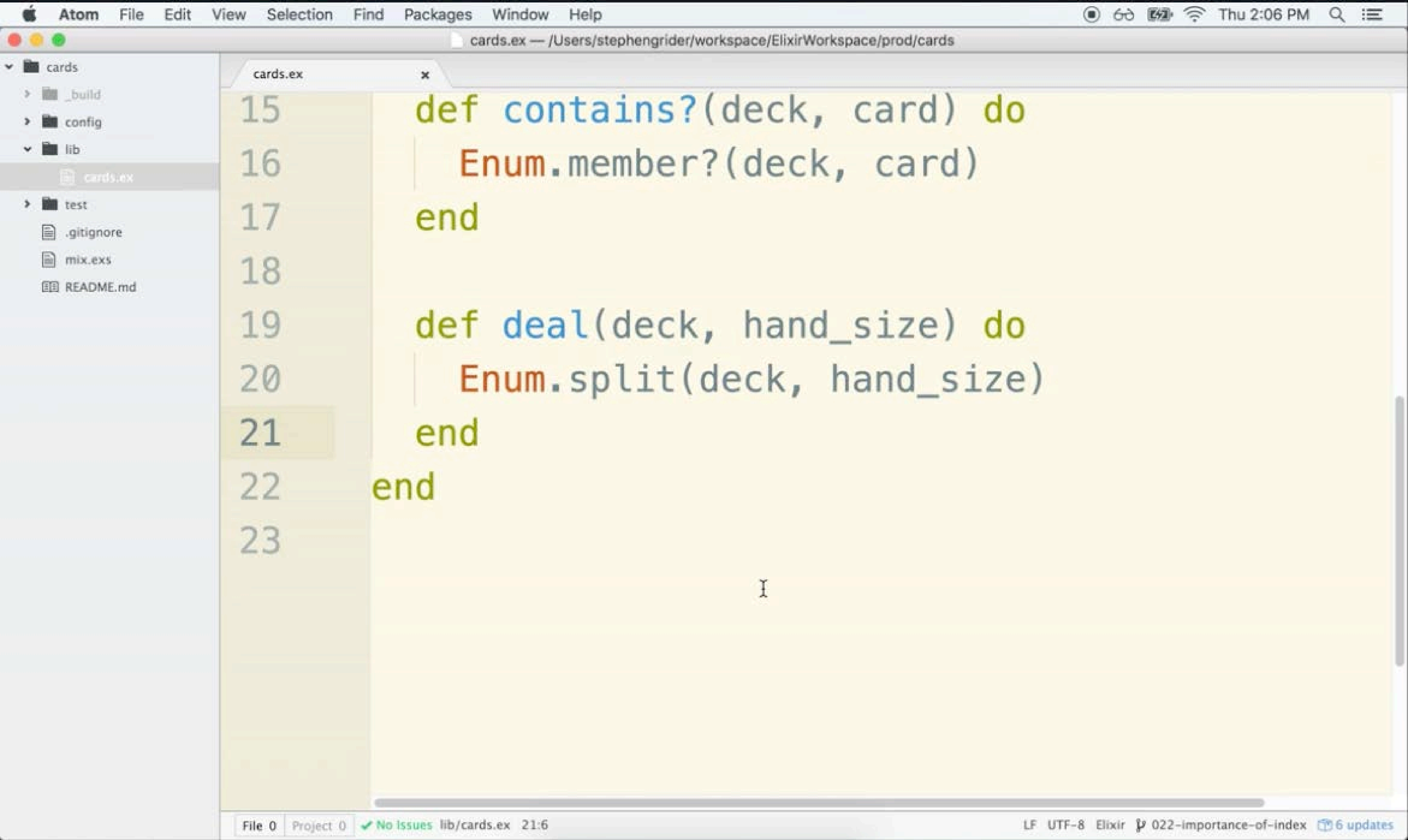
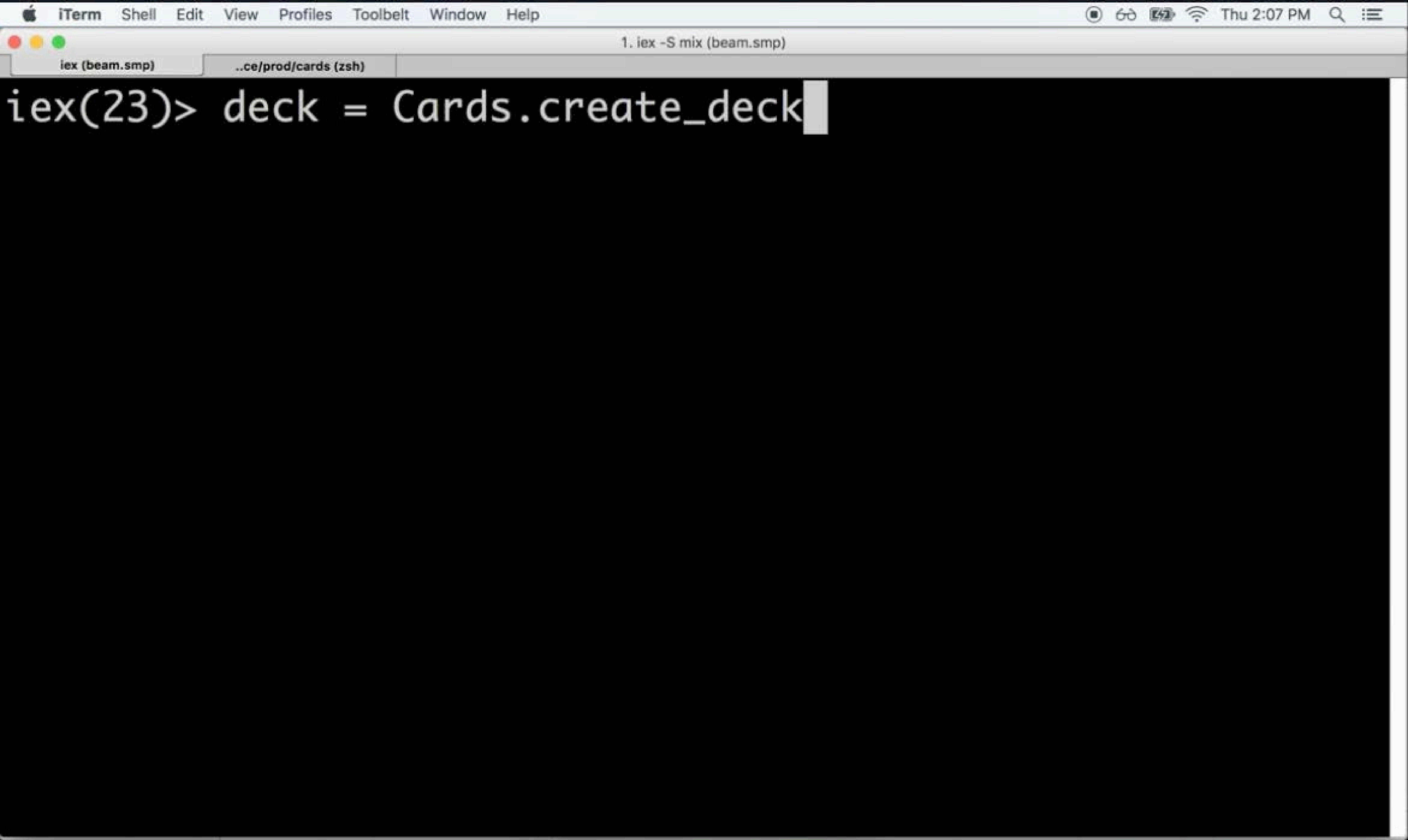
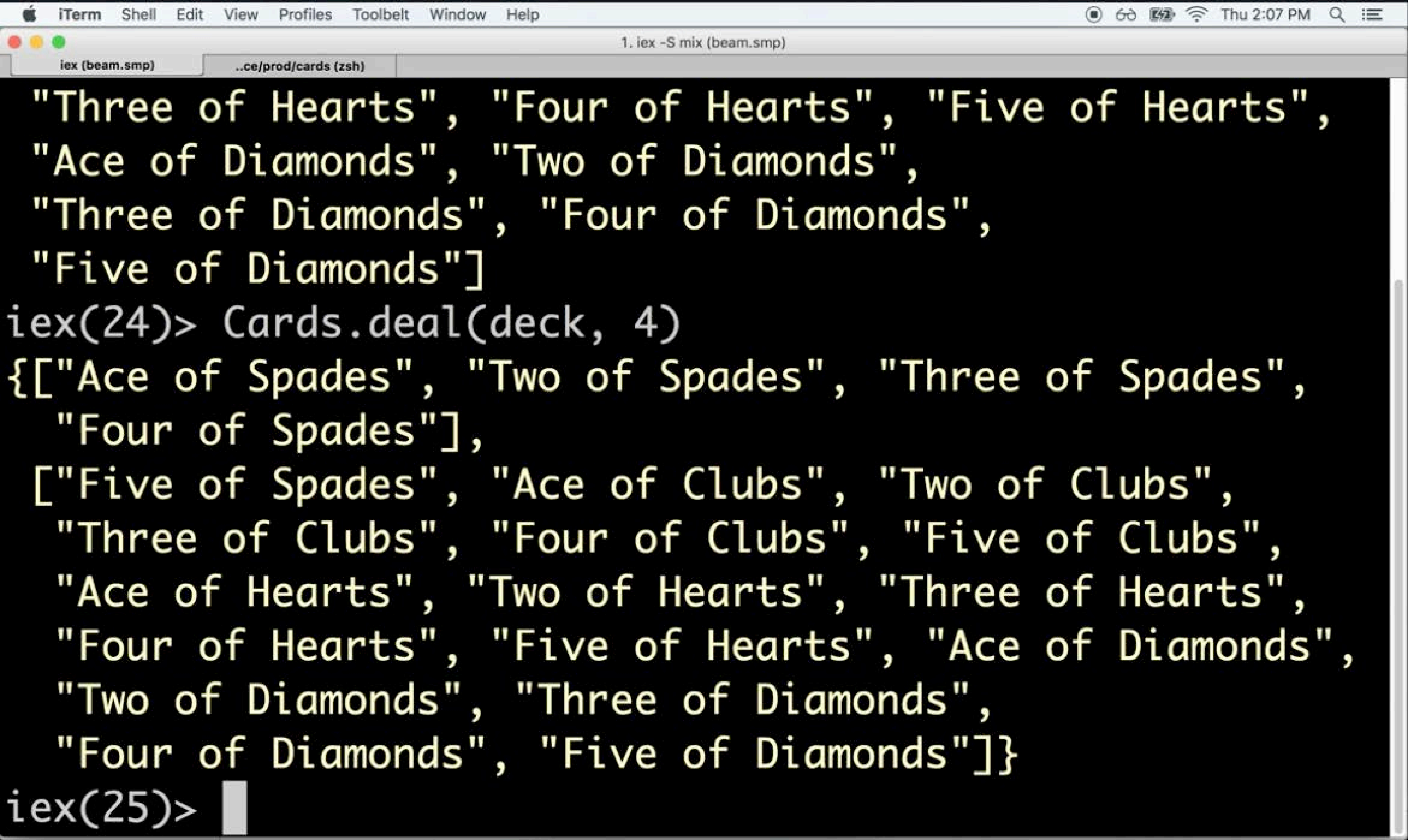

Atom File Edit View Selection Find Packages Window Help
Thu 2:06 PM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
lib
cards.ex
test
.gitignore
mix.exs
README.md
cards.ex
File 0 Project 0 No Issues lib/cards.ex 21:6
LF UTF-8 Elixir 022-importance-of-index 6 updates

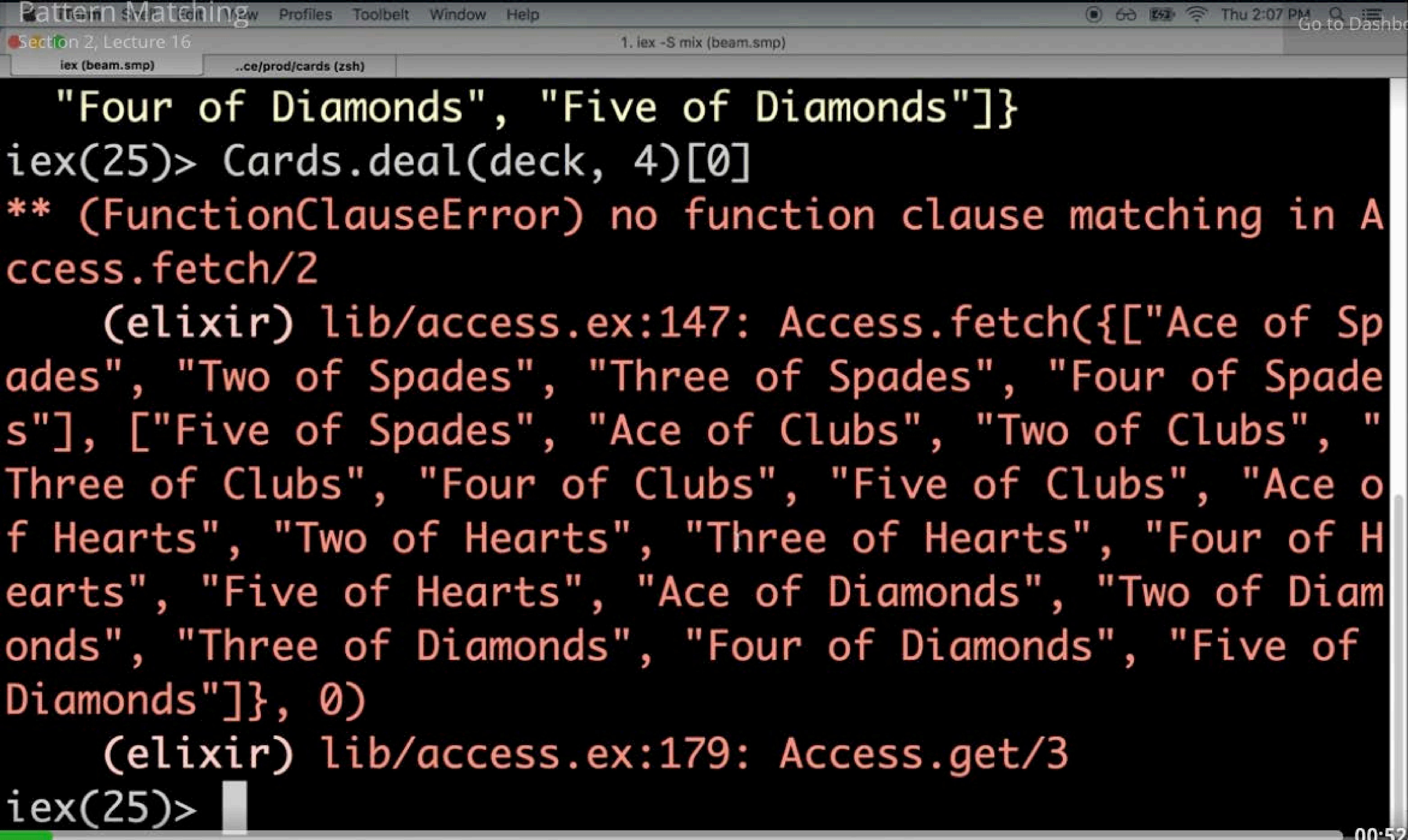
```
  def contains?(deck, card) do
    Enum.member?(deck, card)
  end

  def deal(deck, hand_size) do
    Enum.split(deck, hand_size)
  end
end

```



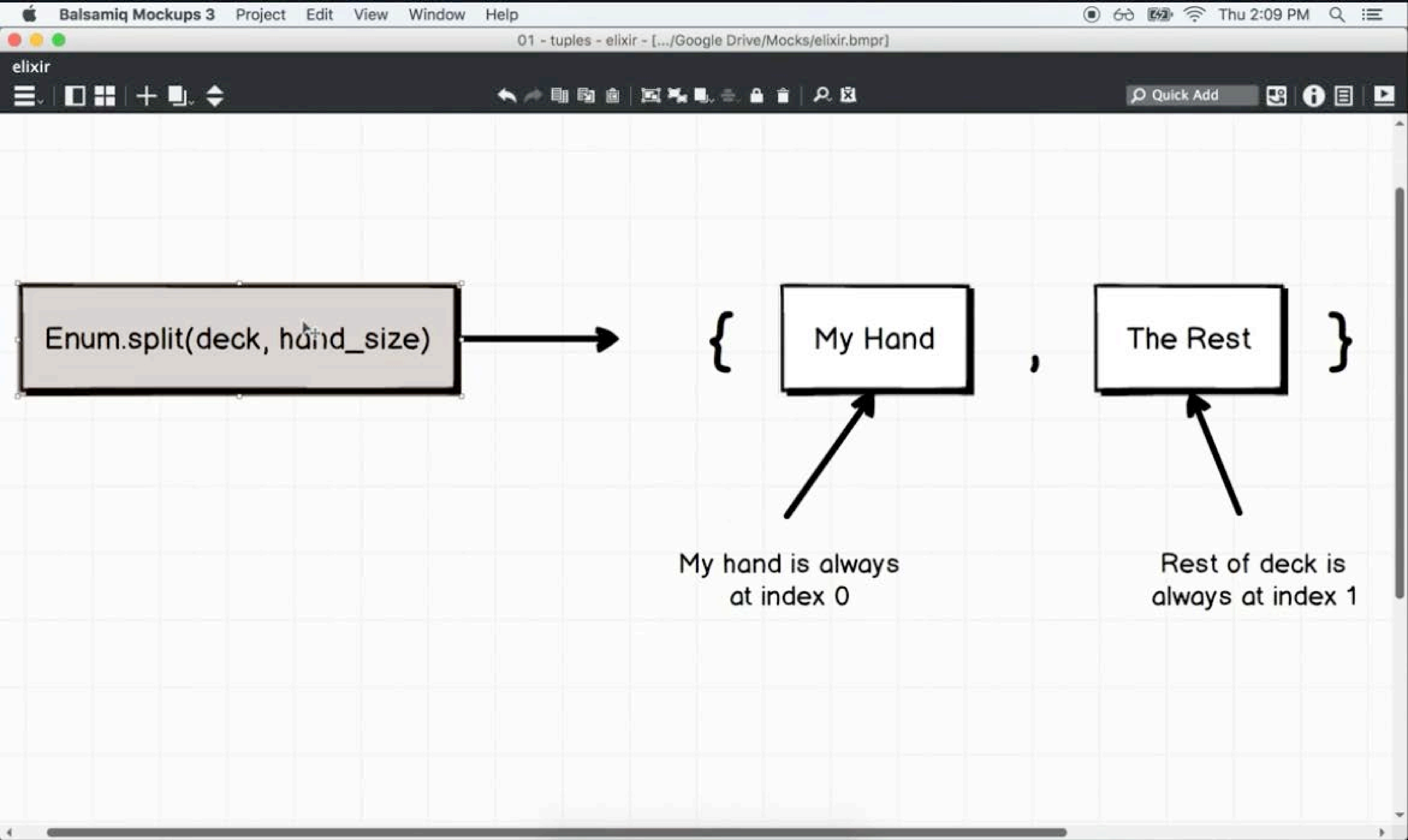
```
iex(23)> deck = Cards.create_deck
```

