

```
→ cards git:(031-generating-docs) mix docs  
Compiling 1 file (.ex)  
Docs successfully generated.  
View them at "doc/index.html".  
→ cards git:(031-generating-docs) x
```



```

→ cards git:(031-generating-docs) mix docs
Compiling 1 file (.ex)
Docs successfully generated.
View them at "doc/index.html".
→ cards git:(031-generating-docs) x ls
README.md      doc            mix.lock
_build         erl_crash.dump my_deck
config         lib            test
deps           mix.exs
→ cards git:(031-generating-docs) x cd doc
→ doc git:(031-generating-docs) x ls
404.html      api-reference.html  fonts
Cards.html    dist                index.html
→ doc git:(031-generating-docs) x

```

1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards/doc (zsh)

Compiling 1 file (.ex)

Docs successfully generated.

View them at "doc/index.html".

→ cards git:(031-generating-docs) x ls

README.md	doc	mix.lock
_build	erl_crash.dump	my_deck
config	lib	test
deps	mix.exs	

→ cards git:(031-generating-docs) x cd doc

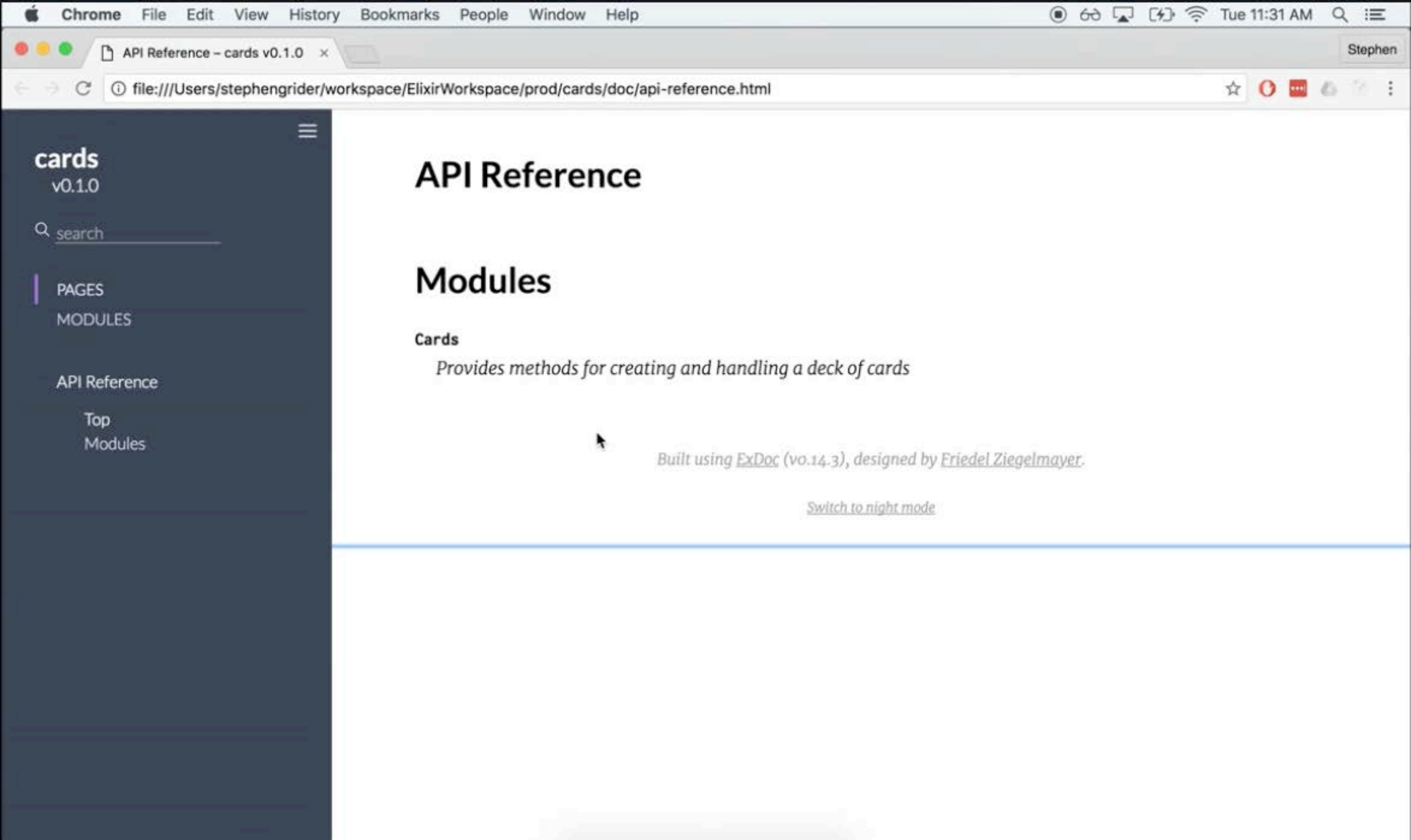
→ doc git:(031-generating-docs) x ls

