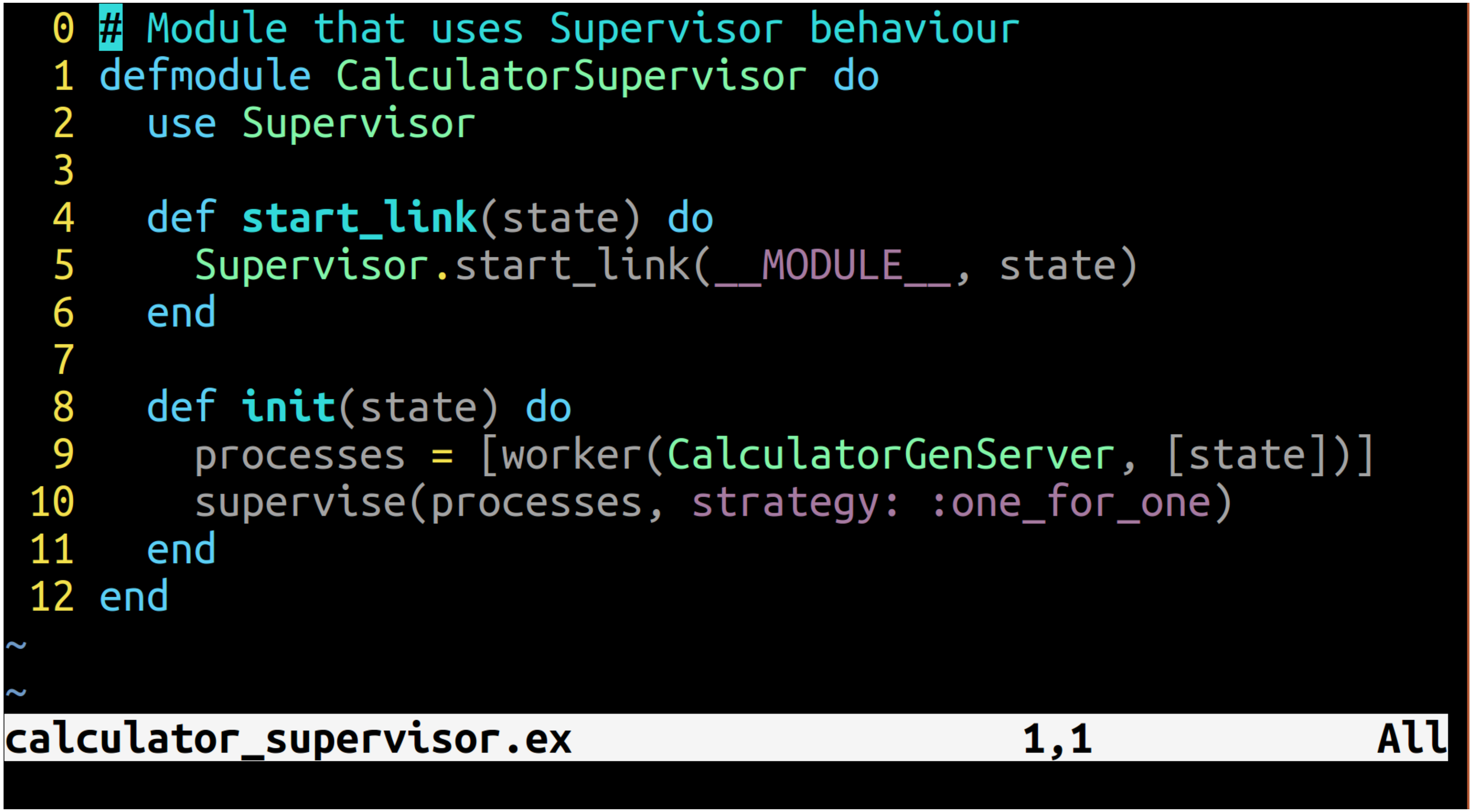
- Code that implements a common pattern.
- OTP provides:
  - GenServer
  - Supervisor
  - Application