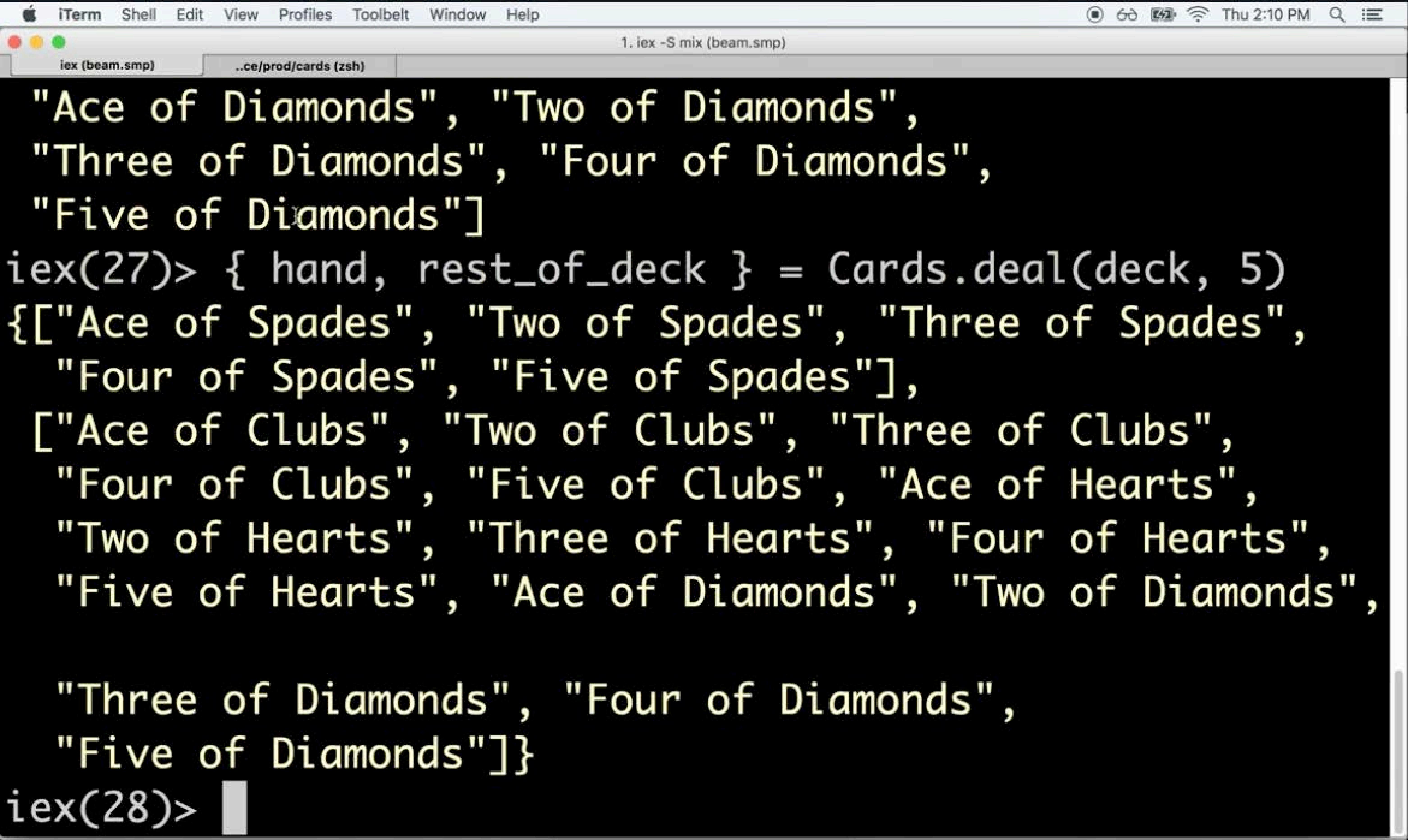


```
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds",
 "Three of Diamonds", "Four of Diamonds",
 "Five of Diamonds"]
iex(24)> Cards.deal(deck, 4)
{["Ace of Spades", "Two of Spades", "Three of Spades",
  "Four of Spades"],
 ["Five of Spades", "Ace of Clubs", "Two of Clubs",
  "Three of Clubs", "Four of Clubs", "Five of Clubs",
  "Ace of Hearts", "Two of Hearts", "Three of Hearts",
  "Four of Hearts", "Five of Hearts", "Ace of Diamonds",
  "Two of Diamonds", "Three of Diamonds",
  "Four of Diamonds", "Five of Diamonds"]}
iex(25)> 
```

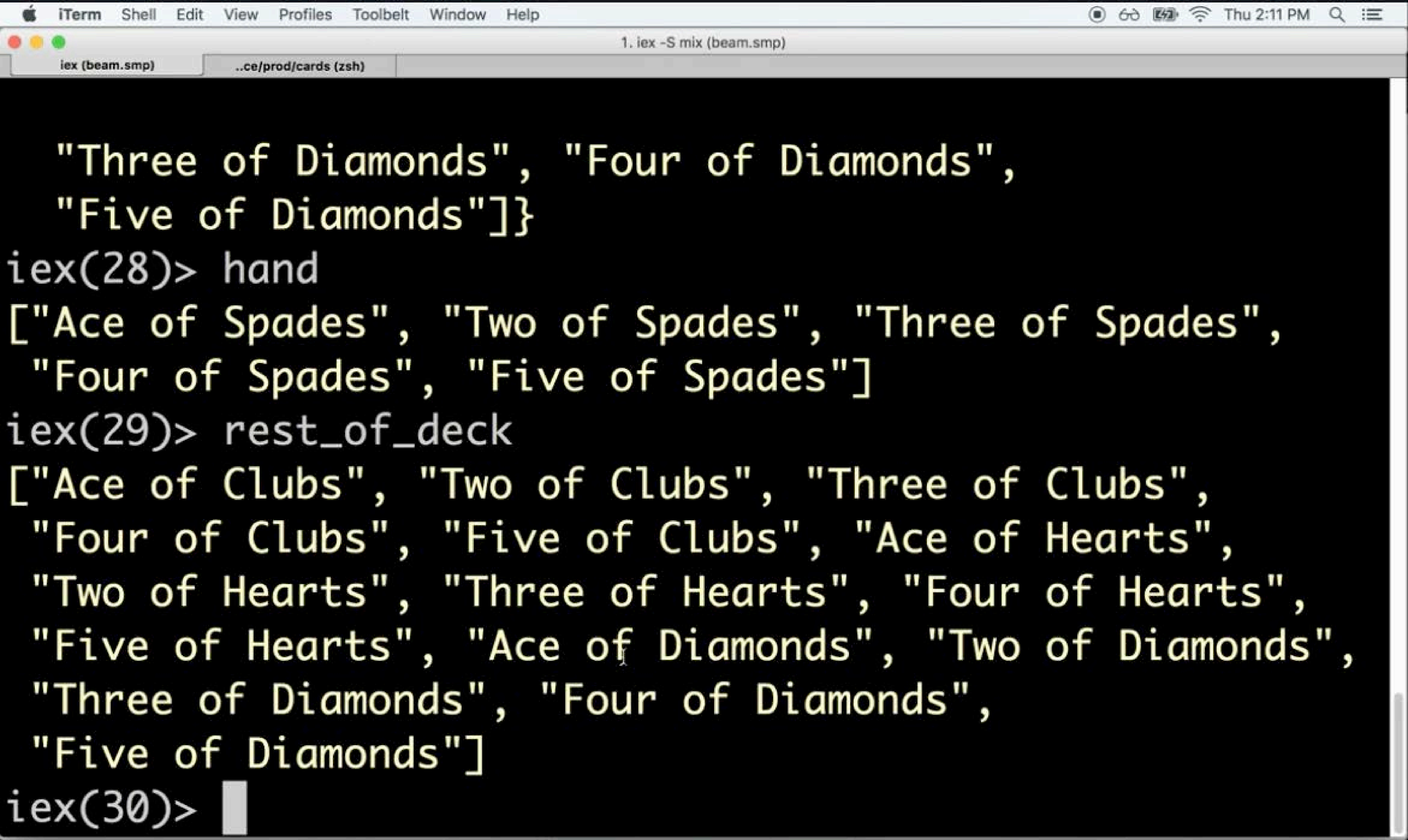


```
  "Four of Diamonds", "Five of Diamonds"]}
iex(25)> Cards.deal(deck, 4)[0]
** (FunctionClauseError) no function clause matching in A
ccess.fetch/2
    (elixir) lib/access.ex:147: Access.fetch({["Ace of Sp
ades", "Two of Spades", "Three of Spades", "Four of Spade
s"], ["Five of Spades", "Ace of Clubs", "Two of Clubs", "
Three of Clubs", "Four of Clubs", "Five of Clubs", "Ace o
f Hearts", "Two of Hearts", "Three of Hearts", "Four of H
earts", "Five of Hearts", "Ace of Diamonds", "Two of Diam
onds", "Three of Diamonds", "Four of Diamonds", "Five of 
Diamonds"]}, 0)
    (elixir) lib/access.ex:179: Access.get/3
iex(25)> 
```

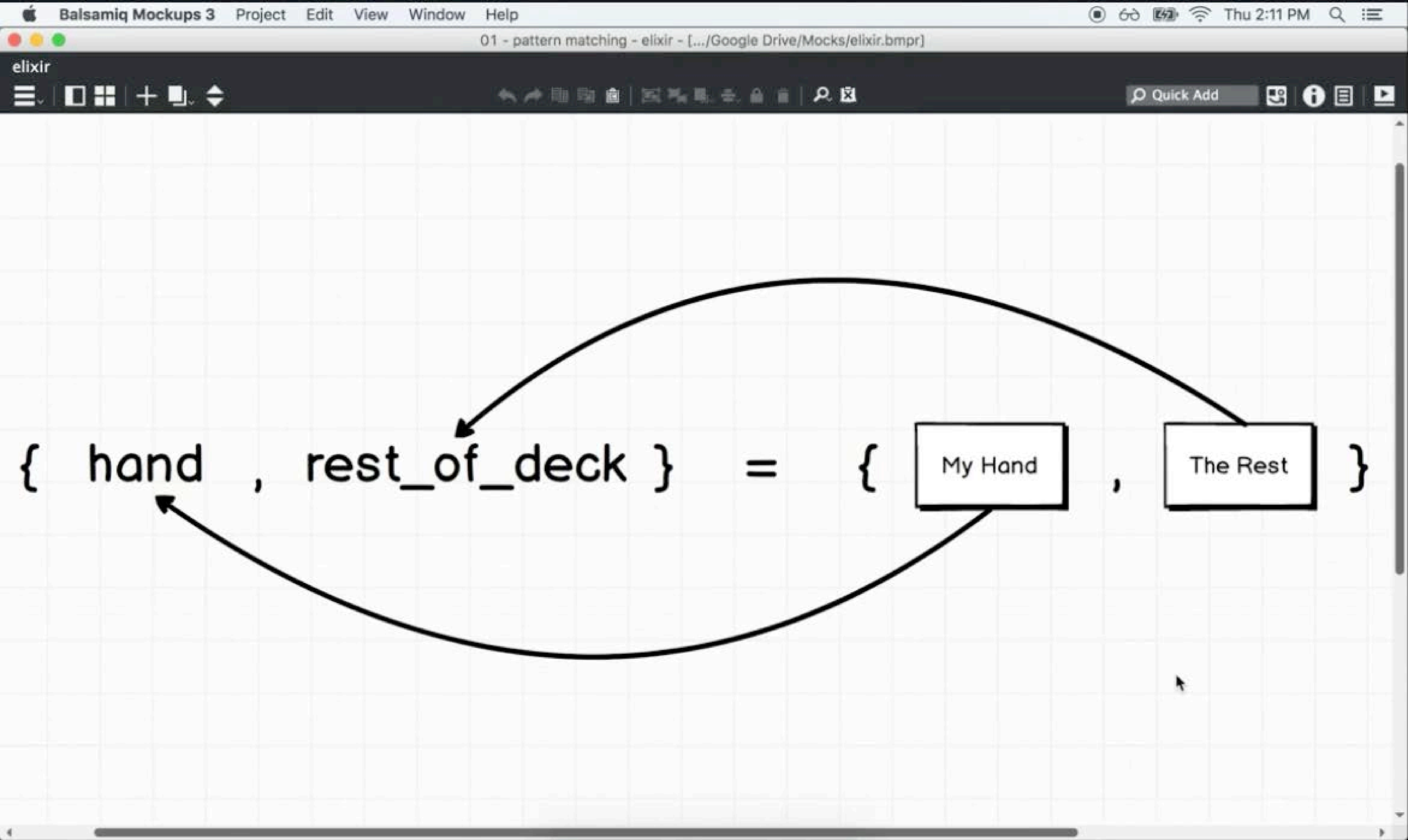

Balsamiq Mockups 3
Project
Edit
View
Window
Help
Thu 2:09 PM
01 - tuples - elixir - [.../Google Drive/Mocks/elixir.bmpr]
elixir
Quick Add
Enum.split(deck, hand_size)
{
My Hand
,
The Rest
}
My hand is always at index 0
Rest of deck is always at index 1



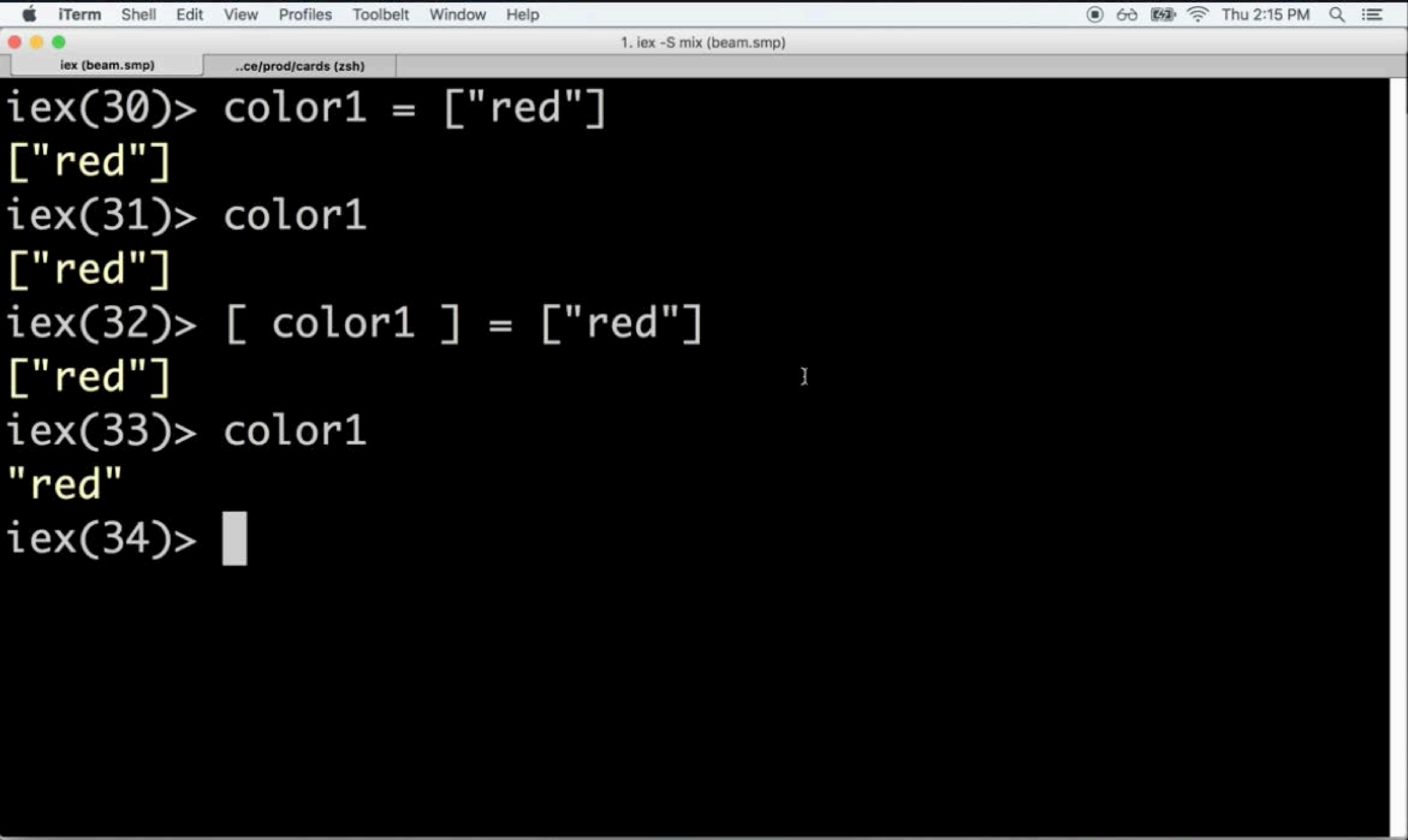
```
 "Ace of Diamonds", "Two of Diamonds",
 "Three of Diamonds", "Four of Diamonds",
 "Five of Diamonds"]
iex(27)> { hand, rest_of_deck } = Cards.deal(deck, 5)
{["Ace of Spades", "Two of Spades", "Three of Spades",
  "Four of Spades", "Five of Spades"],
 ["Ace of Clubs", "Two of Clubs", "Three of Clubs",
  "Four of Clubs", "Five of Clubs", "Ace of Hearts",
  "Two of Hearts", "Three of Hearts", "Four of Hearts",
  "Five of Hearts", "Ace of Diamonds", "Two of Diamonds",

  "Three of Diamonds", "Four of Diamonds",
  "Five of Diamonds"]}
iex(28)> 
```