404.html	api-reference.html	fonts
Cards.html	dist	index.html

→ doc git:(031-generating-docs) x open index.html

→ doc git:(031-generating-docs) x





cards  
v0.1.0

search

PAGES

MODULES

API Reference

Top  
Modules

# API Reference

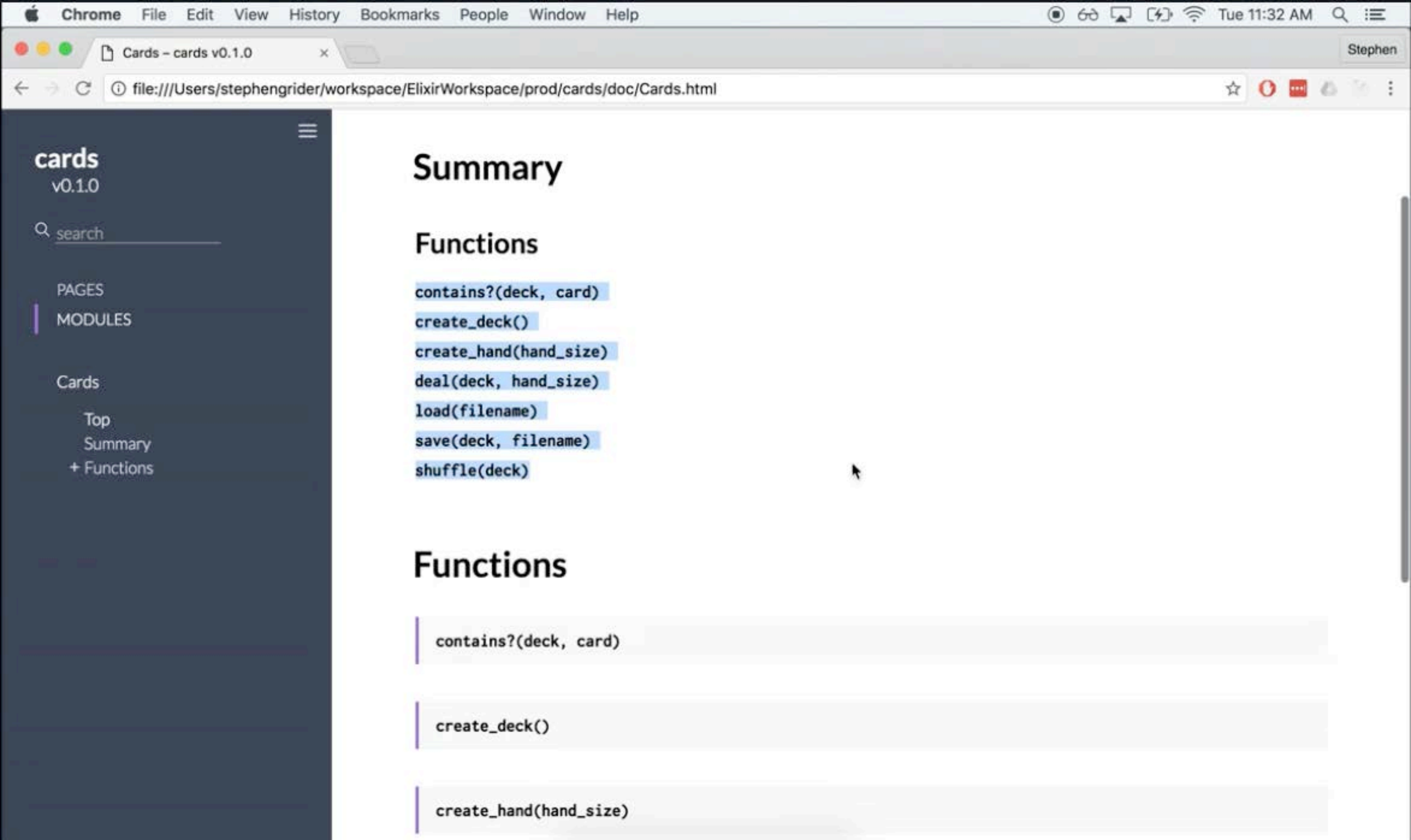
## Modules

### Cards

*Provides methods for creating and handling a deck of cards*

Built using [ExDoc](#) (v0.14.3), designed by [Friedel Ziegelmayer](#).

[Switch to night mode](#)



cards

v0.1.0

search

PAGES

MODULES

Cards

Top

Summary

+ Functions

## Summary

### Functions

`contains?(deck, card)`

`create_deck()`

`create_hand(hand_size)`

`deal(deck, hand_size)`

`load(filename)`

`save(deck, filename)`

`shuffle(deck)`

### Functions

`contains?(deck, card)`

`create_deck()`

`create_hand(hand_size)`

cards

v0.1.0

search

PAGES

MODULES

Cards

Top

Summary

+ Functions

contains?(deck, card)

create\_deck()

create\_hand(hand\_size)

deal(deck, hand\_size)

load(filename)

save(deck, filename)

shuffle(deck)

Built using [ExDoc \(v0.14.3\)](#), designed by [Friedel Ziegelmayer](#).