GenServer



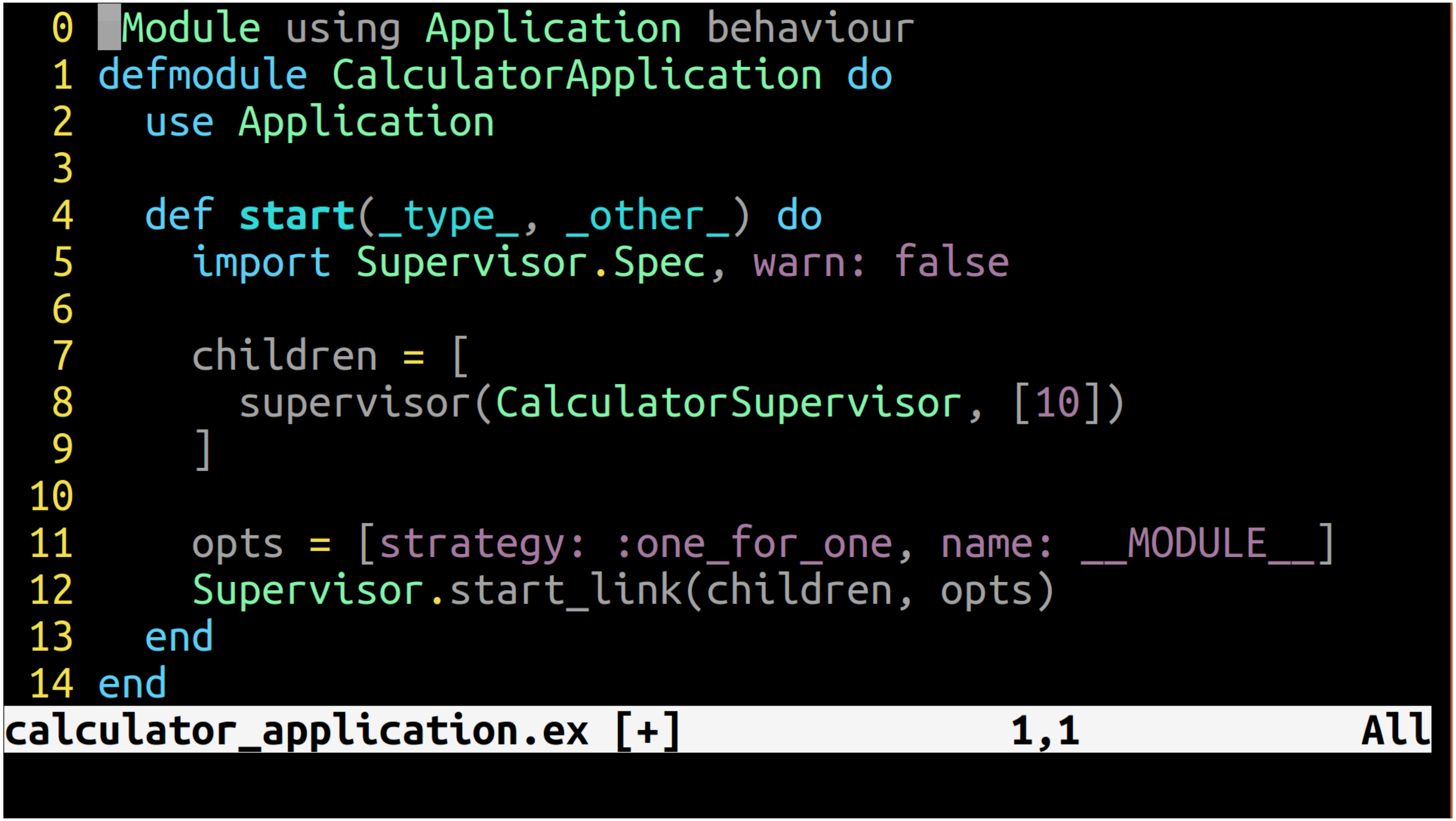
```
# Sample of GenServer
defmodule CalculatorGenServer do
  use GenServer

  def start_link(val) do
    GenServer.start_link(__MODULE__, val, name: __MODULE__)
  end

  def init(val) do
    {:ok, val}
  end

  def handle_cast({:+, val}, state) do
    {:noreply, state + val}
  end

  def handle_cast({:-, val}, state) do
    {:noreply, state - val}
  end

  def handle_call({:=}, _from, state) do
    {:reply, state, state}
  end

  # API
  def add(val) do
    GenServer.cast(__MODULE__, {:+, val})
  end

  def sub(val) do
    GenServer.cast(__MODULE__, {:-, val})
  end

  def res() do
    GenServer.call(__MODULE__, {:=})
  end
end
~
~
~
calculator_genserver.ex                                                    30,1          All
```