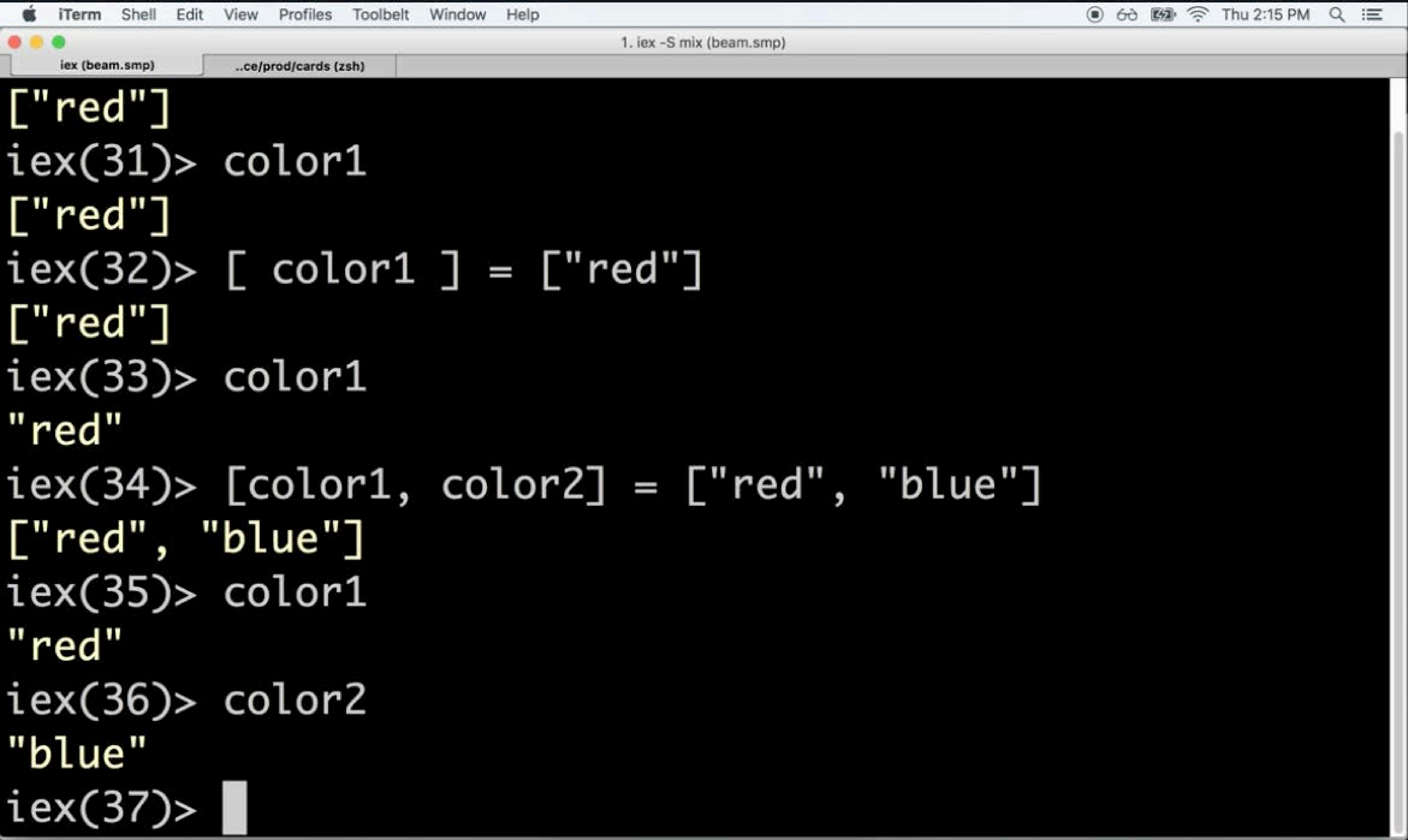
```
  "Three of Diamonds", "Four of Diamonds",
  "Five of Diamonds"]}
iex(28)> hand
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades"]
iex(29)> rest_of_deck
["Ace of Clubs", "Two of Clubs", "Three of Clubs",
 "Four of Clubs", "Five of Clubs", "Ace of Hearts",
 "Two of Hearts", "Three of Hearts", "Four of Hearts",
 "Five of Hearts", "Ace of Diamonds", "Two of Diamonds",
 "Three of Diamonds", "Four of Diamonds",
 "Five of Diamonds"]
iex(30)>
```



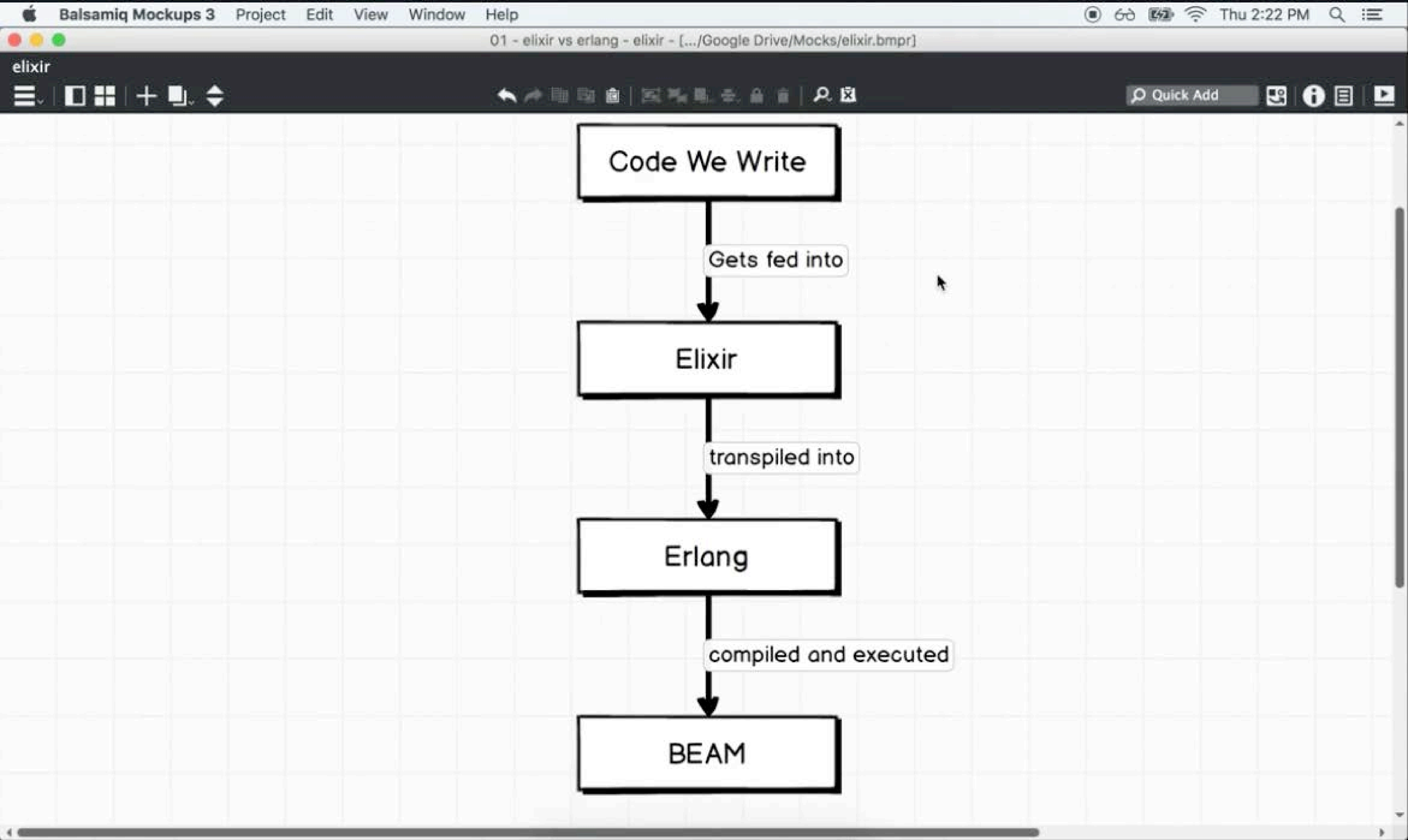
```
{ hand , rest_of_deck } = {     ,     }
```

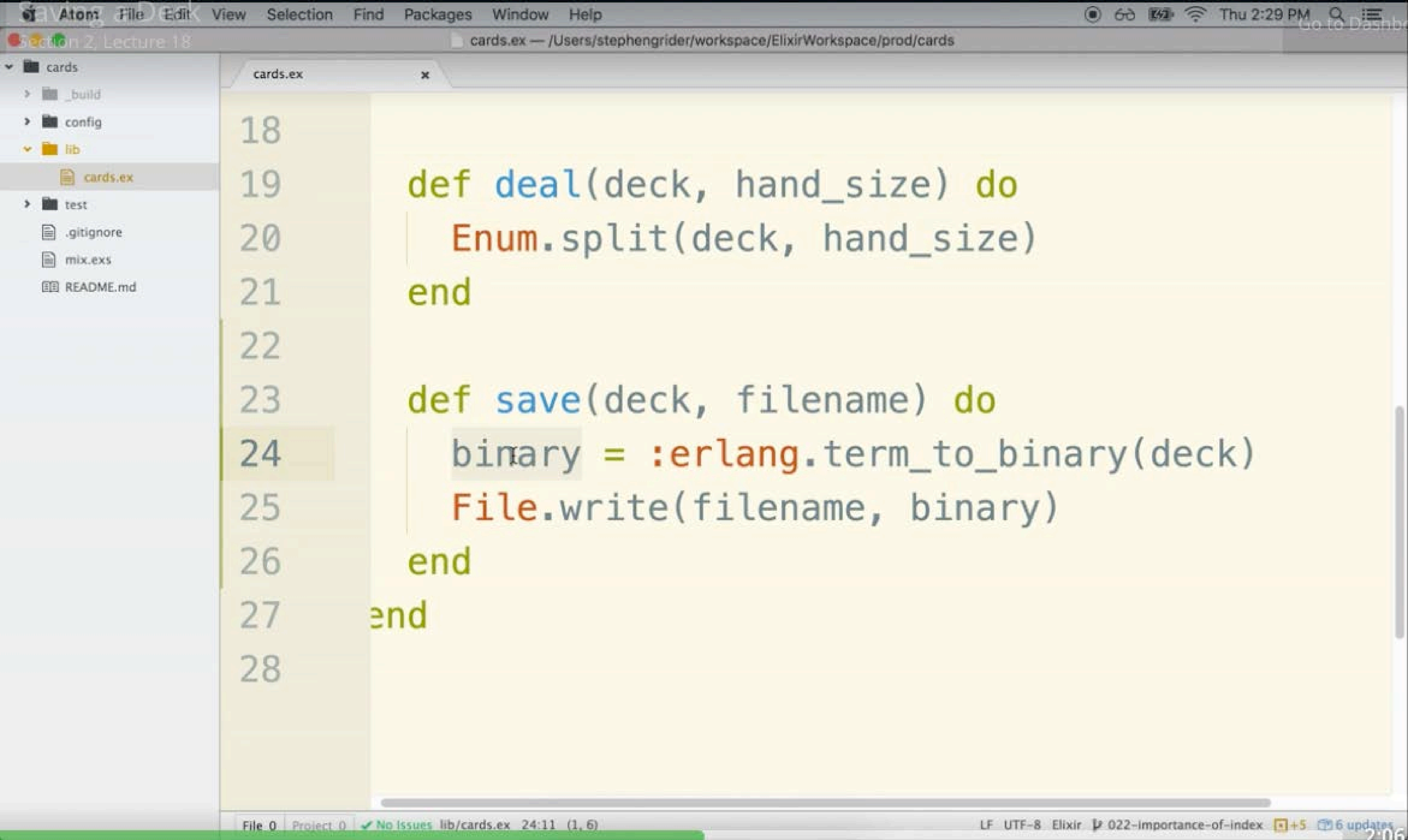


```
iex(30)> color1 = ["red"]
["red"]
iex(31)> color1
["red"]
iex(32)> [ color1 ] = ["red"]
["red"]
iex(33)> color1
"red"
iex(34)>
```

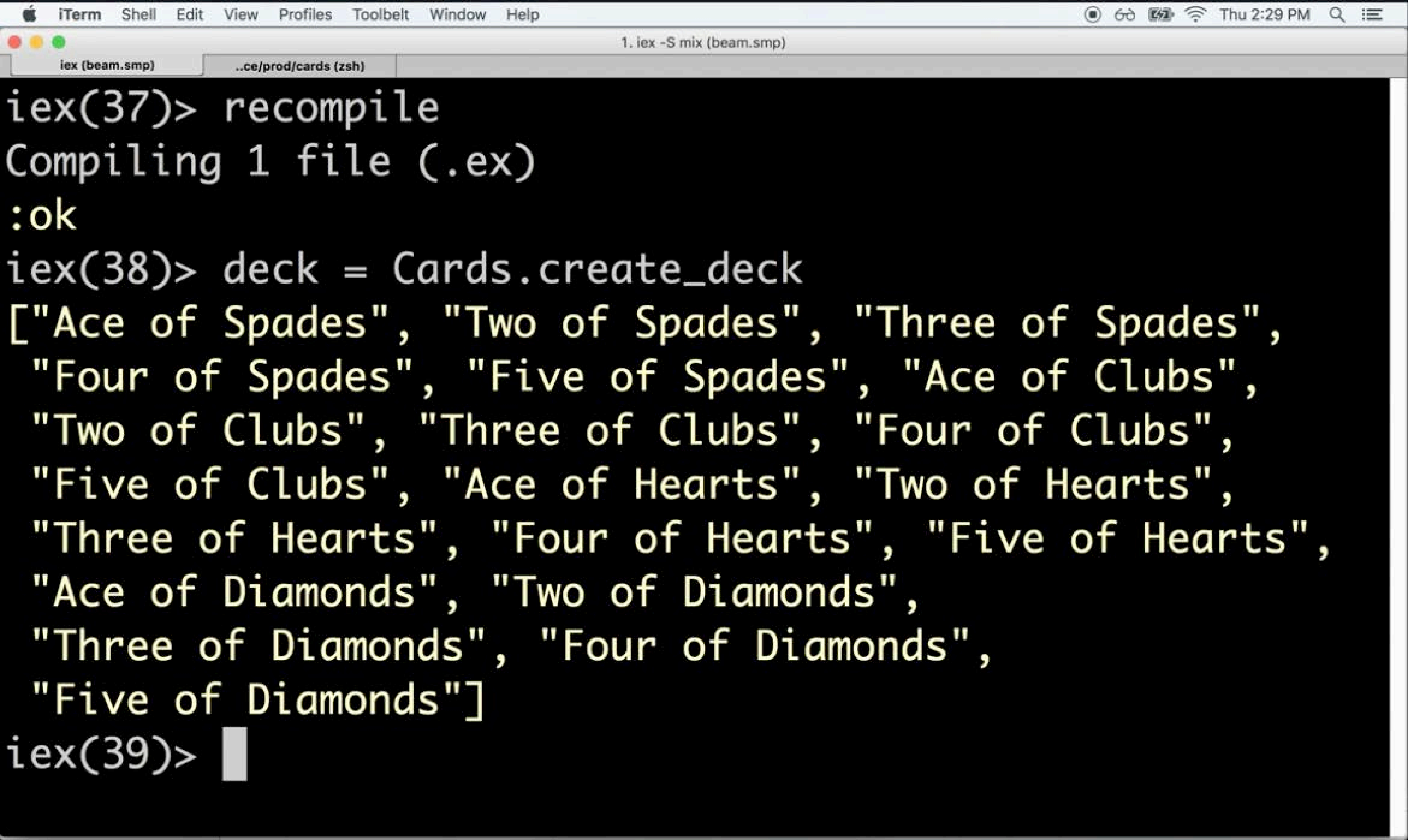


```
["red"]
iex(31)> color1
["red"]
iex(32)> [ color1 ] = ["red"]
["red"]
iex(33)> color1
"red"
iex(34)> [color1, color2] = ["red", "blue"]
["red", "blue"]
iex(35)> color1
"red"
iex(36)> color2
"blue"
iex(37)>
```

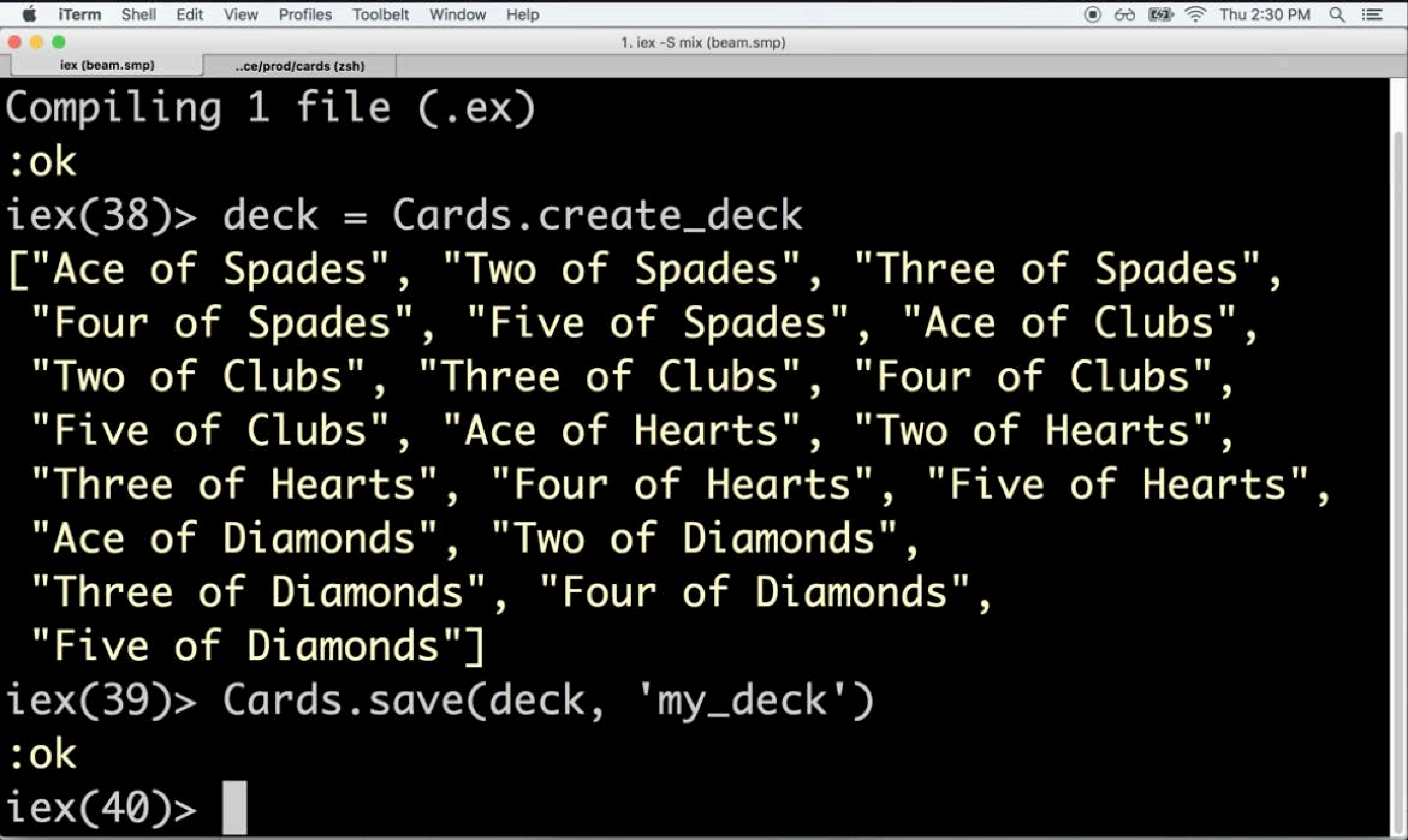

Balsamiq Mockups 3
Project
Edit
View
Window
Help
Thu 2:22 PM
01 - elixir vs erlang - elixir - [.../Google Drive/Mocks/elixir.bmpr]
elixir
Quick Add
Code We Write
Gets fed into
Elixir
transpiled into
Erlang
compiled and executed
BEAM


Atom File Edit View Selection Find Packages Window Help
Thu 2:29 PM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
lib
cards.ex
test
.gitignore
mix.exs
README.md
File 0 Project 0 No Issues lib/cards.ex 24:11 (1, 6)
LF UTF-8 Elixir 022-importance-of-index +5 6 updates