[Switch to night mode](#)

```

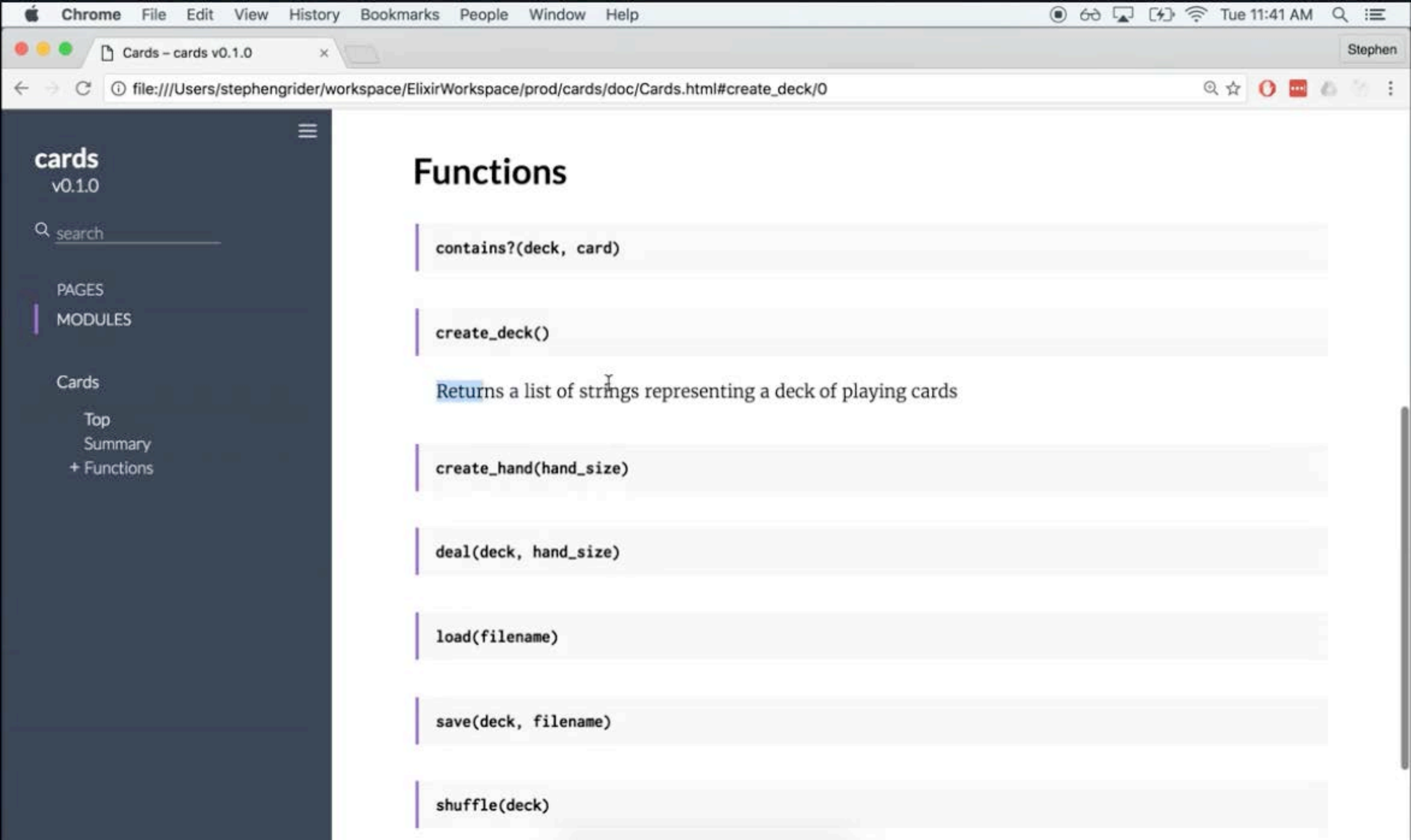
1  defmodule Cards do
2    @moduledoc """
3      Provides methods for creating and handling a deck of cards
4    """
5
6    @doc """
7      Returns a list of strings representing a deck of playing cards
8    """
9
10   def create_deck do
11     values = ["Ace", "Two", "Three", "Four", "Five"]
12     suits = ["Spades", "Clubs", "Hearts", "Diamonds"]
13
14     for suit <- suits, value <- values do
15       "#{value} of #{suit}"
16     end
17
18   def shuffle(deck) do

```



1. mix docs (sh)

→ cards git:(031-generating-docs) mix docs



cards  
v0.1.0

search

PAGES  
MODULES

Cards  
Top  
Summary  
+ Functions

## Functions

`contains?(deck, card)`

`create_deck()`

Returns a list of strings representing a deck of playing cards

`create_hand(hand_size)`

`deal(deck, hand_size)`

`load(filename)`

`save(deck, filename)`

`shuffle(deck)`

AtomFileEditViewSelectionFindPackagesWindowHelpTue 11:42 AM

cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards

cards.ex

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

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

@doc """
  Divides a deck into a hand and the remainder of the deck.
  The `hand_size` argument indicates how many cards should
  be in the hand.
"""

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
```

File 0Project 0No Issueslib/cards.ex\* 29:20LF UTF-8Elixir031-generating-docs+37 updates



```
1. mix docs (zsh)

→ cards git:(031-generating-docs) mix docs
Compiling 1 file (.ex)
Docs successfully generated.
View them at "doc/index.html".
→ cards git:(031-generating-docs) x open doc/index.html
→ cards git:(031-generating-docs) x mix docs
```

Chrome

File Edit View History Bookmarks People Window Help

Tue 11:43 AM

Cards - cards v0.1.0

file:///Users/stephengrider/workspace/ElixirWorkspace/prod/cards/doc/Cards.html#create\_deck/0

Stephen

cards

v0.1.0

search

PAGES

MODULES

Cards

Top

Summary

- Functions

contains?/2

> create\_deck/0

create\_hand/1

deal/2

load/1

save/2

shuffle/1

Returns a list of strings representing a deck of playing cards

create\_hand(hand\_size)

deal(deck, hand\_size)

Divides a deck into a hand and the remainder of the deck. The hand\_size argument indicates how many cards should be in the hand.

load(filename)

save(deck, filename)

shuffle(deck)



```
iTerm2 Shell Edit View Profiles Toolbelt Window Help
1. iex -S mix (beam.smp)

"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(2)> {hand, deck} = Cards.deal(deck, 1)
{"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(3)> hand
"Ace of Spades"
iex(4)>
```



AtomFile Edit View Selection Find Packages Window Help

cards.ex — /Users/stephengrider/workspace/ElixirWorkspace/prod/cards

cards.ex

24end

25

26@doc """

27 Divides a deck into a hand and the remainder of the deck.

28 The `hand\_size` argument indicates how many cards should

29 be in the hand.

30

31## Examples

32

33 iex> deck = Cards.create\_deck

34 iex> {hand, deck} = Cards.deal(deck, 1)

35 iex> hand

36 ["Ace of Spades"]

37

38 """

39def deal(deck, hand\_size) do

40 Enum.split(deck, hand\_size)

41end

File 0Project 0

✓ No Issueslib/cards.ex\* 36:17

LF UTF-8Elixir 031-generating-docs +8 7 updates

```
iTerm2 Shell Edit View Profiles Toolbelt Window Help
1. mix docs (beam.smp)

"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(3)> hand
["Ace of Spades"]
iex(4)>
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded
        (v)ersion (k)ill (D)b-tables (d)istribution
^C%
→ cards git:(031-generating-docs) x
→ cards git:(031-generating-docs) x mix docs
Compiling 1 file (.ex)
```



**cards**

v0.1.0

Q search

## PAGES

## MODULES

## Cards

Top

## Summary

- Functions

contains?/2