Supervisor



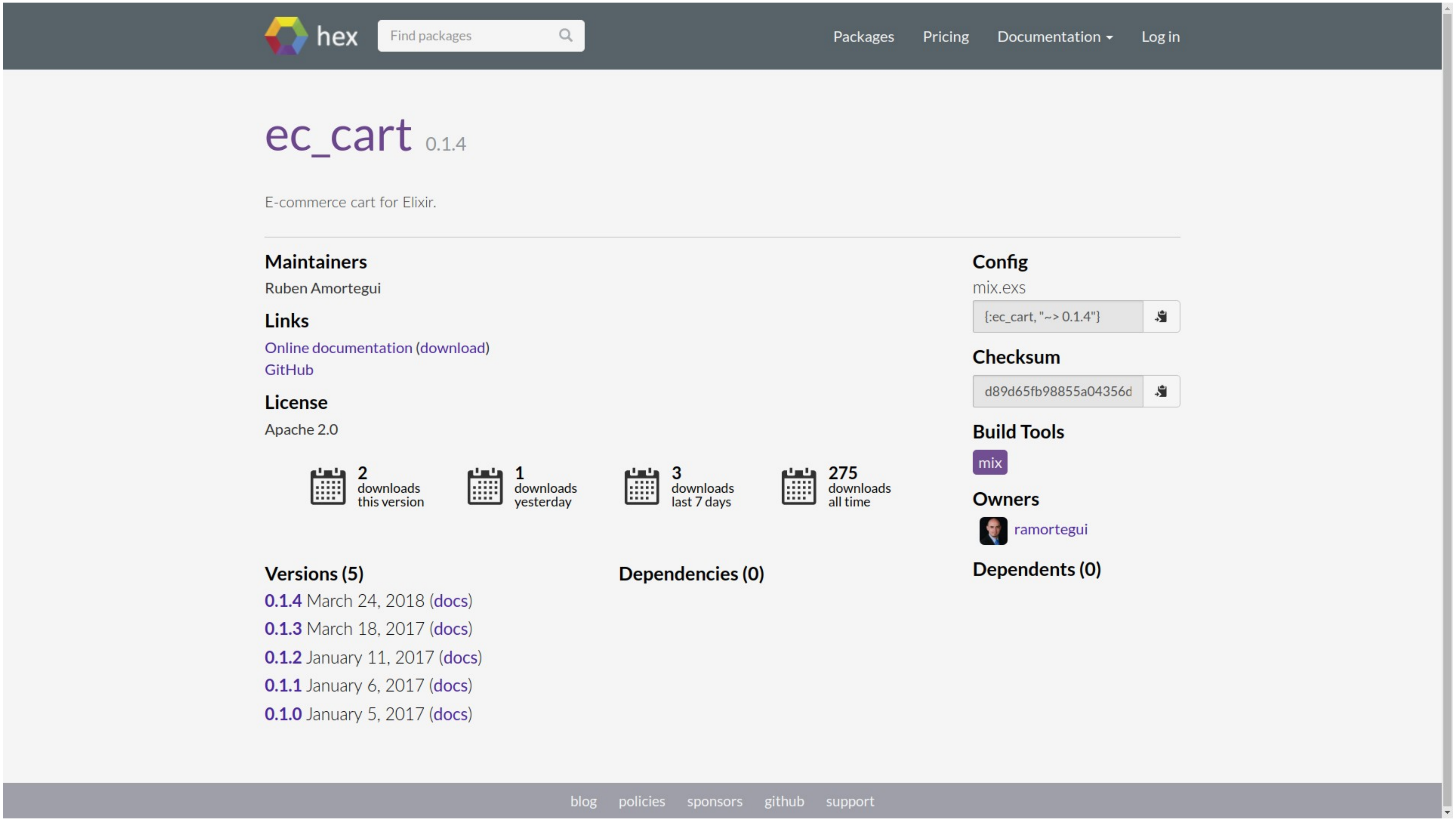
```
# Module that uses Supervisor behaviour
defmodule CalculatorSupervisor do
  use Supervisor

  def start_link(state) do
    Supervisor.start_link(__MODULE__, state)
  end

  def init(state) do
    processes = [worker(CalculatorGenServer, [state])]
    supervise(processes, strategy: :one_for_one)
  end
end
~
~
calculator_supervisor.ex                                1,1           All
```