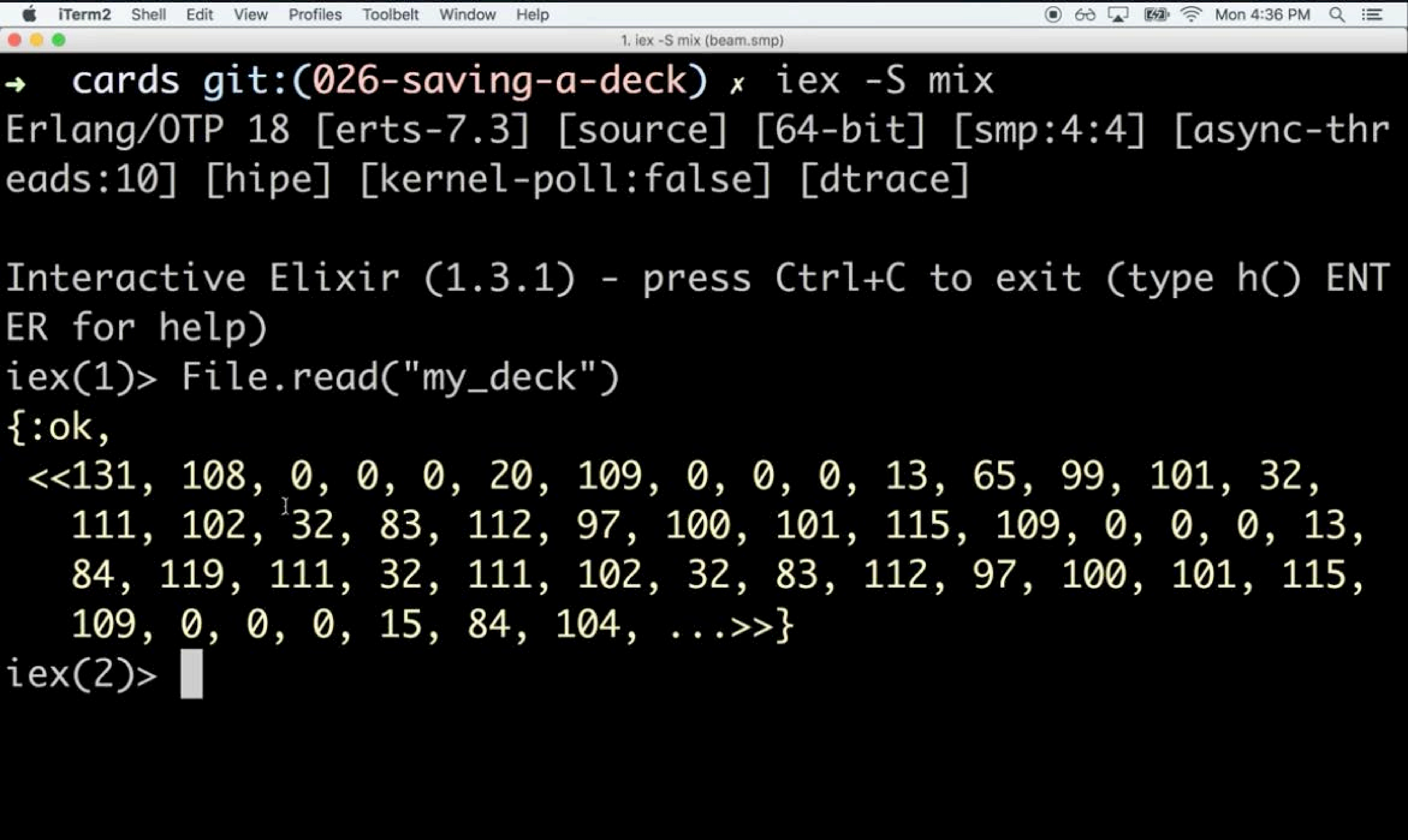
```
  def deal(deck, hand_size) do
    Enum.split(deck, hand_size)
  end

  def save(deck, filename) do
    binary = :erlang.term_to_binary(deck)
    File.write(filename, binary)
  end
end
```



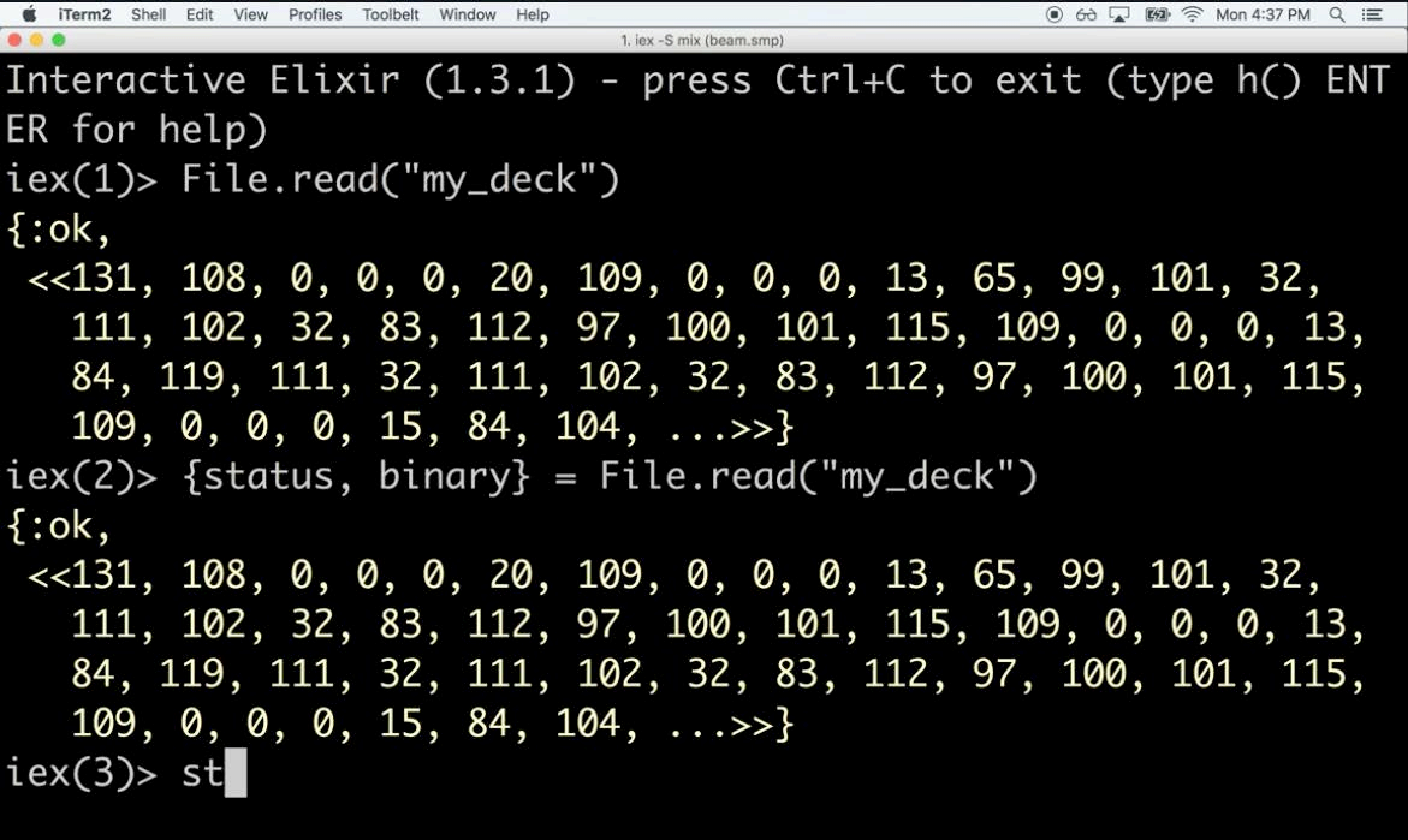
```
iex(37)> recompile
Compiling 1 file (.ex)
:ok
iex(38)> deck = Cards.create_deck
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades", "Ace of Clubs",
 "Two of Clubs", "Three of Clubs", "Four of Clubs",
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds",
 "Three of Diamonds", "Four of Diamonds",
 "Five of Diamonds"]
iex(39)> 
```



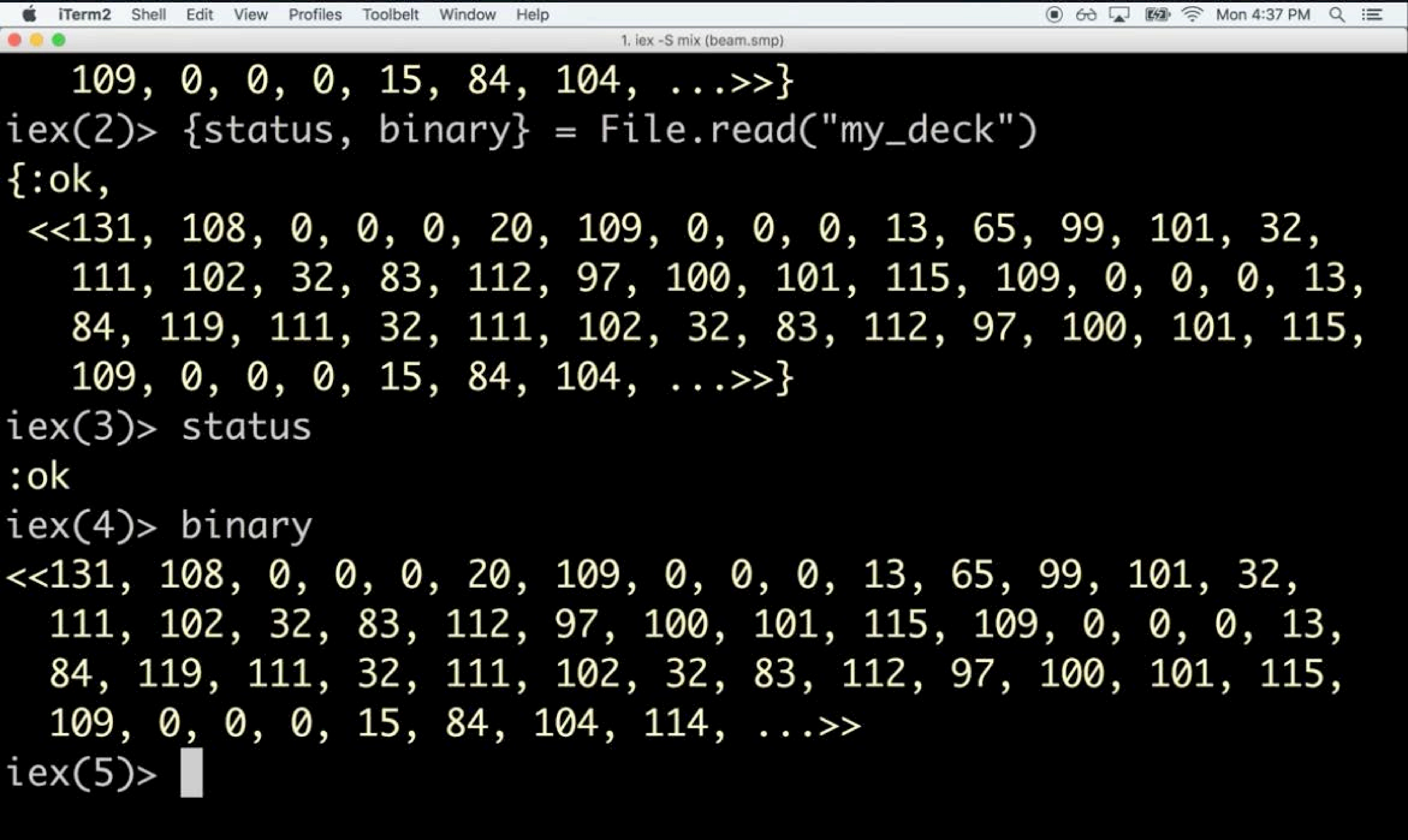
```
Compiling 1 file (.ex)
:ok
iex(38)> deck = Cards.create_deck
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades", "Ace of Clubs",
 "Two of Clubs", "Three of Clubs", "Four of Clubs",
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds",
 "Three of Diamonds", "Four of Diamonds",
 "Five of Diamonds"]
iex(39)> Cards.save(deck, 'my_deck')
:ok
iex(40)> 
```



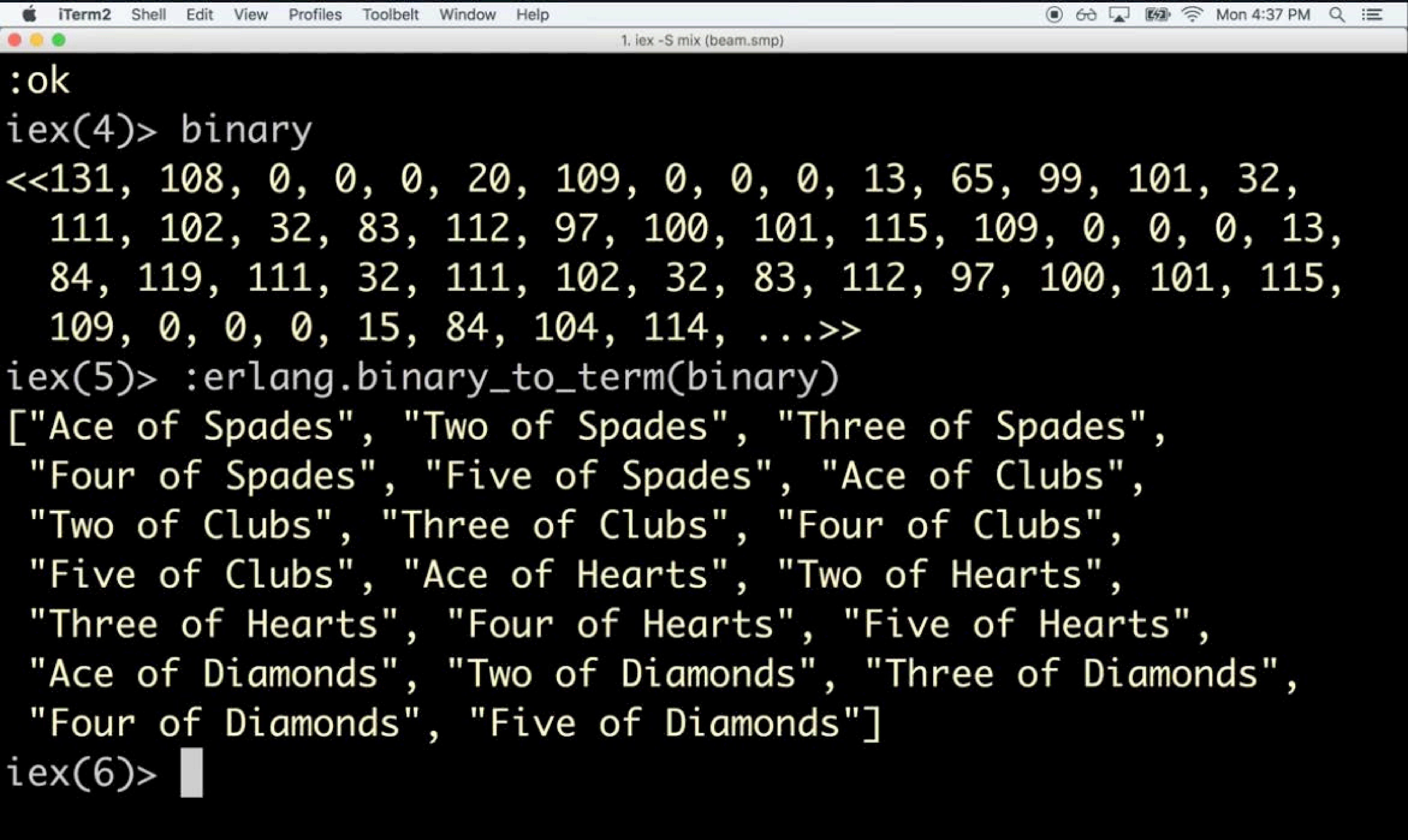
```
→  cards git:(026-saving-a-deck) ✗ iex -S mix
Erlang/OTP 18 [erts-7.3] [source] [64-bit] [smp:4:4] [async-thr
eads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.3.1) - press Ctrl+C to exit (type h() ENT
ER for help)
iex(1)> File.read("my_deck")
{:ok,
 <<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
   111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
   84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
   109, 0, 0, 0, 15, 84, 104, ...>>}
iex(2)> 
```



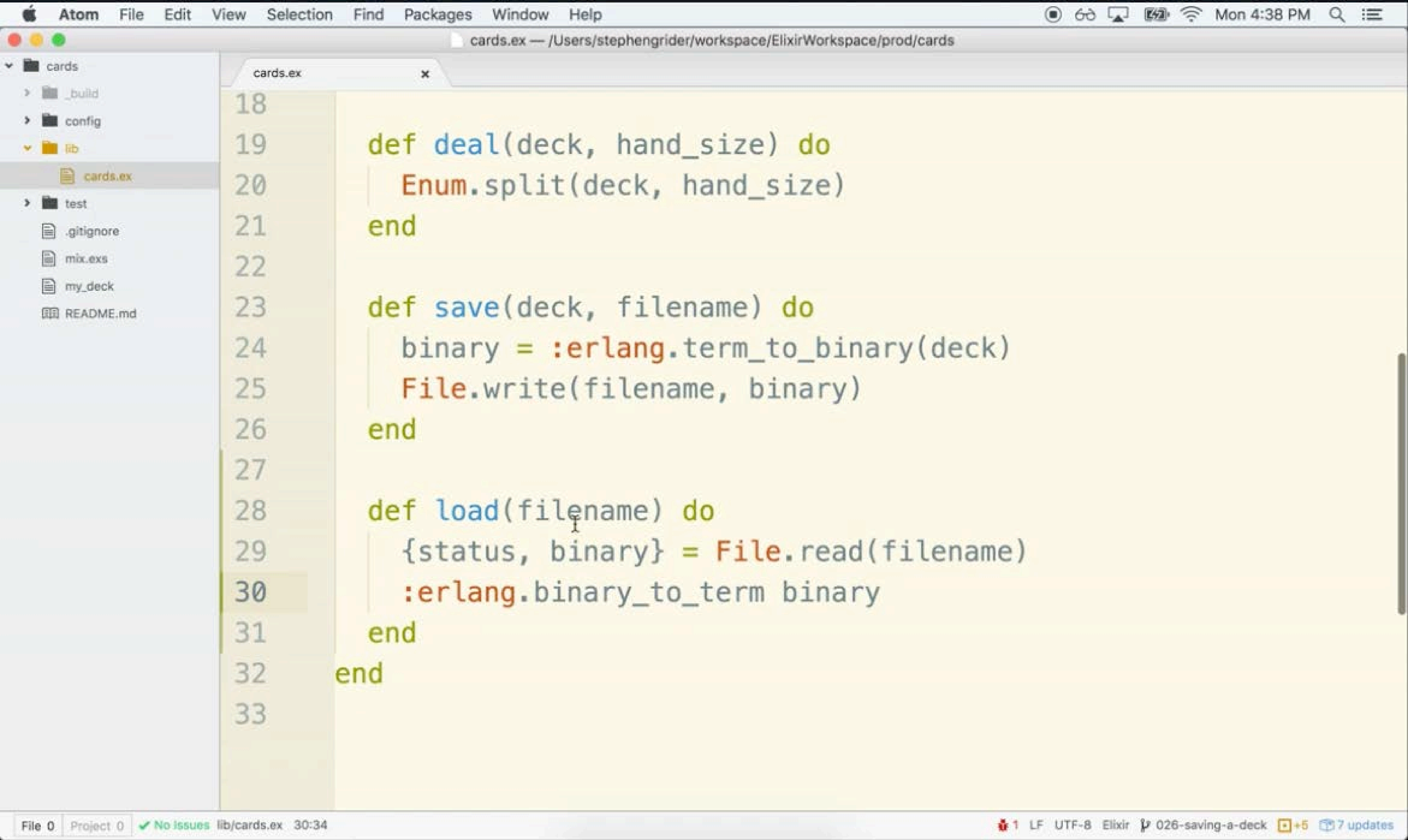
```
Interactive Elixir (1.3.1) - press Ctrl+C to exit (type h() ENT
ER for help)
iex(1)> File.read("my_deck")
{:ok,
 <<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
   111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
   84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
   109, 0, 0, 0, 15, 84, 104, ...>>}
iex(2)> {status, binary} = File.read("my_deck")
{:ok,
 <<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
   111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
   84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
   109, 0, 0, 0, 15, 84, 104, ...>>}
iex(3)> st
```



```
   109, 0, 0, 0, 15, 84, 104, ...>>}
iex(2)> {status, binary} = File.read("my_deck")
{:ok,
 <<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
   111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
   84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
   109, 0, 0, 0, 15, 84, 104, ...>>}
iex(3)> status
:ok
iex(4)> binary
<<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
  111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
  84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
  109, 0, 0, 0, 15, 84, 104, 114, ...>>
iex(5)> 
```