```
> create_deck/0
```

```
create hand/1
```

deal/2

load/1

save/2

shuffle/1

```
deal(deck, hand_size)
```

Divides a deck into a hand and the remainder of the deck. The `hand_size` argument indicates how many cards should be in the hand.

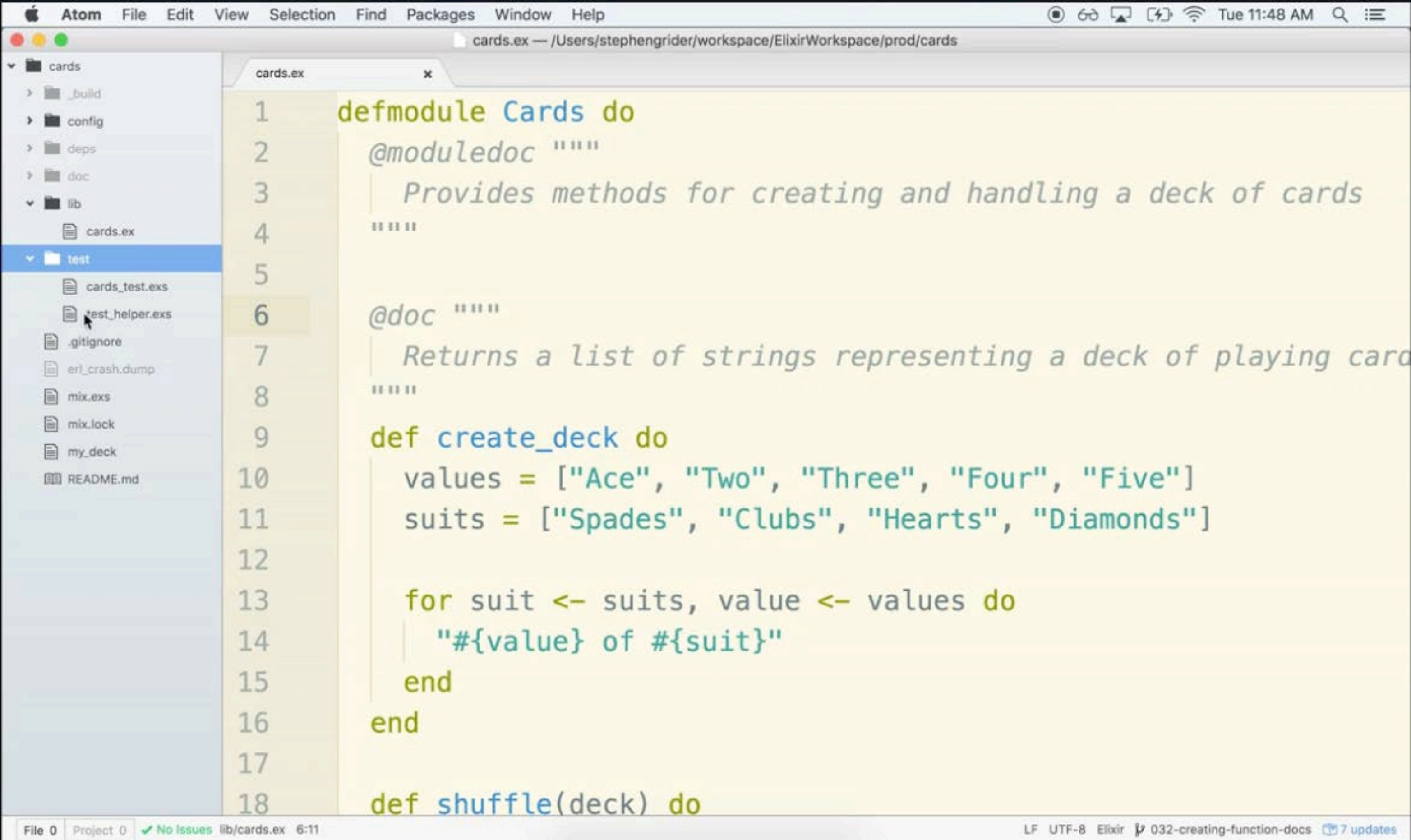
## Examples

```
iex> deck = Cards.create_deck
iex> {hand, deck} = Cards.deal(deck, 1)
iex> hand
["Ace of Spades"]
```

`load(filename)`

```
save(deck, filename)
```







```
1 defmodule CardsTest do
2   use ExUnit.Case
3   doctest Cards
4
5   test "the truth" do
6     assert 1 + 1 == 2
7   end
8 end
9
```

```
iTerm2  Shell  Edit  View  Profiles  Toolbelt  Window  Help
1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (beam.smp)

Compiling 14 files (.ex)
Generated ex_doc app
==> cards
Compiling 1 file (.ex)
Generated cards app
warning: variable deck is unused
  (for doctest at) lib/cards.ex:34

..

Finished in 0.06 seconds
2 tests, 0 failures

Randomized with seed 496208
→ cards git:(032-creating-function-docs)
```



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

Generated earmark app
==> ex_doc
Compiling 14 files (.ex)
Generated ex_doc app
==> cards
Compiling 1 file (.ex)
Generated cards app
warning: variable deck is unused
  (for doctest at) lib/cards.ex:34

..

Finished in 0.06 seconds
2 tests, 0 failures
```