Application



```
Module using Application behaviour
defmodule CalculatorApplication do
  use Application

  def start(_type_, _other_) do
    import Supervisor.Spec, warn: false

    children = [
      supervisor(CalculatorSupervisor, [10])
    ]

    opts = [strategy: :one_for_one, name: __MODULE__]
    Supervisor.start_link(children, opts)
  end
end
calculator_application.ex [+]                 1,1           All
```

EcCart App



hex | Find packages | Packages | Pricing | Documentation | Log in

ec_cart 0.1.4

E-commerce cart for Elixir.

Maintainers
Ruben Amortegui

Links
Online documentation (download)
GitHub

License
Apache 2.0

2 downloads this version
1 downloads yesterday
3 downloads last 7 days
275 downloads all time

Config
mix.exs

```
{:ec_cart, "~> 0.1.4"}
```

Checksum

```
d89d65fb98855a04356d
```

Build Tools
mix

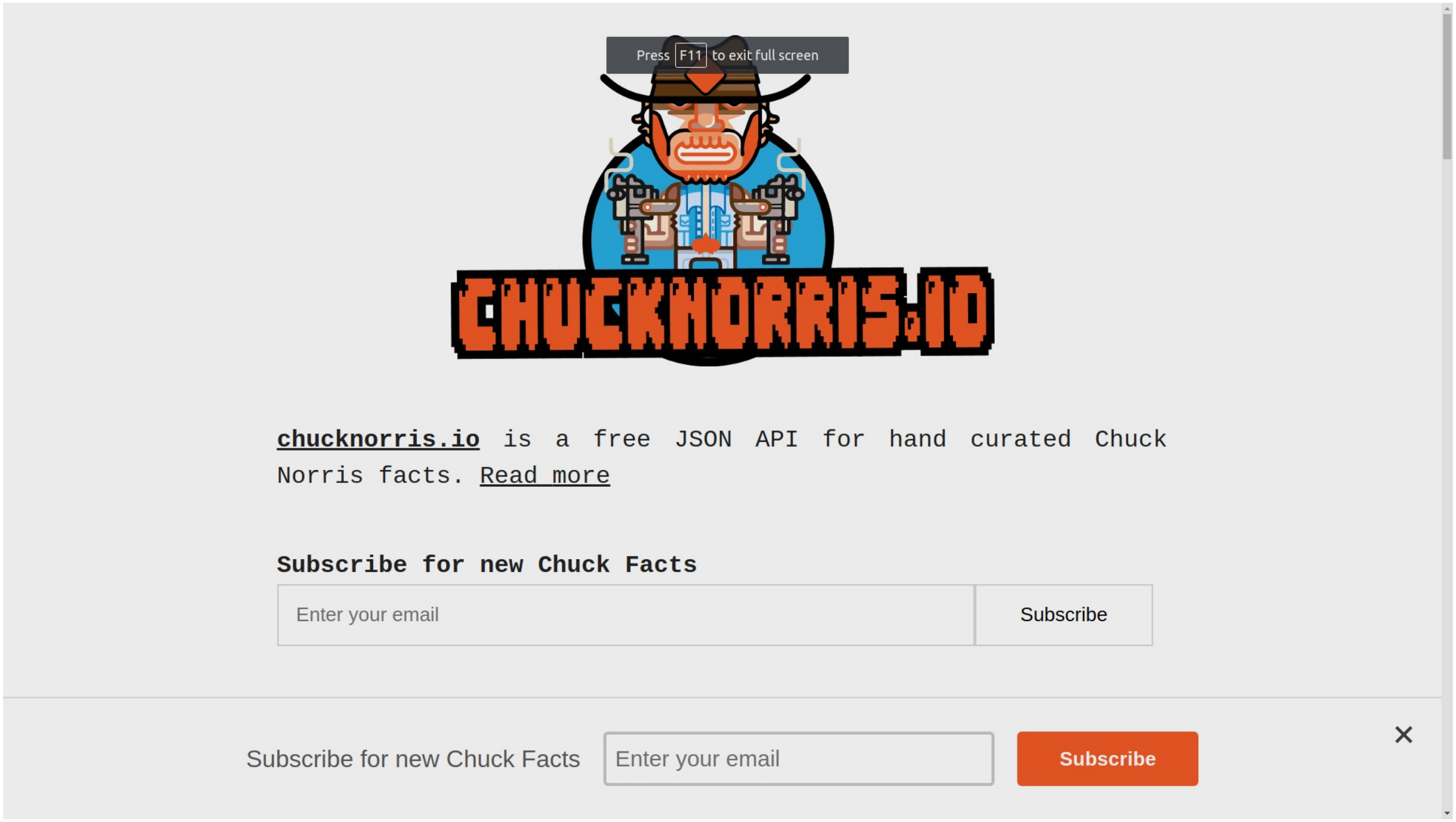
Owners
ramortegui

Dependents (0)

Versions (5)
0.1.4 March 24, 2018 (docs)
0.1.3 March 18, 2017 (docs)
0.1.2 January 11, 2017 (docs)
0.1.1 January 6, 2017 (docs)
0.1.0 January 5, 2017 (docs)

Dependencies (0)