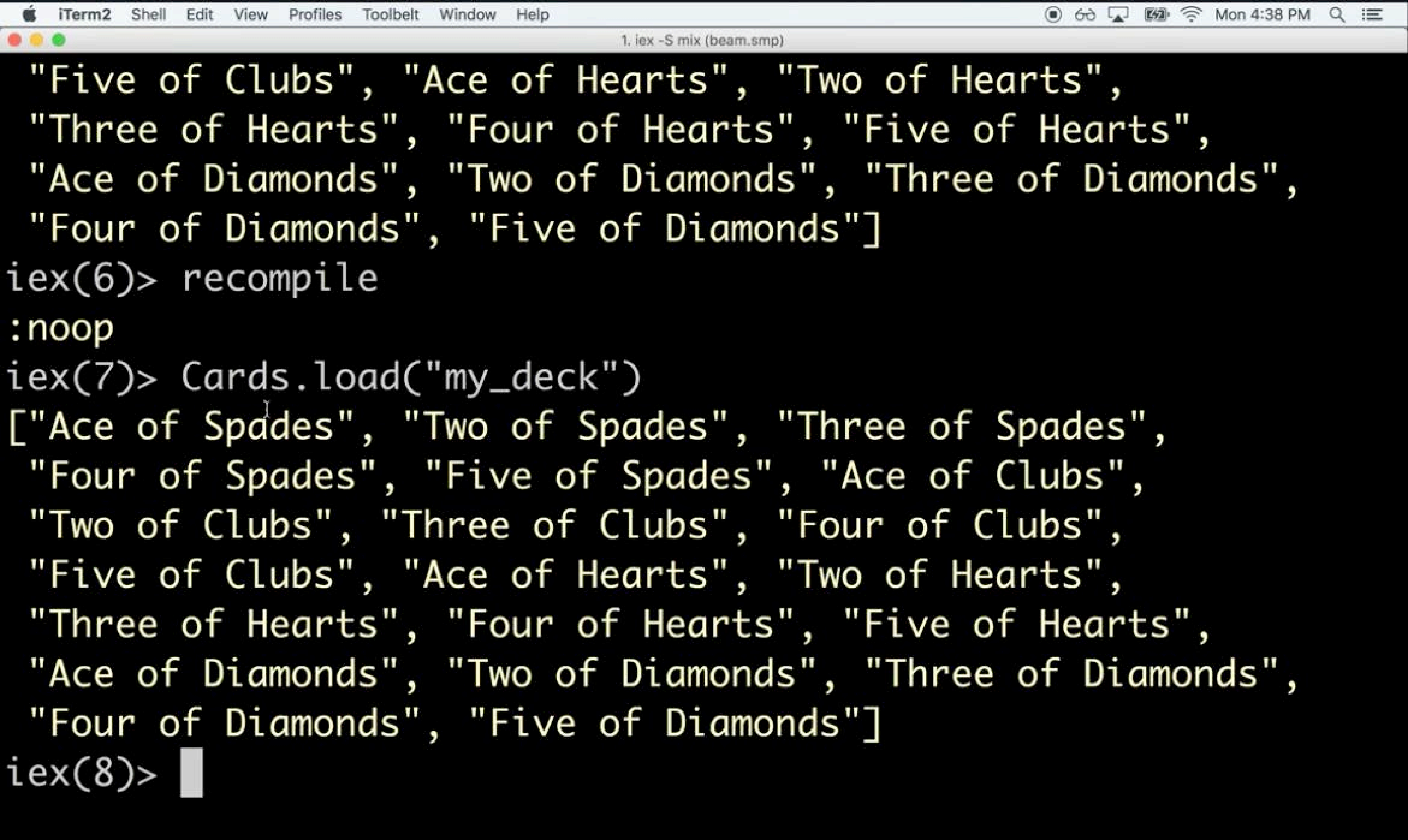


```
:ok
iex(4)> binary
<<131, 108, 0, 0, 0, 20, 109, 0, 0, 0, 13, 65, 99, 101, 32,
  111, 102, 32, 83, 112, 97, 100, 101, 115, 109, 0, 0, 0, 13,
  84, 119, 111, 32, 111, 102, 32, 83, 112, 97, 100, 101, 115,
  109, 0, 0, 0, 15, 84, 104, 114, ...>>
iex(5)> :erlang.binary_to_term(binary)
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades", "Ace of Clubs",
 "Two of Clubs", "Three of Clubs", "Four of Clubs",
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds", "Three of Diamonds",
 "Four of Diamonds", "Five of Diamonds"]
iex(6)> 
```

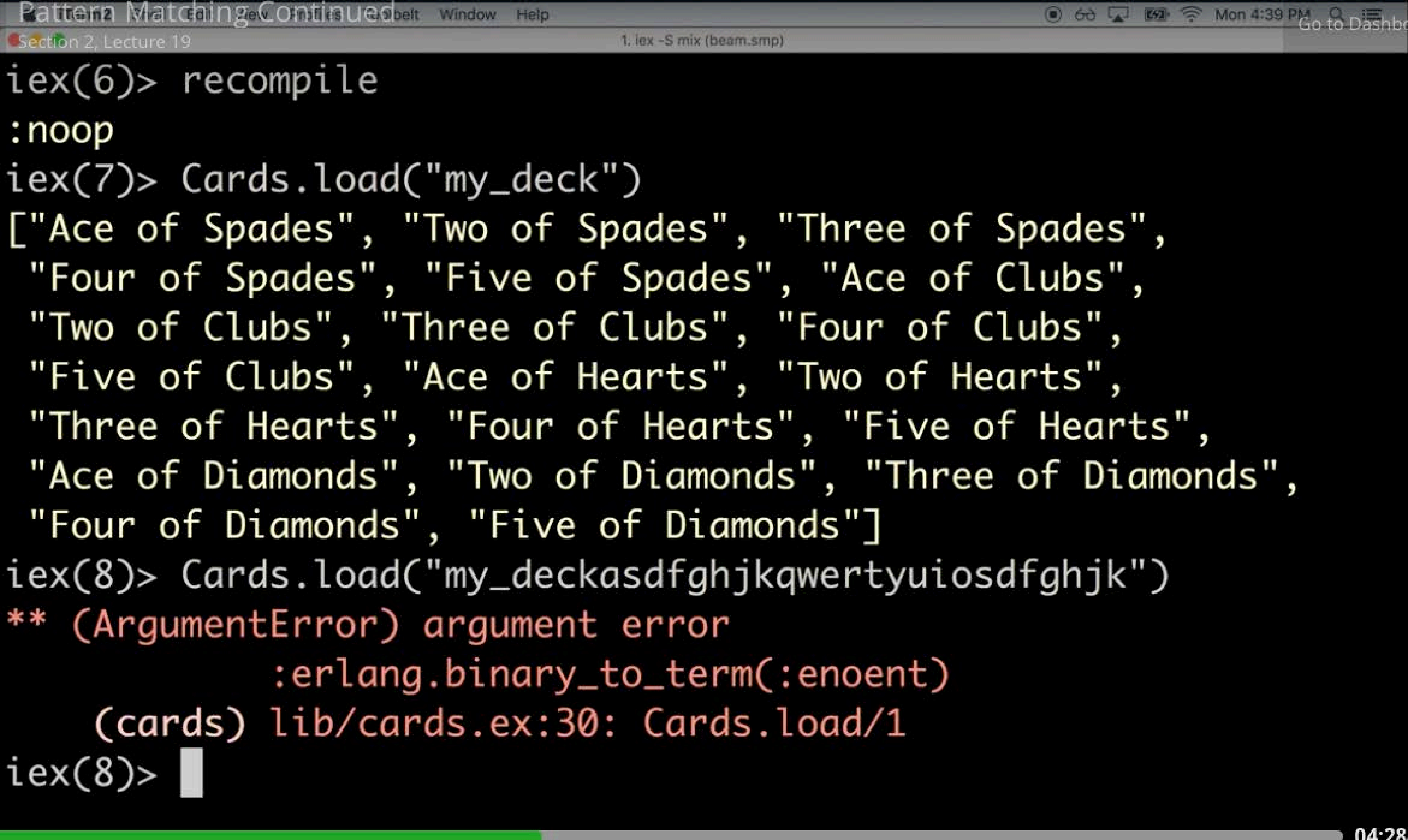


Atom File Edit View Selection Find Packages Window Help
Mon 4:38 PM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
lib
cards.ex
test
.gitignore
mix.exs
my_deck
README.md
File 0 Project 0 No Issues lib/cards.ex 30:34
1 LF UTF-8 Elixir 026-saving-a-deck +5 7 updates

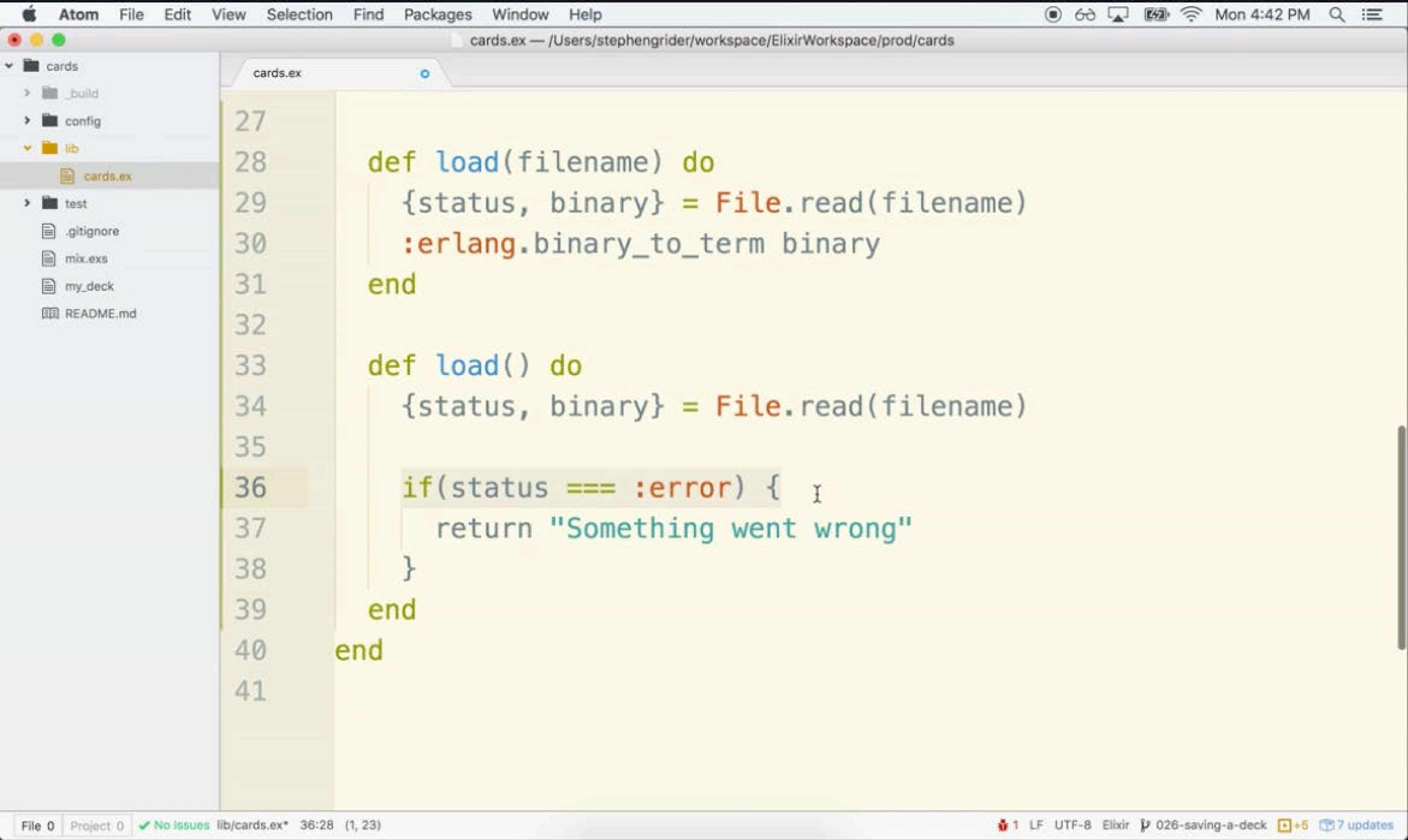
```elixir

  def deal(deck, hand_size) do
    Enum.split(deck, hand_size)
  end

  def save(deck, filename) do
    binary = :erlang.term_to_binary(deck)
    File.write(filename, binary)
  end

  def load(filename) do
    {status, binary} = File.read(filename)
    :erlang.binary_to_term binary
  end
end

```



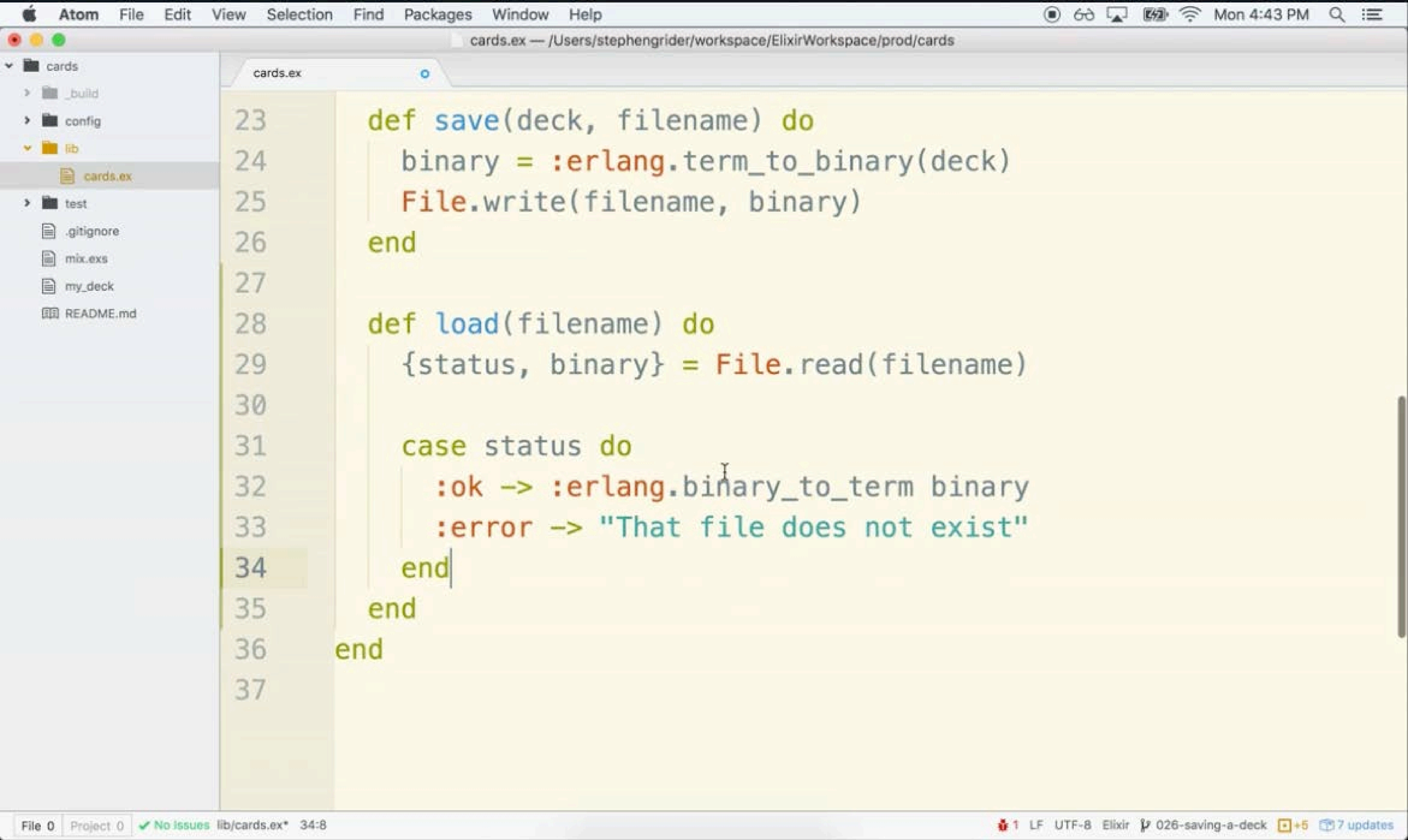
```
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds", "Three of Diamonds",
 "Four of Diamonds", "Five of Diamonds"]
iex(6)> recompile
:noop
iex(7)> Cards.load("my_deck")
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades", "Ace of Clubs",
 "Two of Clubs", "Three of Clubs", "Four of Clubs",
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds", "Three of Diamonds",
 "Four of Diamonds", "Five of Diamonds"]
iex(8)> 
```


Pattern Matching Continued
Section 2, Lecture 19
iTerm2 Shell Edit View Profiles Toolbelt Window Help
Mon 4:39 PM
Go to Dashbo
1. iex -S mix (beam.smp)
04:28