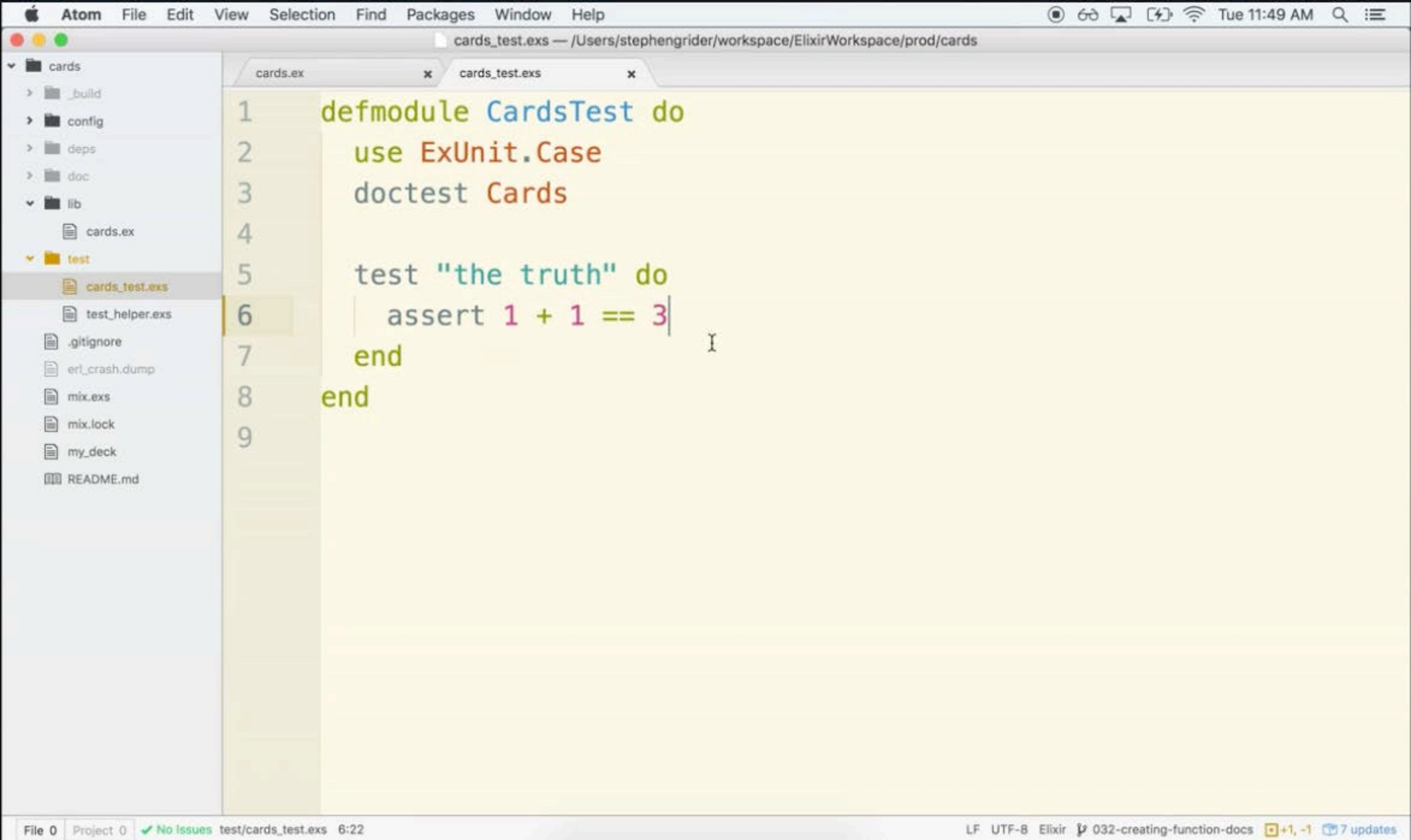
```
iTerm2  Shell  Edit  View  Profiles  Toolbelt  Window  Help
1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

Compiling 14 files (.ex)
Generated ex_doc app
==> cards
Compiling 1 file (.ex)
Generated cards app
warning: variable deck is unused
  (for doctest at) lib/cards.ex:34

..

Finished in 0.06 seconds
2 tests, 0 failures

Randomized with seed 496208
→ cards git:(032-creating-function-docs)
```



cards

- > \_build
- > config
- > deps
- > doc
- ▼ lib
  - cards.ex
- ▼ test
  - cards\_test.exs
  - test\_helper.exs
- .gitignore
- erl\_crash.dump
- mix.exs
- mix.lock
- my\_deck
- README.md

```
1 defmodule CardsTest do
2   use ExUnit.Case
3   doctest Cards
4
5   test "the truth" do
6     assert 1 + 1 == 3
7   end
8 end
9
```



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

Compiling 14 files (.ex)
Generated ex_doc app
==> cards
Compiling 1 file (.ex)
Generated cards app
warning: variable deck is unused
  (for doctest at) lib/cards.ex:34

..

Finished in 0.06 seconds
2 tests, 0 failures

Randomized with seed 496208
→ cards git:(032-creating-function-docs) mix test
```

1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

```
test/cards_test.exs:5
```

```
Assertion with == failed
```

```
code: 1 + 1 == 3
```

```
lhs: 2
```

```
rhs: 3
```

```
stacktrace:
```

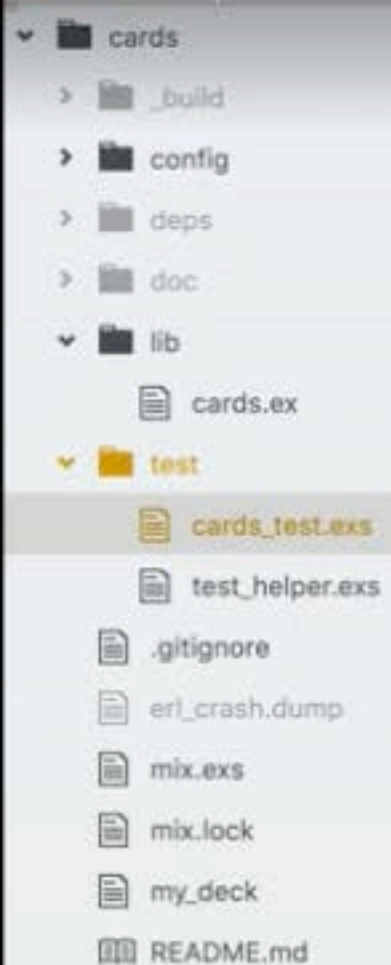
```
test/cards_test.exs:6: (test)
```