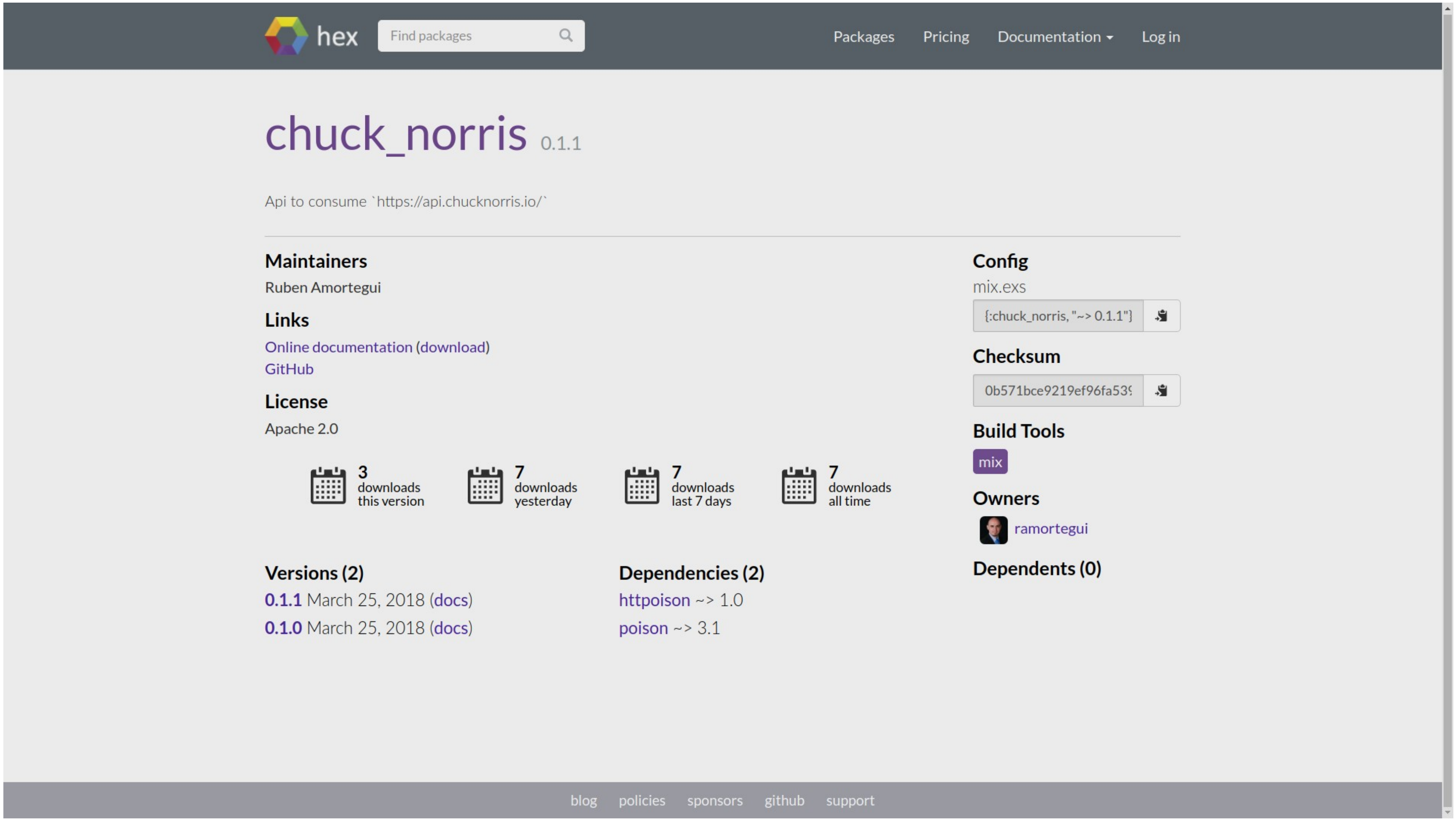
blog policies sponsors github support



chucknorris.io is a free JSON API for hand curated Chuck Norris facts. Read more

Subscribe for new Chuck Facts

Subscribe for new Chuck Facts



hex Find packages Packages Pricing Documentation Log in

chuck_norris 0.1.1

Api to consume `https://api.chucknorris.io/`

Maintainers
Ruben Amortegui

Links
Online documentation (download)
GitHub

License
Apache 2.0

3 downloads this version
7 downloads yesterday
7 downloads last 7 days
7 downloads all time

Versions (2)
0.1.1 March 25, 2018 (docs)
0.1.0 March 25, 2018 (docs)