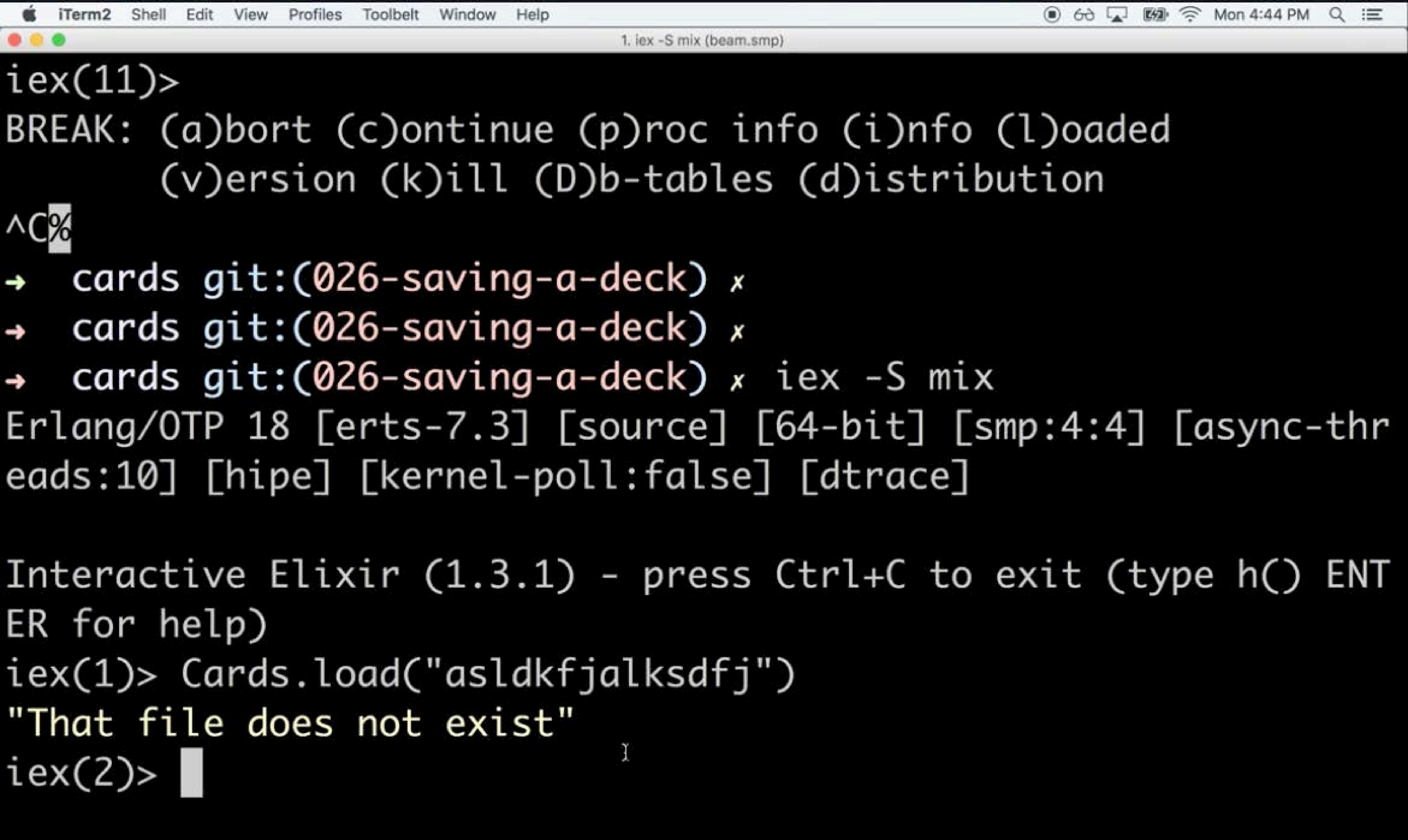
```
iex(6)> recompile
:noop
iex(7)> Cards.load("my_deck")
["Ace of Spades", "Two of Spades", "Three of Spades",
 "Four of Spades", "Five of Spades", "Ace of Clubs",
 "Two of Clubs", "Three of Clubs", "Four of Clubs",
 "Five of Clubs", "Ace of Hearts", "Two of Hearts",
 "Three of Hearts", "Four of Hearts", "Five of Hearts",
 "Ace of Diamonds", "Two of Diamonds", "Three of Diamonds",
 "Four of Diamonds", "Five of Diamonds"]
iex(8)> Cards.load("my_deckasdfghjkqwertyuiosdfghjk")
** (ArgumentError) argument error
            :erlang.binary_to_term(:enoent)
    (cards) lib/cards.ex:30: Cards.load/1
iex(8)>
```


Atom File Edit View Selection Find Packages Window Help
Mon 4:42 PM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
lib
cards.ex
test
.gitignore
mix.exs
my_deck
README.md
File 0 Project 0 No Issues lib/cards.ex* 36:28 (1, 23)
LF UTF-8 Elixir 026-saving-a-deck +5 7 updates

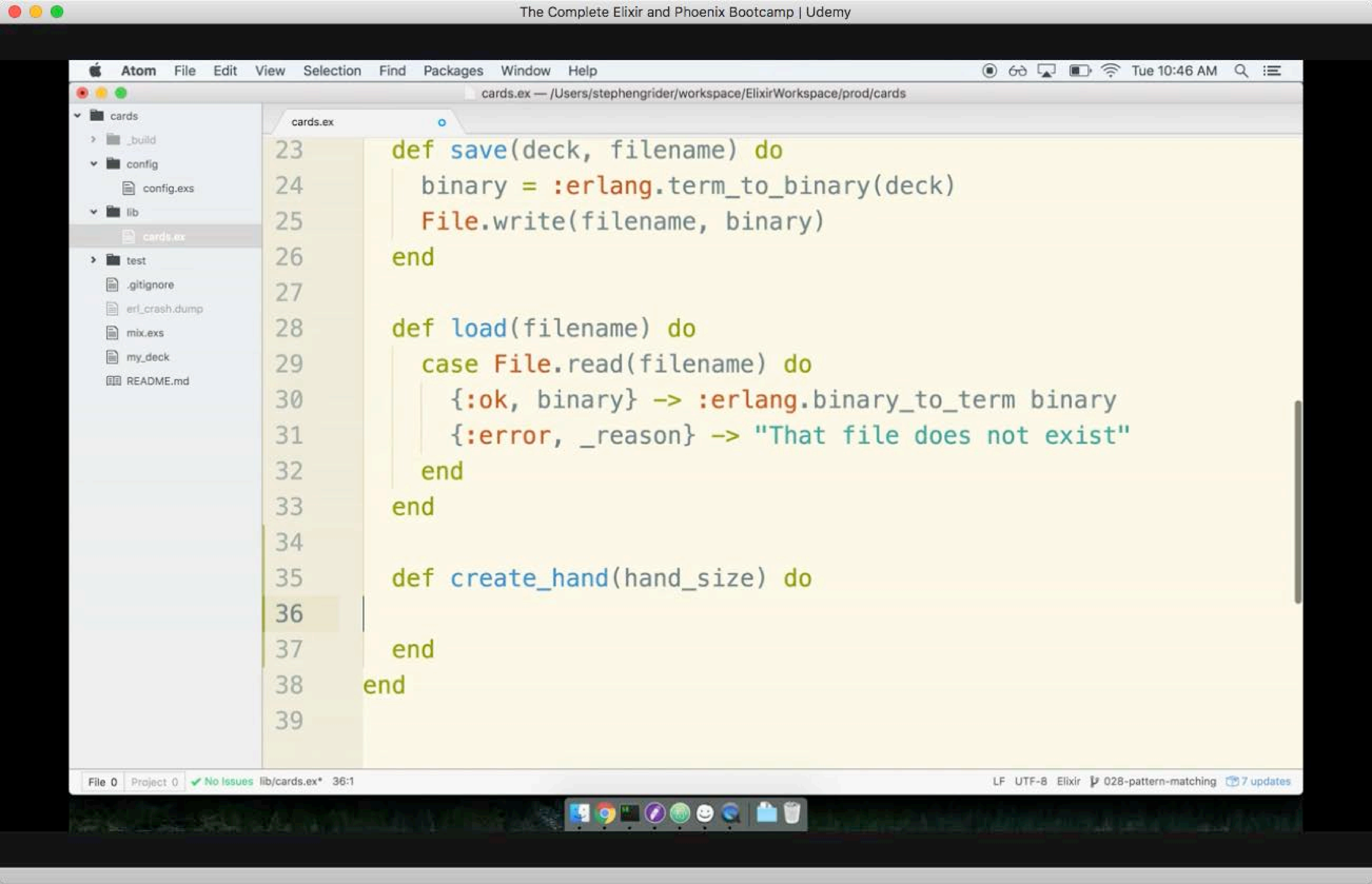
```

  def load(filename) do
    {status, binary} = File.read(filename)
    :erlang.binary_to_term binary
  end

  def load() do
    {status, binary} = File.read(filename)

    if(status === :error) {
      return "Something went wrong"
    }
  end
end

```


Atom File Edit View Selection Find Packages Window Help
Mon 4:43 PM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
lib
cards.ex
test
.gitignore
mix.exs
my_deck
README.md
cards.ex
File 0 Project 0 No Issues lib/cards.ex* 34:8
1 LF UTF-8 Elixir 026-saving-a-deck +5 7 updates

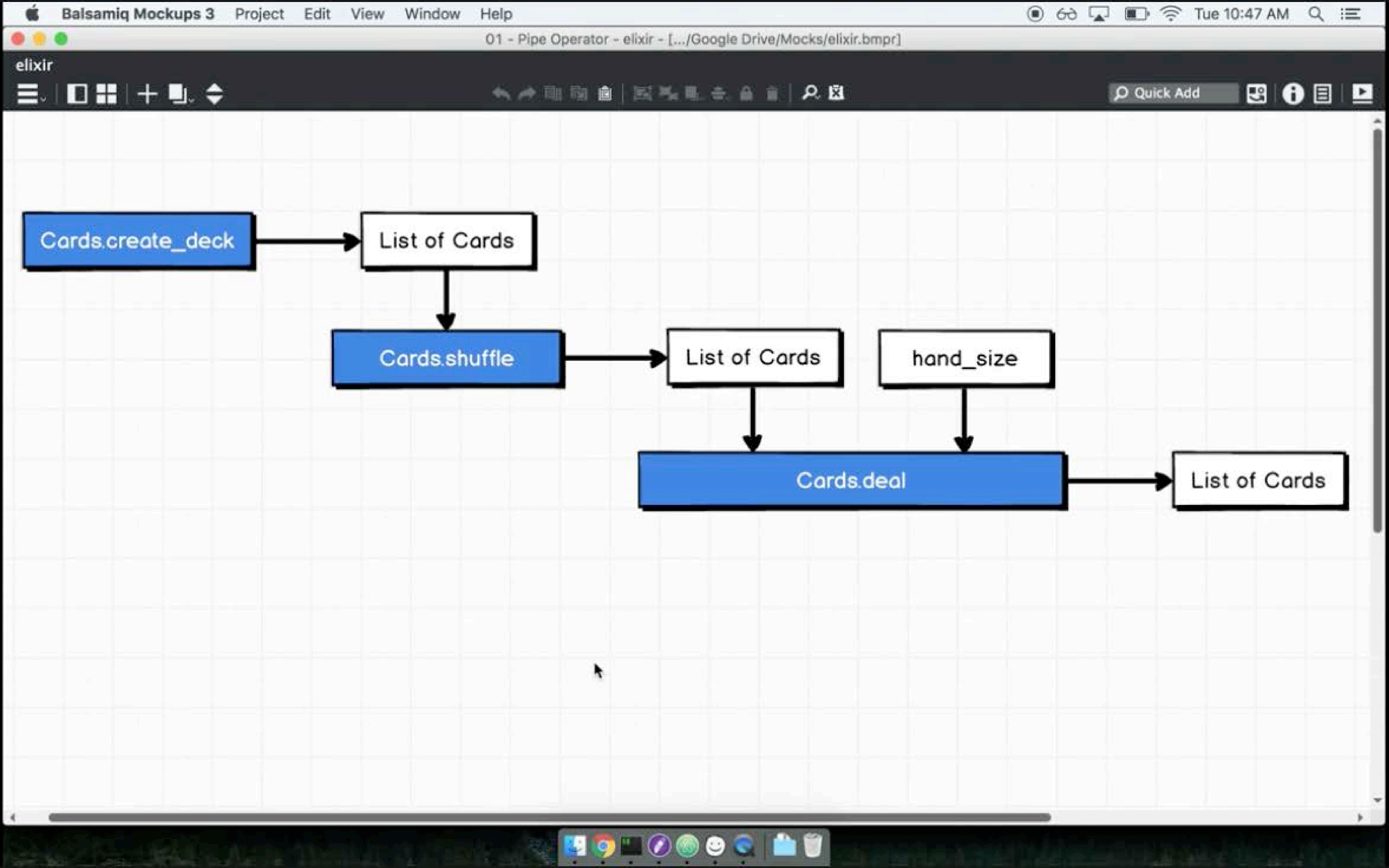
```
  def save(deck, filename) do
    binary = :erlang.term_to_binary(deck)
    File.write(filename, binary)
  end

  def load(filename) do
    {status, binary} = File.read(filename)

    case status do
      :ok -> :erlang.binary_to_term binary
      :error -> "That file does not exist"
    end
  end
end

```

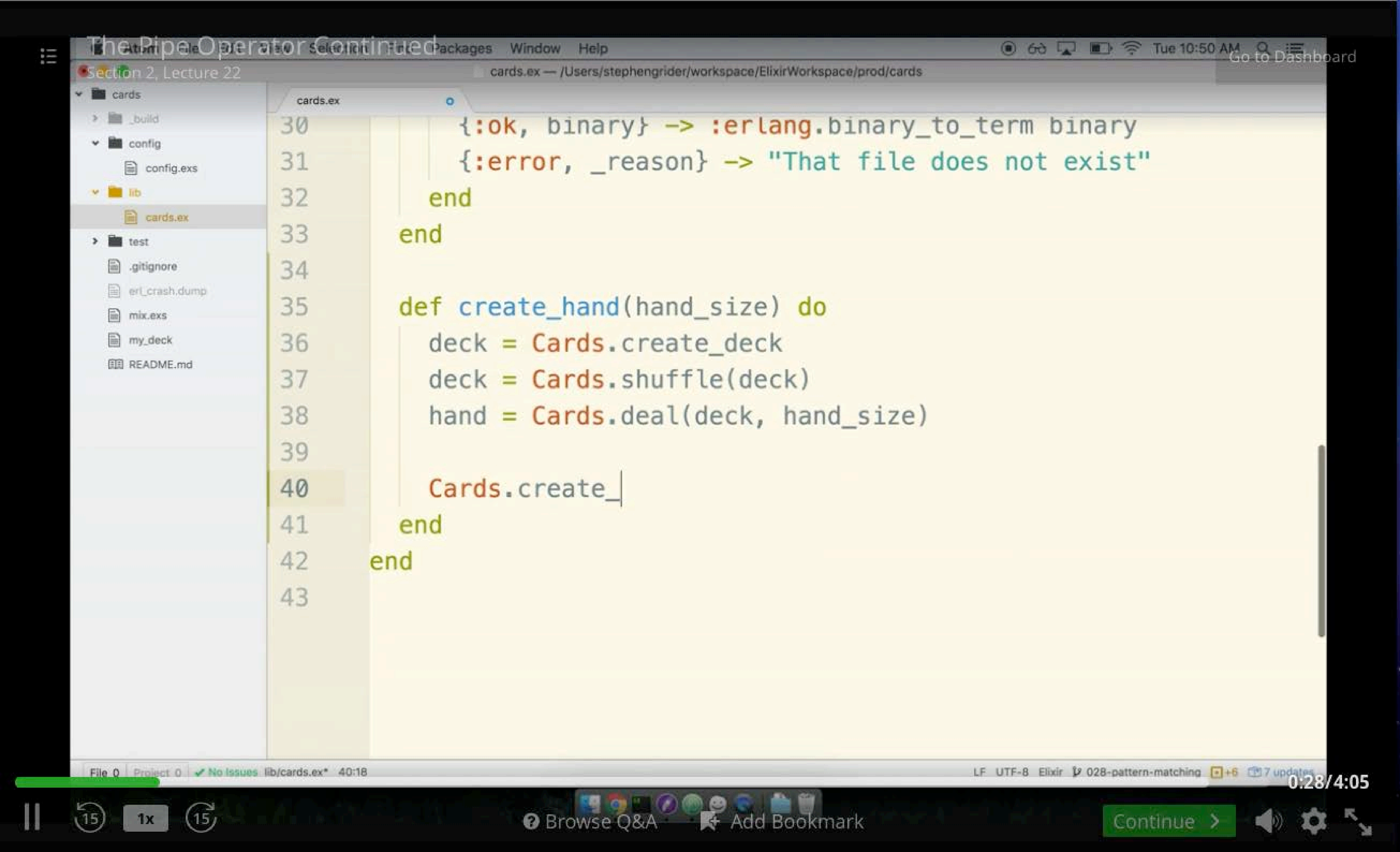


```
iex(11)>
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded
       (v)ersion (k)ill (D)b-tables (d)istribution
^C%
➜  cards git:(026-saving-a-deck) ✗
➜  cards git:(026-saving-a-deck) ✗
➜  cards git:(026-saving-a-deck) ✗ iex -S mix
Erlang/OTP 18 [erts-7.3] [source] [64-bit] [smp:4:4] [async-thr
eads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.3.1) - press Ctrl+C to exit (type h() ENT
ER for help)
iex(1)> Cards.load("asldkfjalksdfj")
"That file does not exist"
iex(2)>
```