```
Finished in 0.09 seconds
```

```
2 tests, 1 failure
```

```
Randomized with seed 837699
```

```
→ cards git:(032-creating-function-docs) x
```



```
1  defmodule CardsTest do
2    use ExUnit.Case
3    doctest Cards
4
5    test "the truth" do
6      | assert 1 + 1 == 3
7    end
8  end
9
```

I



1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

Finished in 0.08 seconds

2 tests, 1 failure

Randomized with seed 704310

→ cards git:(032-creating-function-docs) x mix test

warning: variable deck is unused  
(for doctest at) lib/cards.ex:34

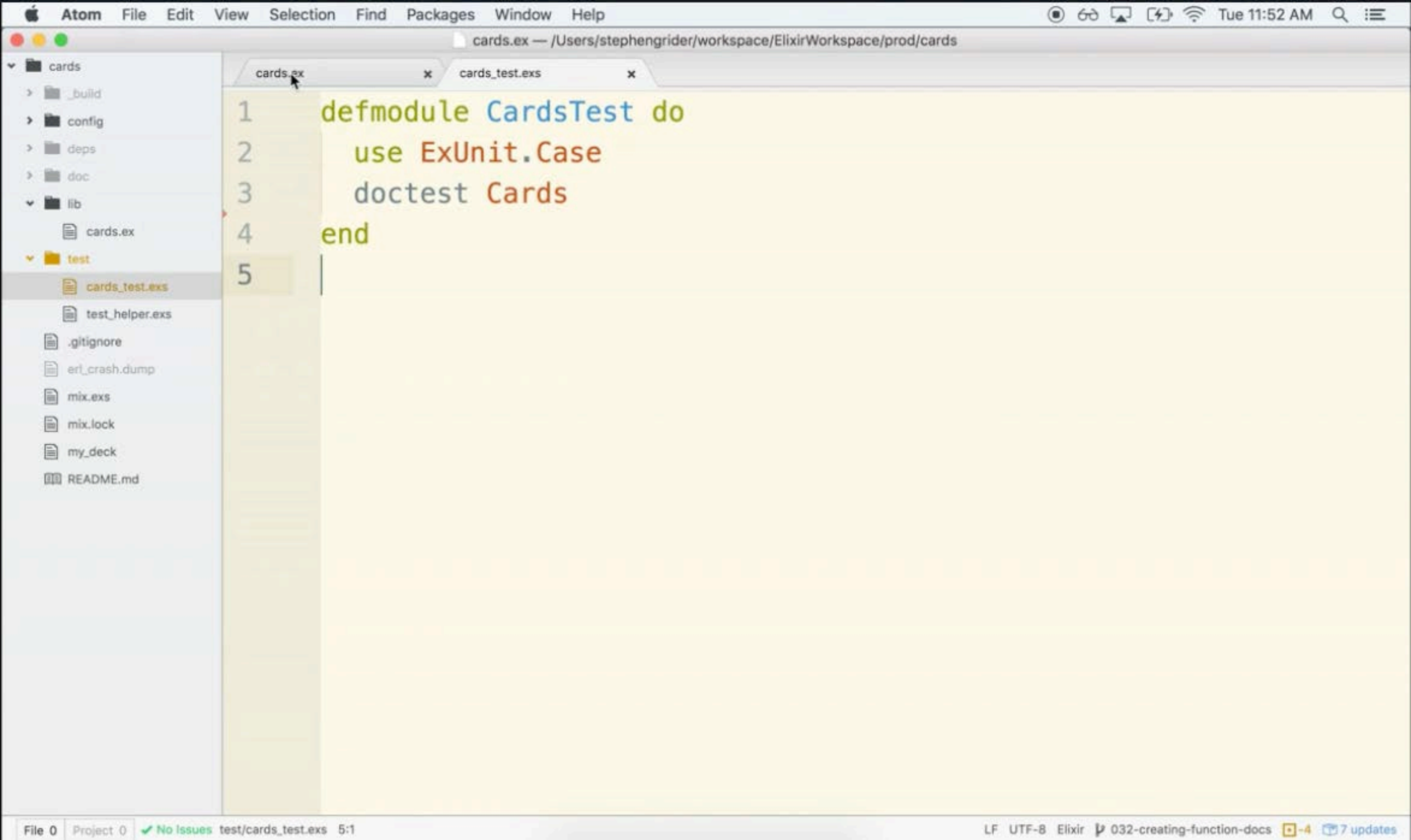
.

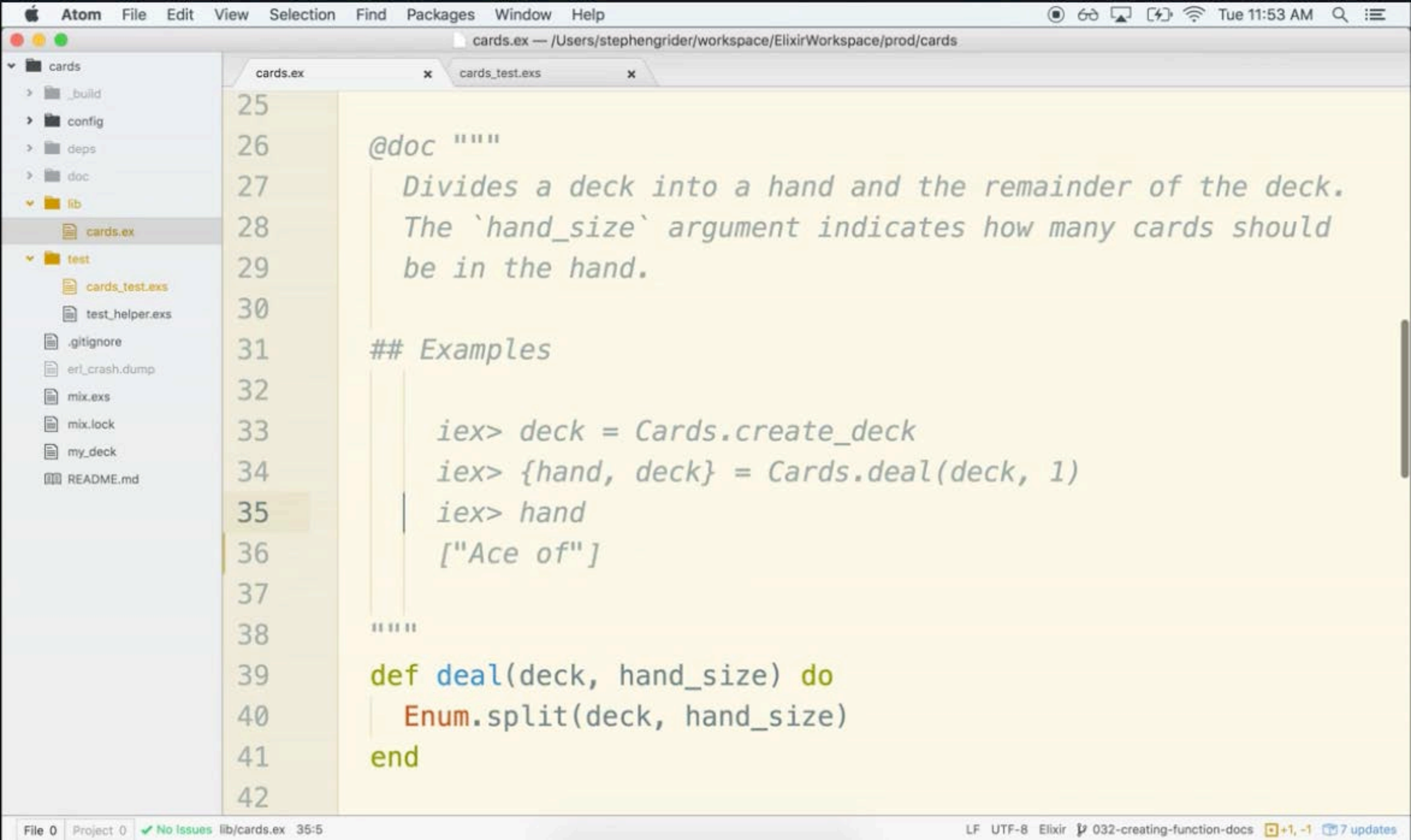
Finished in 0.06 seconds

1 test, 0 failures