Dependencies (2)
httpoison ~> 1.0
poison ~> 3.1

Config
mix.exs

```
{:chuck_norris, "~> 0.1.1"}
```

Checksum

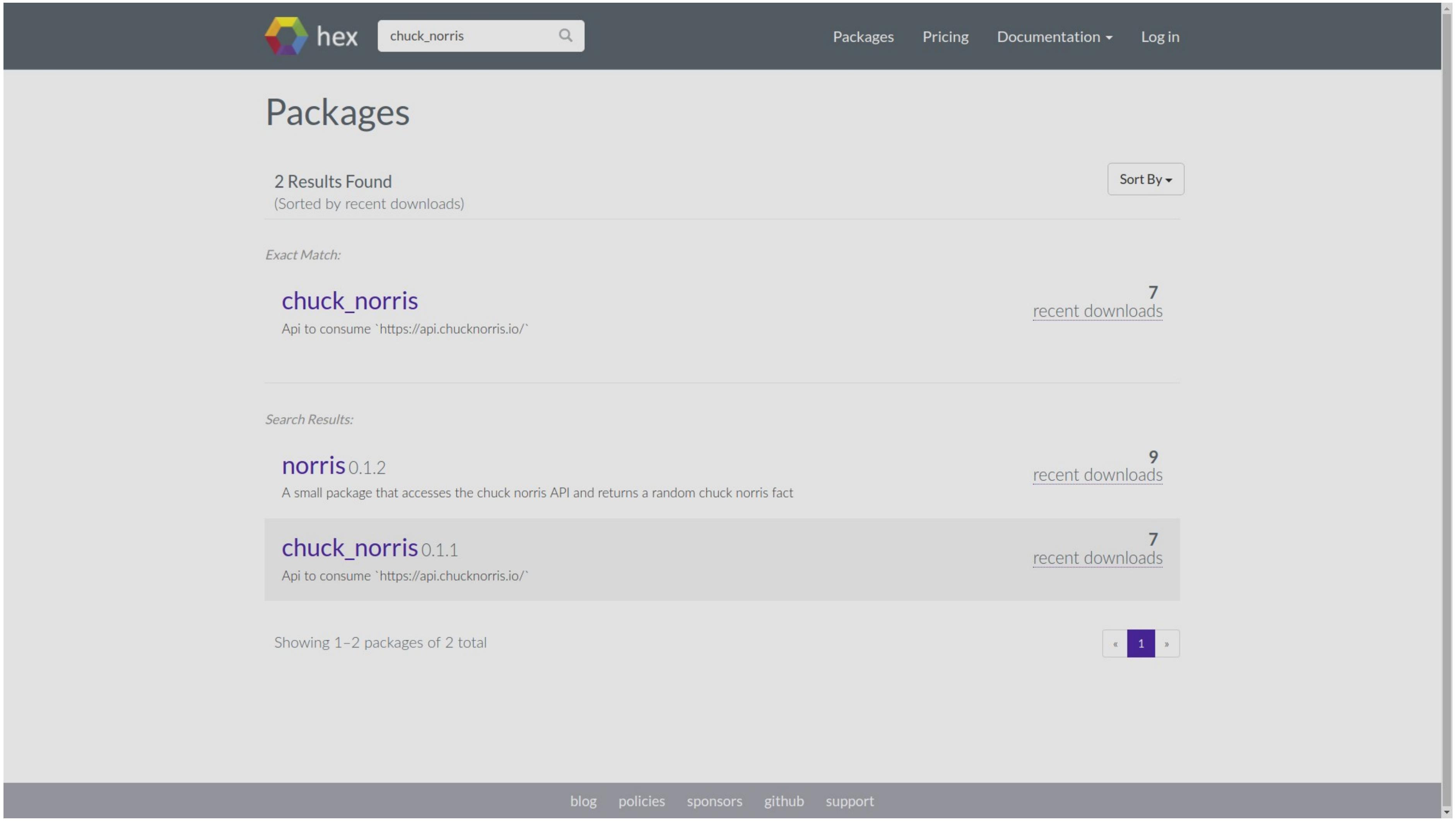
```
0b571bce9219ef96fa53
```

Build Tools
mix

Owners
ramortegui

Dependents (0)

blog policies sponsors github support



Packages

2 Results Found
(Sorted by recent downloads)

Exact Match:

chuck_norris
Api to consume `https://api.chucknorris.io/`
7
recent downloads

Search Results:

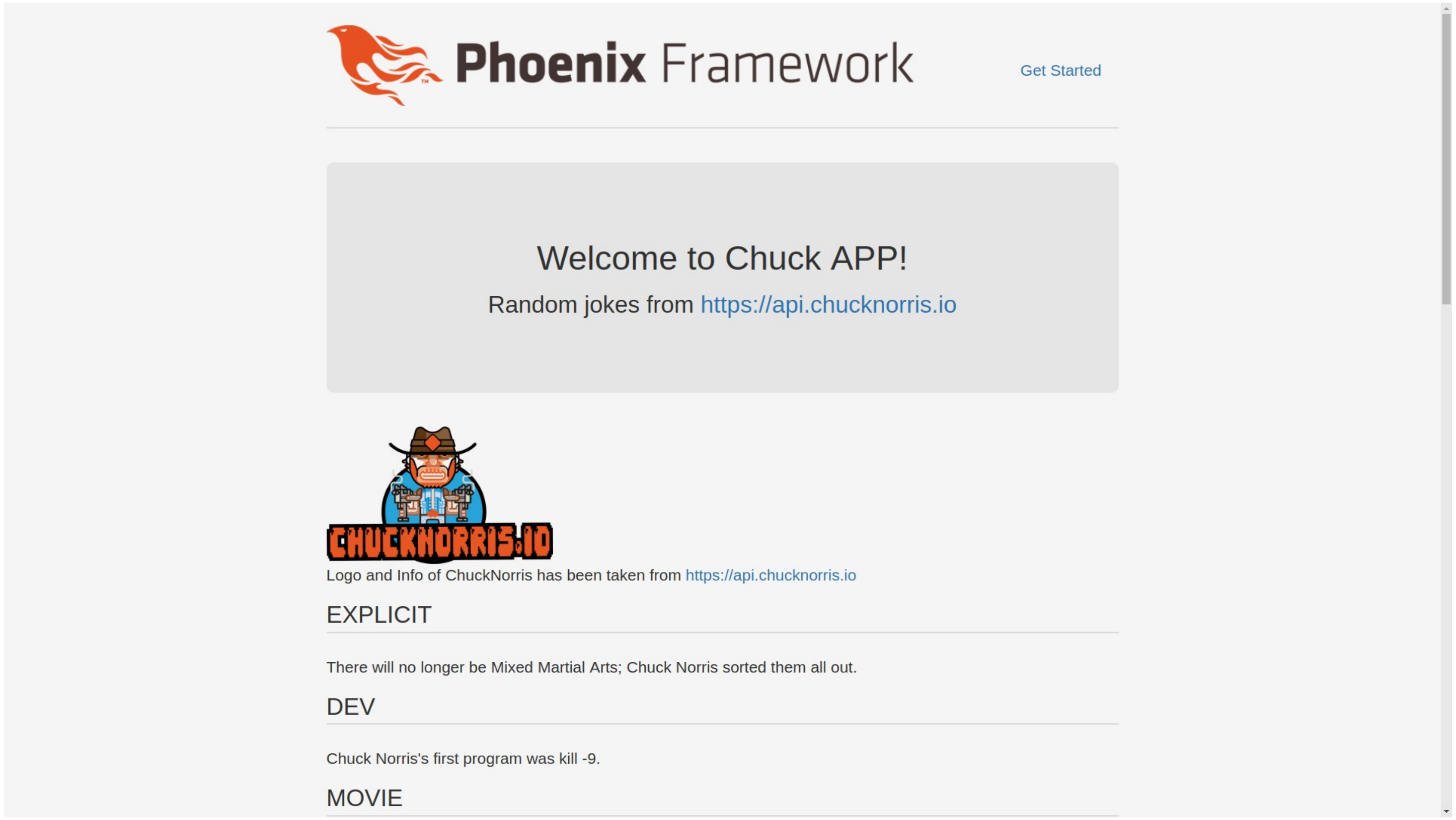
norris 0.1.2
A small package that accesses the chuck norris API and returns a random chuck norris fact
9
recent downloads

chuck_norris 0.1.1
Api to consume `https://api.chucknorris.io/`
7
recent downloads

Showing 1–2 packages of 2 total

Chuck Norris APP

- https://phoenix-chuck-norris.herokuapp.com



Welcome to Chuck APP!

Random jokes from https://api.chucknorris.io

Logo and Info of ChuckNorris has been taken from https://api.chucknorris.io

EXPLICIT

There will no longer be Mixed Martial Arts; Chuck Norris sorted them all out.

DEV

Chuck Norris's first program was kill -9.

MOVIE

Summary

- Processes are the basic primitive of concurrency in Elixir.
- Elixir uses the actor concurrency model.
- The OTP behaviours makes ‘hard’ tasks ‘easy’ tasks.

References

- Jurić, S. (2015). Elixir in action. Shelter Island, NY: Manning Publications.
- Thomas, D. (2016). Programming Elixir 1.3: functional, concurrent, pragmatic, fun. Releigh, NC: Pragmatic Bookshelf.
- Tan Wei Hao, B.(2017). The little Elixir & OTP Guidebook. Shelter Island, NY: Manning Publications.

Thanks!

Q & A?

@ramortegui

http://rubenamortegui.com

https://github.com/ramortegui