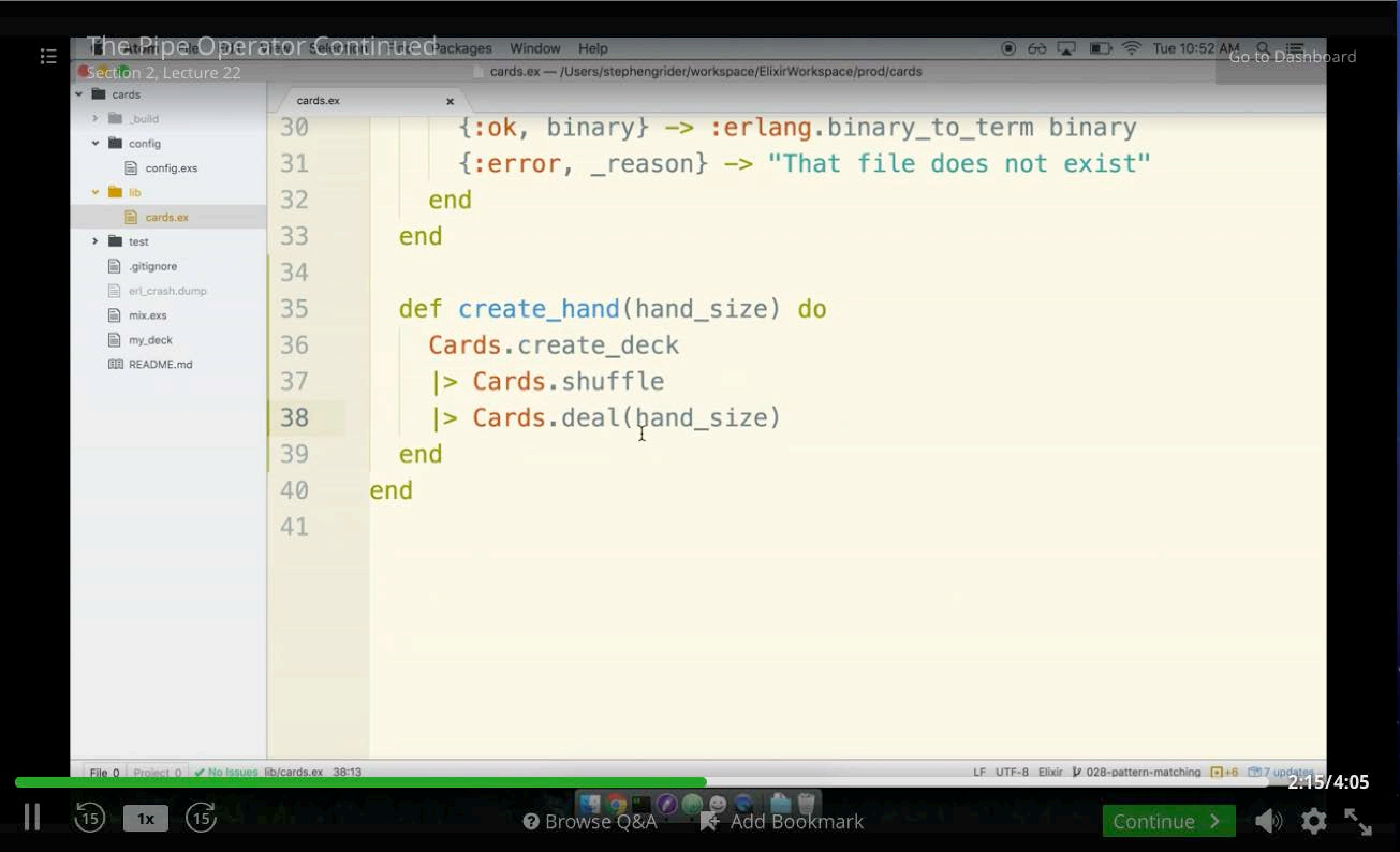

The Complete Elixir and Phoenix Bootcamp | Udemy
Atom File Edit View Selection Find Packages Window Help
Tue 10:46 AM
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
config.exs
lib
cards.ex
test
.gitignore
erl_crash.dump
mix.exs
my_deck
README.md
cards.ex
File 0 Project 0 No Issues lib/cards.ex* 36:1
LF UTF-8 Elixir 028-pattern-matching 7 updates

```elixir
  def save(deck, filename) do
    binary = :erlang.term_to_binary(deck)
    File.write(filename, binary)
  end

  def load(filename) do
    case File.read(filename) do
      {:ok, binary} -> :erlang.binary_to_term binary
      {:error, _reason} -> "That file does not exist"
    end
  end

  def create_hand(hand_size) do

  end
end
```

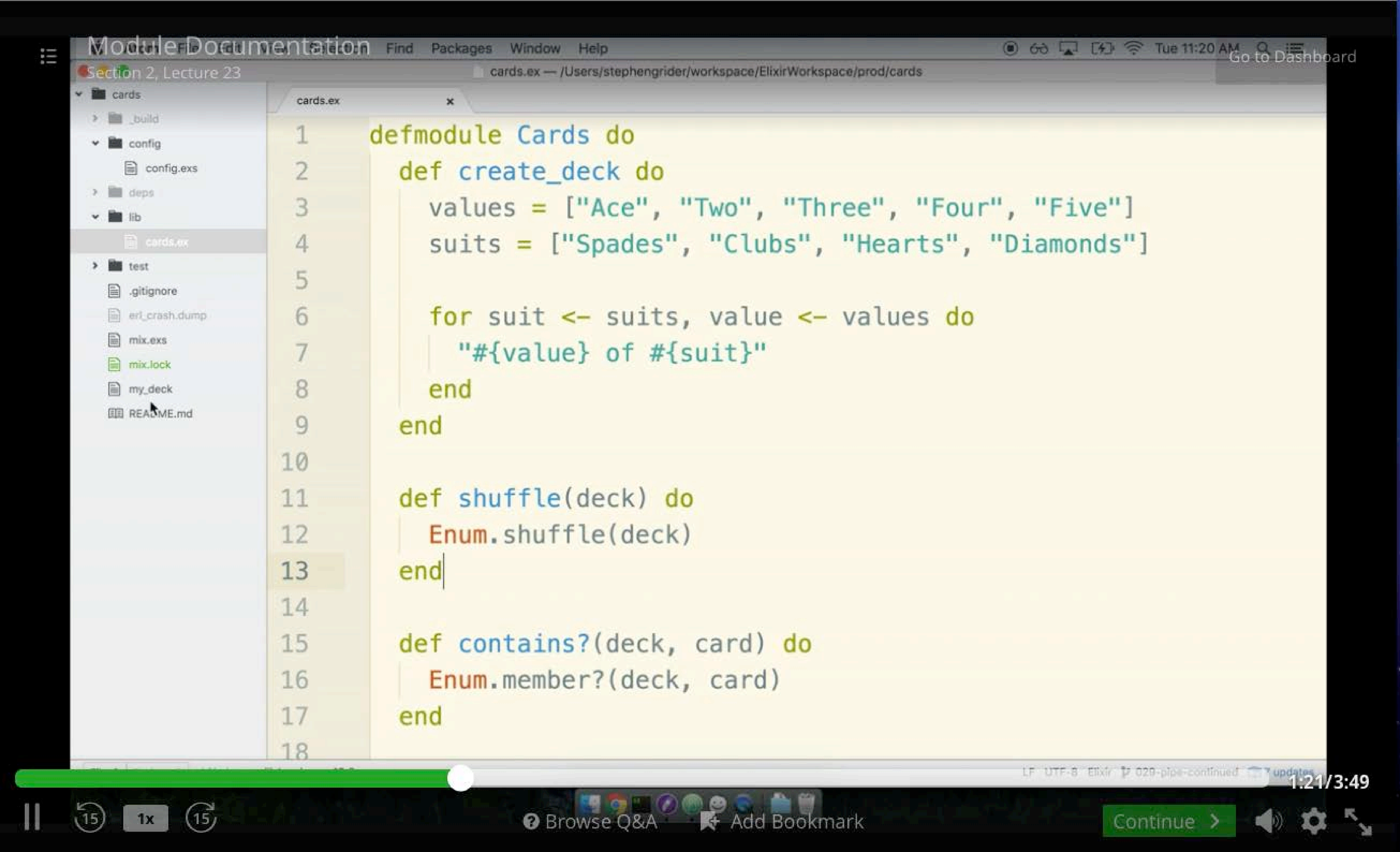

Balsamiq Mockups 3
Project
Edit
View
Window
Help
Tue 10:47 AM
01 - Pipe Operator - elixir - [.../Google Drive/Mocks/elixir.bmpr]
elixir
Quick Add
Cards.create_deck
List of Cards
Cards.shuffle
List of Cards
hand_size
Cards.deal
List of Cards


The Pipe Operator Continued
Section 2, Lecture 22
Go to Dashboard
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
config.exs
lib
cards.ex
test
.gitignore
erl_crash.dump
mix.exs
my_deck
README.md
File 0 Project 0 No Issues lib/cards.ex* 40:18
LF UTF-8 Elixir 028-pattern-matching
0:28/4:05
1x
Browse Q&A
Add Bookmark
Continue

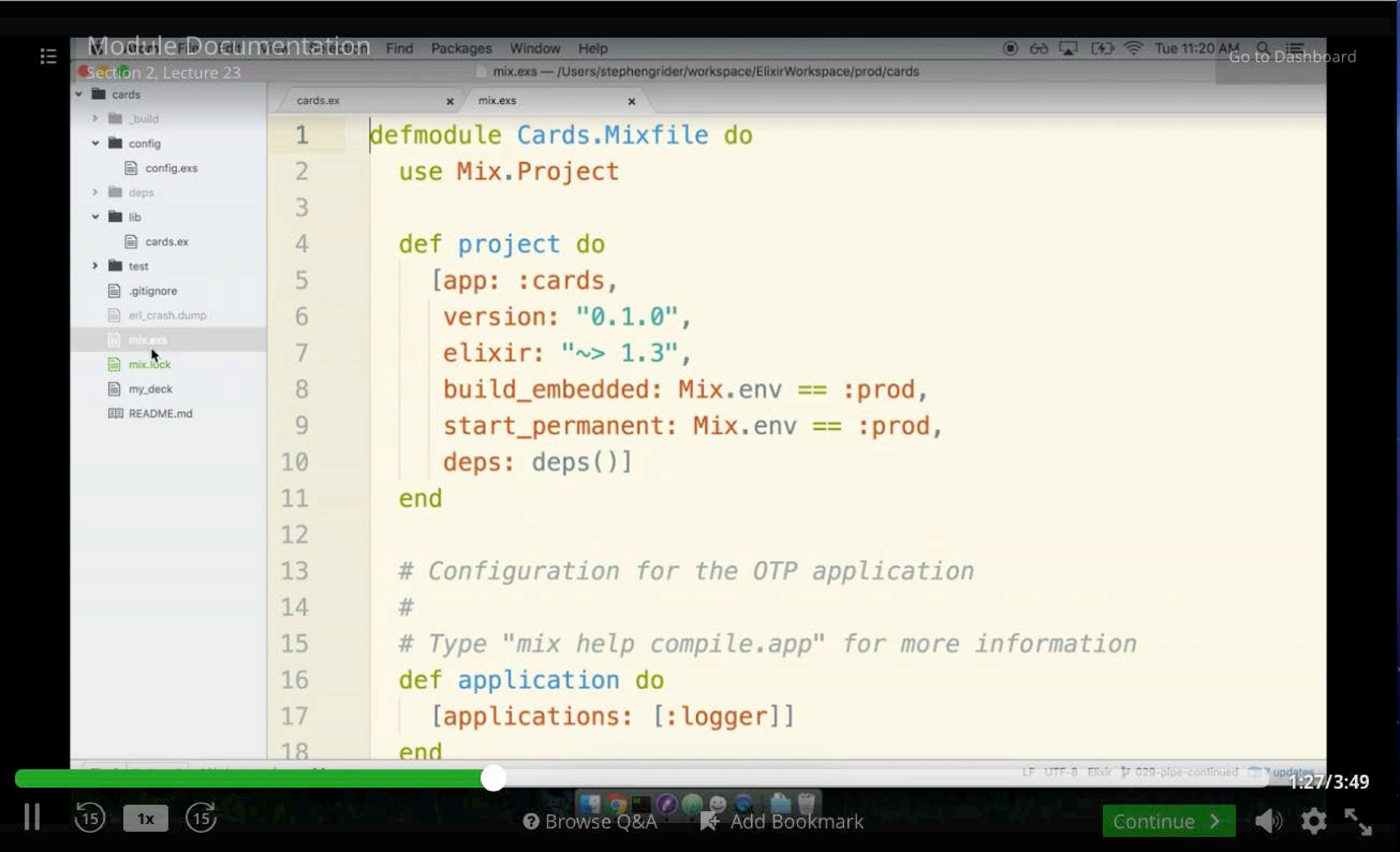
```
      {:ok, binary} -> :erlang.binary_to_term binary
      {:error, _reason} -> "That file does not exist"
    end
  end

  def create_hand(hand_size) do
    deck = Cards.create_deck
    deck = Cards.shuffle(deck)
    hand = Cards.deal(deck, hand_size)

    Cards.create_
  end
end
```


The Pipe Operator Continued
Section 2, Lecture 22
Go to Dashboard
cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
config.exs
lib
cards.ex
test
.gitignore
erl_crash.dump
mix.exs
my_deck
README.md
File 0 Project 0 No Issues lib/cards.ex 38:13
LF UTF-8 Elixir 028-pattern-matching
2:15/4:05
Browse Q&A
Add Bookmark
Continue

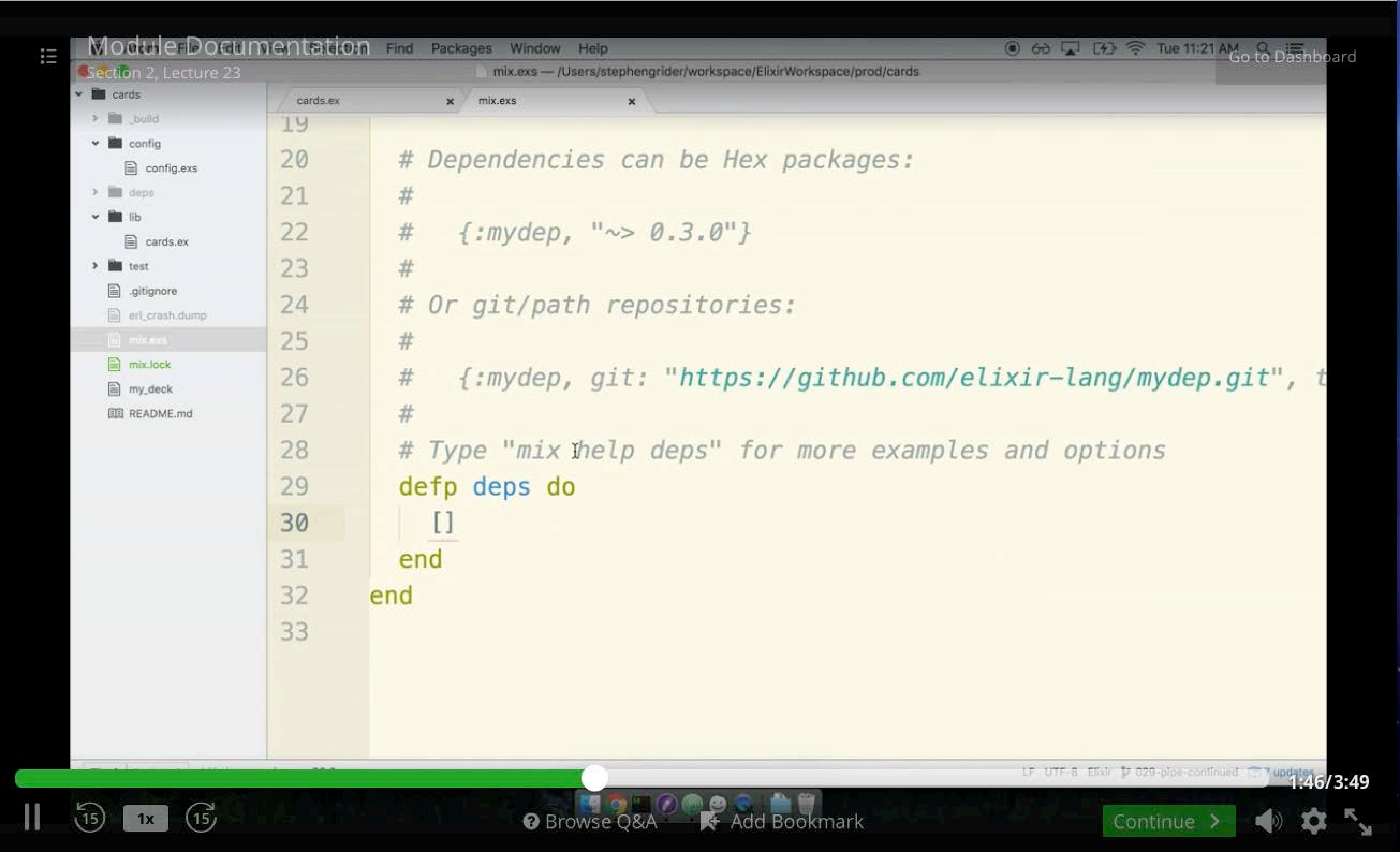
```
        {:ok, binary} -> :erlang.binary_to_term binary
        {:error, _reason} -> "That file does not exist"
      end
    end

    def create_hand(hand_size) do
      Cards.create_deck
      |> Cards.shuffle
      |> Cards.deal(hand_size)
    end
  end
```



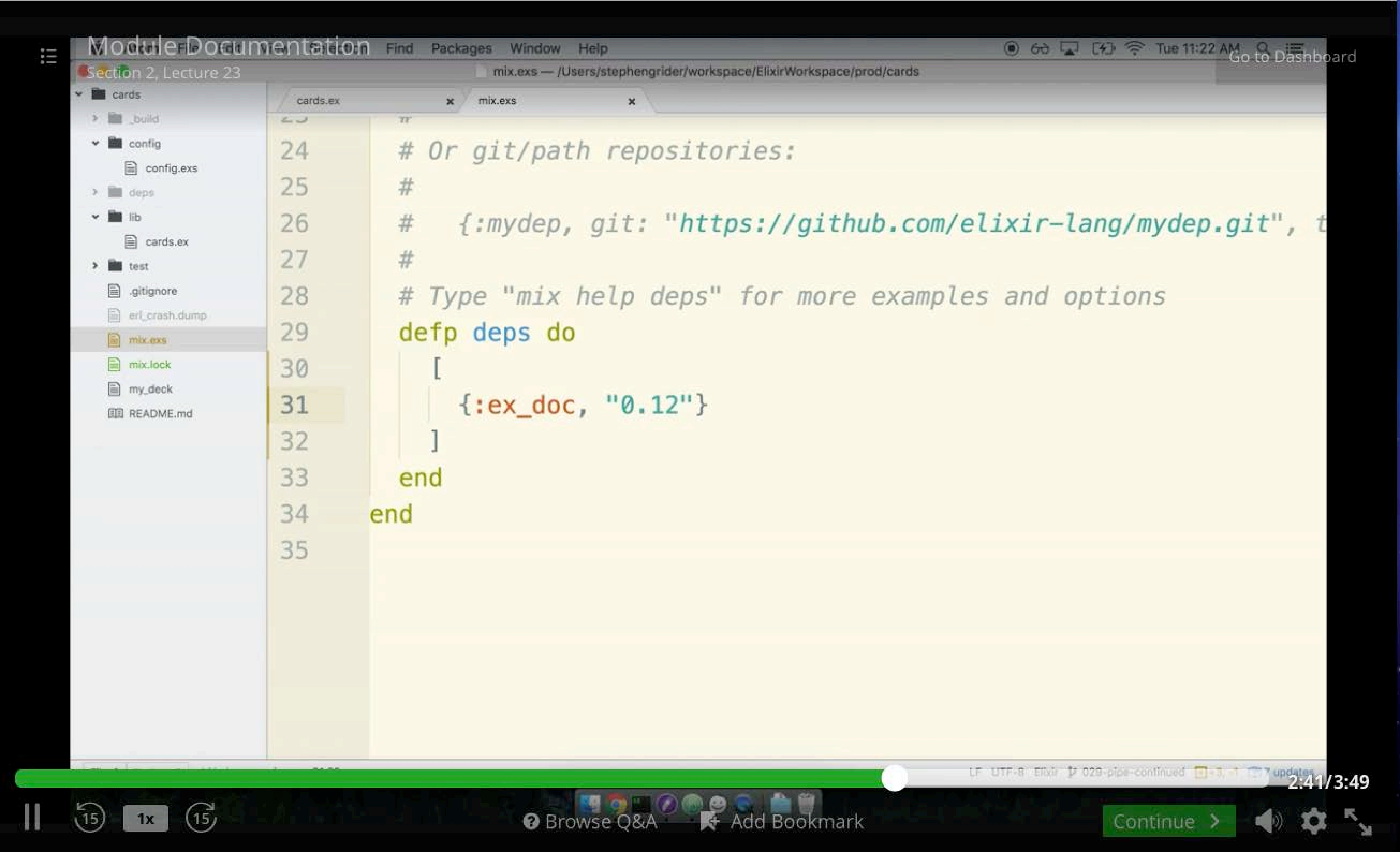
```elixir
defmodule Cards do
  def create_deck do
    values = ["Ace", "Two", "Three", "Four", "Five"]
    suits = ["Spades", "Clubs", "Hearts", "Diamonds"]

    for suit <- suits, value <- values do
      "#{value} of #{suit}"
    end
  end

  def shuffle(deck) do
    Enum.shuffle(deck)
  end

  def contains?(deck, card) do
    Enum.member?(deck, card)
  end
```



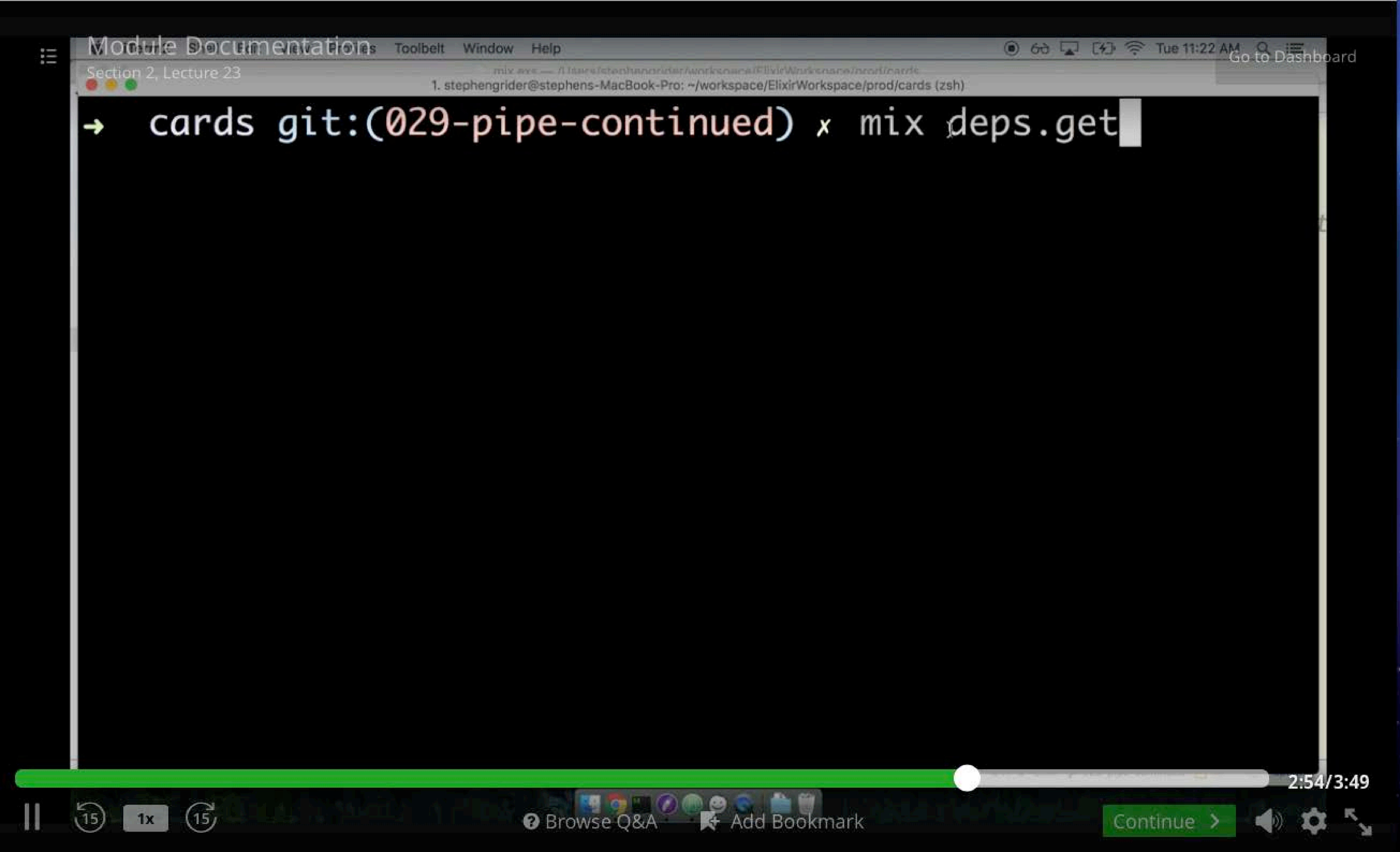
```
defmodule Cards.Mixfile do
  use Mix.Project

  def project do
    [app: :cards,
     version: "0.1.0",
     elixir: "~> 1.3",
     build_embedded: Mix.env == :prod,
     start_permanent: Mix.env == :prod,
     deps: deps()]
  end

  # Configuration for the OTP application
  #
  # Type "mix help compile.app" for more information
  def application do
    [applications: [:logger]]
  end
```



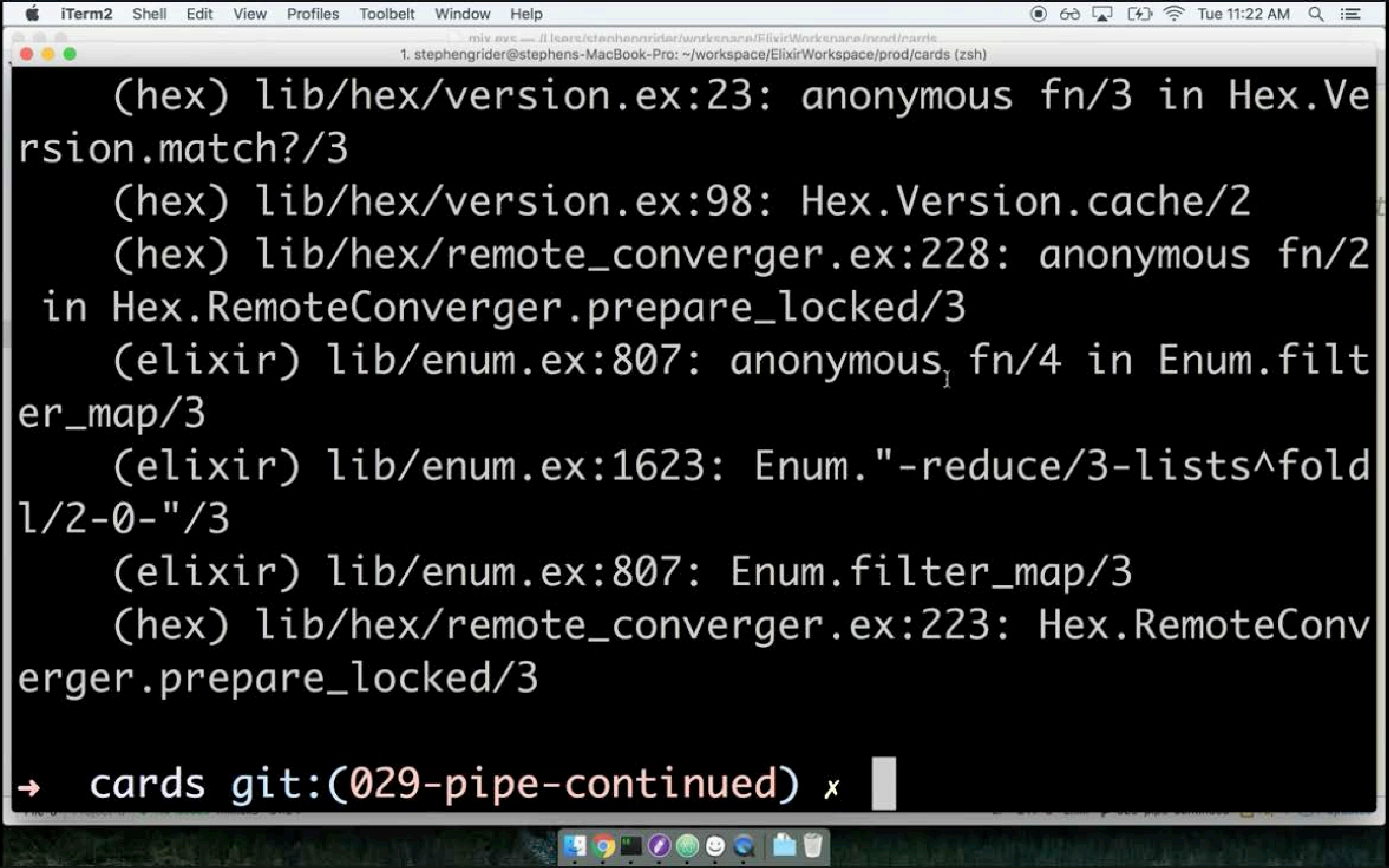
```
  # Dependencies can be Hex packages:
  #
  #   {:mydep, "~> 0.3.0"}
  #
  # Or git/path repositories:
  #
  #   {:mydep, git: "https://github.com/elixir-lang/mydep.git", t
  #
  # Type "mix help deps" for more examples and options
  defp deps do
    []
  end
end
```



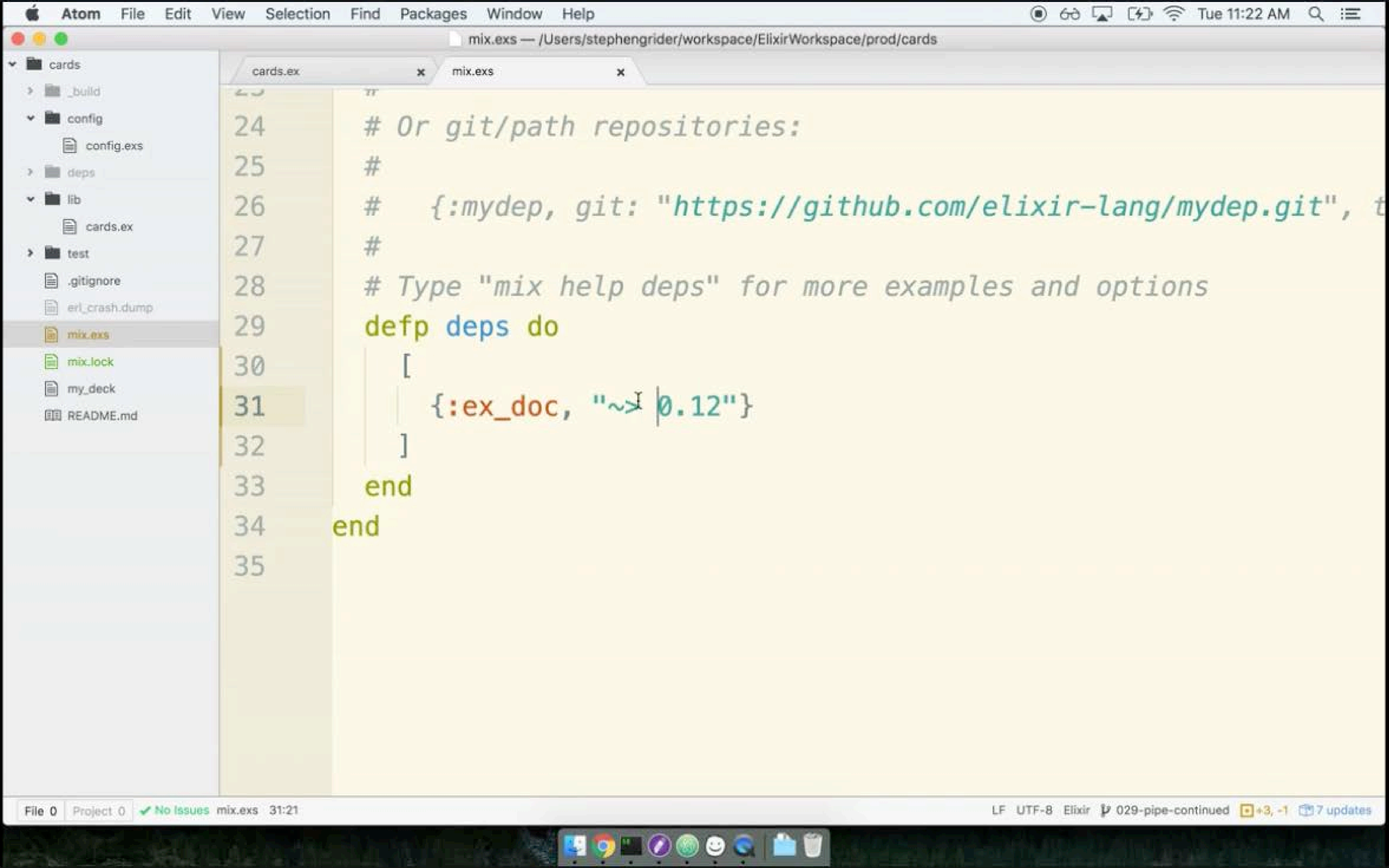
```
  # Or git/path repositories:
  #
  #   {:mydep, git: "https://github.com/elixir-lang/mydep.git", t
  #
  # Type "mix help deps" for more examples and options
  defp deps do
    [
      {:ex_doc, "0.12"}
    ]
  end
end
```



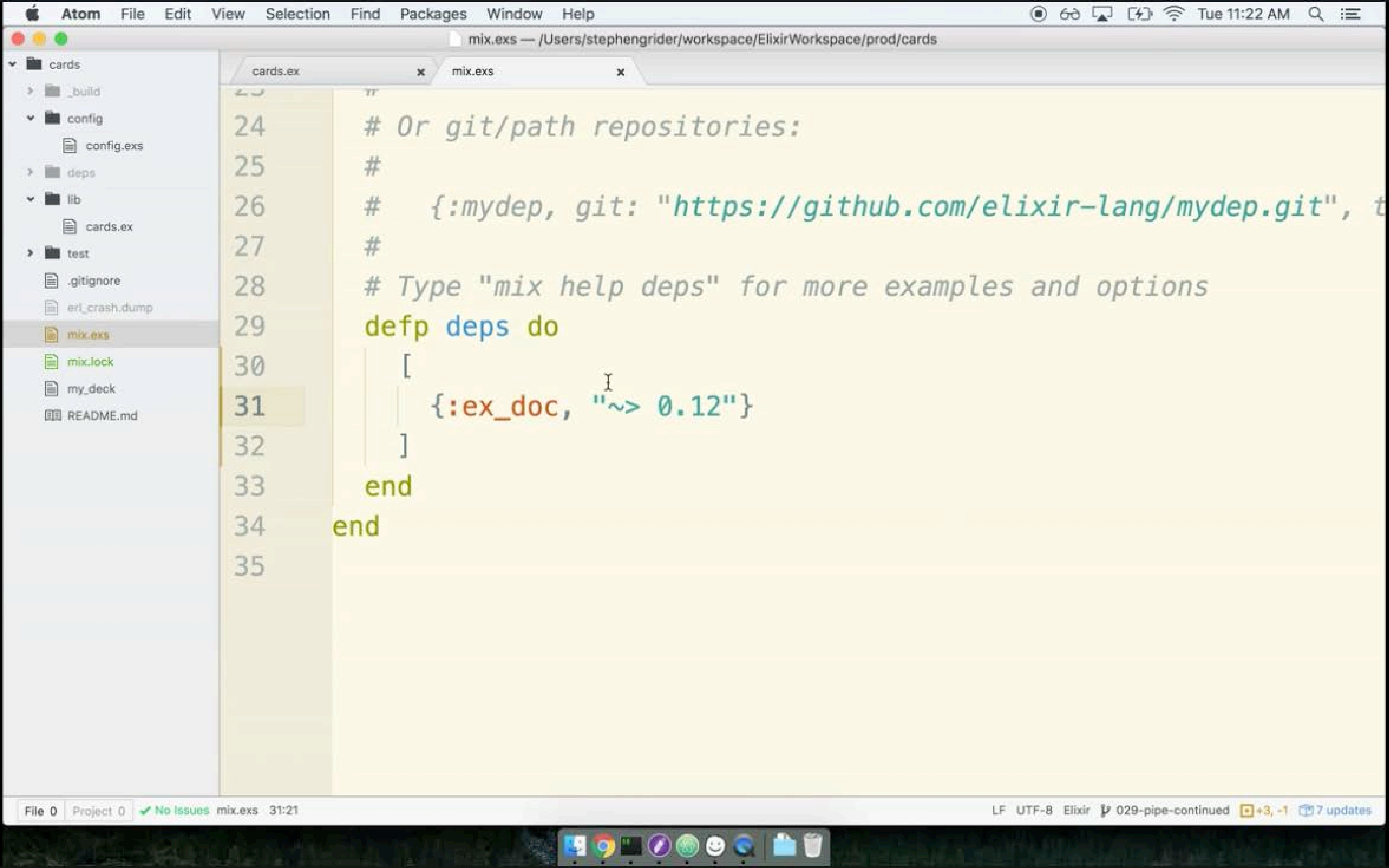
```
➜  cards git:(029-pipe-continued) ✗ mix deps.get
```


iTerm2 Shell Edit View Profiles Toolbelt Window Help
Tue 11:22 AM
1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

```
    (hex) lib/hex/version.ex:23: anonymous fn/3 in Hex.Ve
rsion.match?/3
    (hex) lib/hex/version.ex:98: Hex.Version.cache/2
    (hex) lib/hex/remote_converger.ex:228: anonymous fn/2
 in Hex.RemoteConverger.prepare_locked/3
    (elixir) lib/enum.ex:807: anonymous fn/4 in Enum.filt
er_map/3
    (elixir) lib/enum.ex:1623: Enum."-reduce/3-lists^fold
l/2-0-"/3
    (elixir) lib/enum.ex:807: Enum.filter_map/3
    (hex) lib/hex/remote_converger.ex:223: Hex.RemoteConv
erger.prepare_locked/3


→  cards git:(029-pipe-continued) ✗
```


Atom File Edit View Selection Find Packages Window Help
Tue 11:22 AM
mix.exs — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
config.exs
deps
lib
cards.ex
test
.gitignore
erl_crash.dump
mix.exs
mix.lock
my_deck
README.md
cards.ex
mix.exs
File 0 Project 0 No Issues mix.exs 31:21
LF UTF-8 Elixir 029-pipe-continued +3, -1 7 updates

```
  # Or git/path repositories:
  #
  #   {:mydep, git: "https://github.com/elixir-lang/mydep.git", t
  #
  # Type "mix help deps" for more examples and options
  defp deps do
    [
      {:ex_doc, "~> 0.12"}
    ]
  end
end
```


Atom File Edit View Selection Find Packages Window Help
Tue 11:22 AM
mix.exs — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards
cards
_build
config
config.exs
deps
lib
cards.ex
test
.gitignore
erl_crash.dump
mix.exs
mix.lock
my_deck
README.md
cards.ex
mix.exs
File 0 Project 0 ✓ No Issues mix.exs 31:21
LF UTF-8 Elixir 029-pipe-continued +3, -1 7 updates

```
  # Or git/path repositories:
  #
  #   {:mydep, git: "https://github.com/elixir-lang/mydep.git", t
  #
  # Type "mix help deps" for more examples and options
  defp deps do
    [
      {:ex_doc, "~> 0.12"}
    ]
  end
end
```