Randomized with seed 89962

→ cards git:(032-creating-function-docs) x







1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

1) test doc at Cards.deal/2 (1) (CardsTest)

test/cards\_test.exs:3

Doctest failed

code: deck = Cards.create\_deck

      {hand, deck} = Cards.deal(deck, 1)

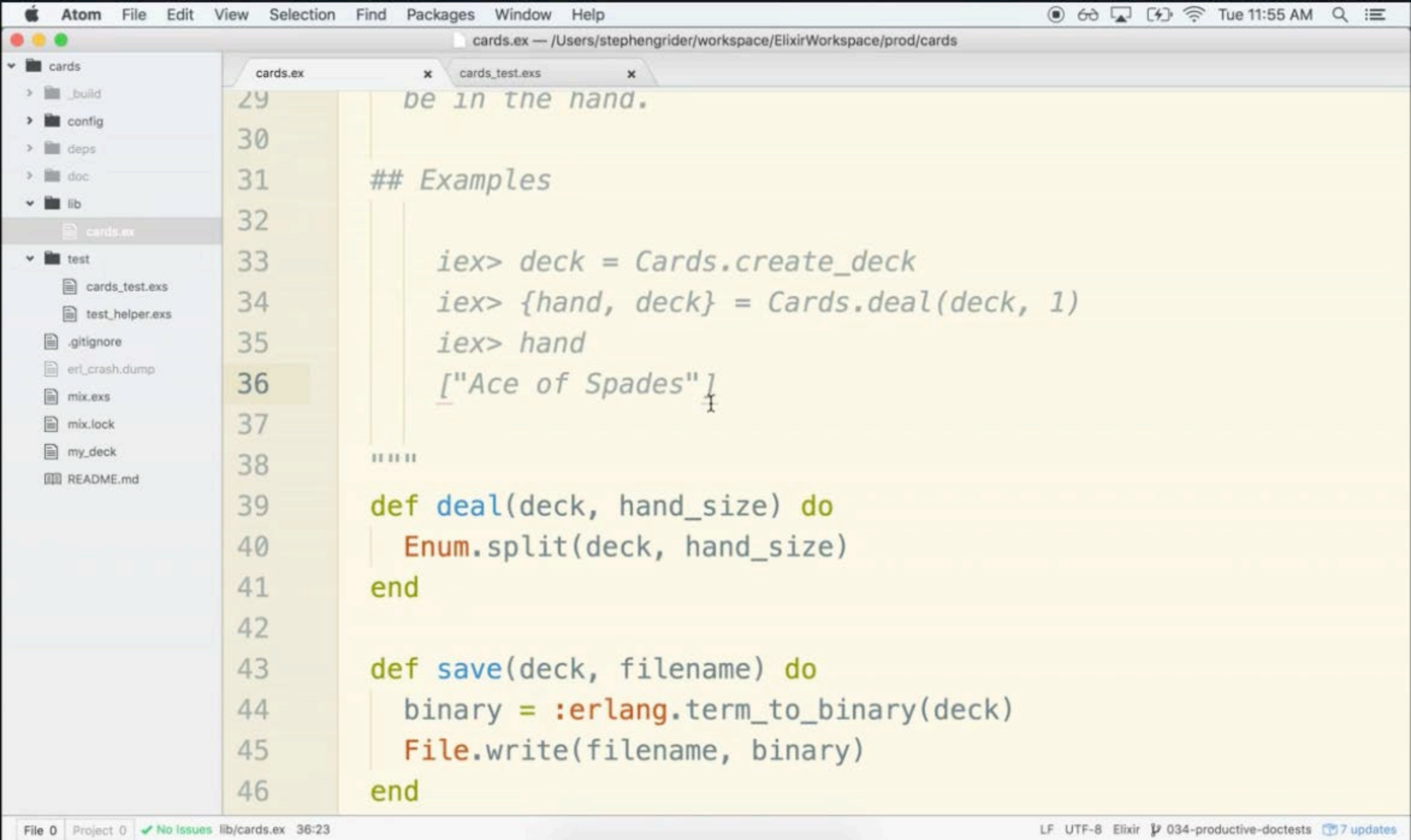
      hand == ["Ace of"]

lhs: ["Ace of Spades"]

stacktrace:

lib/cards.ex:33: Cards (module)

Finished in 0.06 seconds



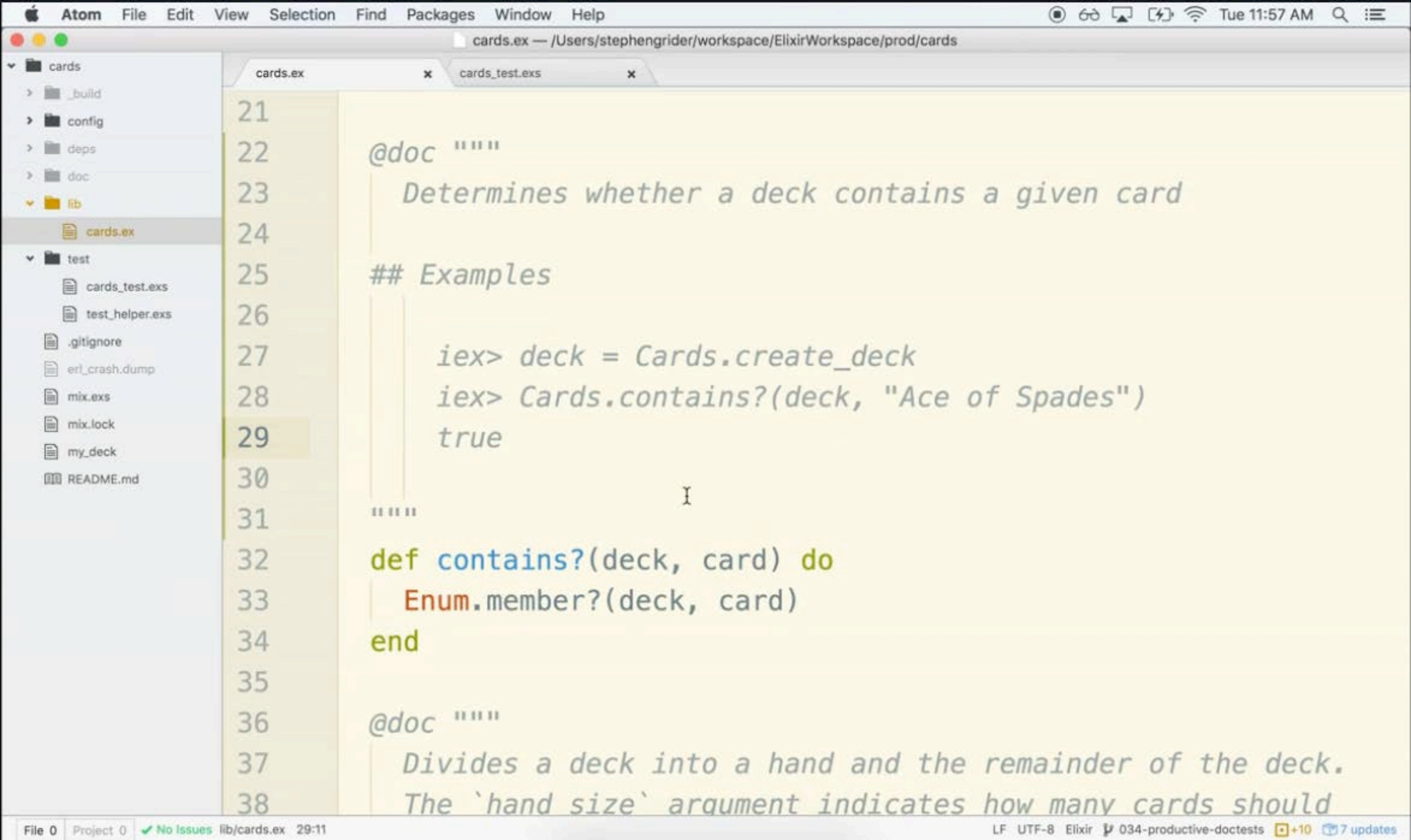


```
iTerm2 Shell Edit View Profiles Toolbelt Window Help
1. iex -S mix (beam.smp)

eads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.3.4) - press Ctrl+C to exit (type h() ENTER
for help)
iex(1)> 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(2)> Cards.contains?(deck, "Ace of Spades")
true
iex(3)>
```





```
"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(2)> Cards.contains?(deck, "Ace of Spades")
```

```
true
```

```
iex(3)>
```

```
BREAK: (a)bort (c)ontinue (p)roc info (i)nfo (l)oaded  
(v)ersion (k)ill (D)b-tables (d)istribution
```

```
^C%
```

```
→ cards git:(034-productive-doctests) x
```

```
→ cards git:(034-productive-doctests) x
```

```
→ cards git:(034-productive-doctests) x mix tests
```

```
Compiling 1 file (.ex)
```



```
iTerm2  Shell  Edit  View  Profiles  Toolbelt  Window  Help
1. mix test (beam.smp)

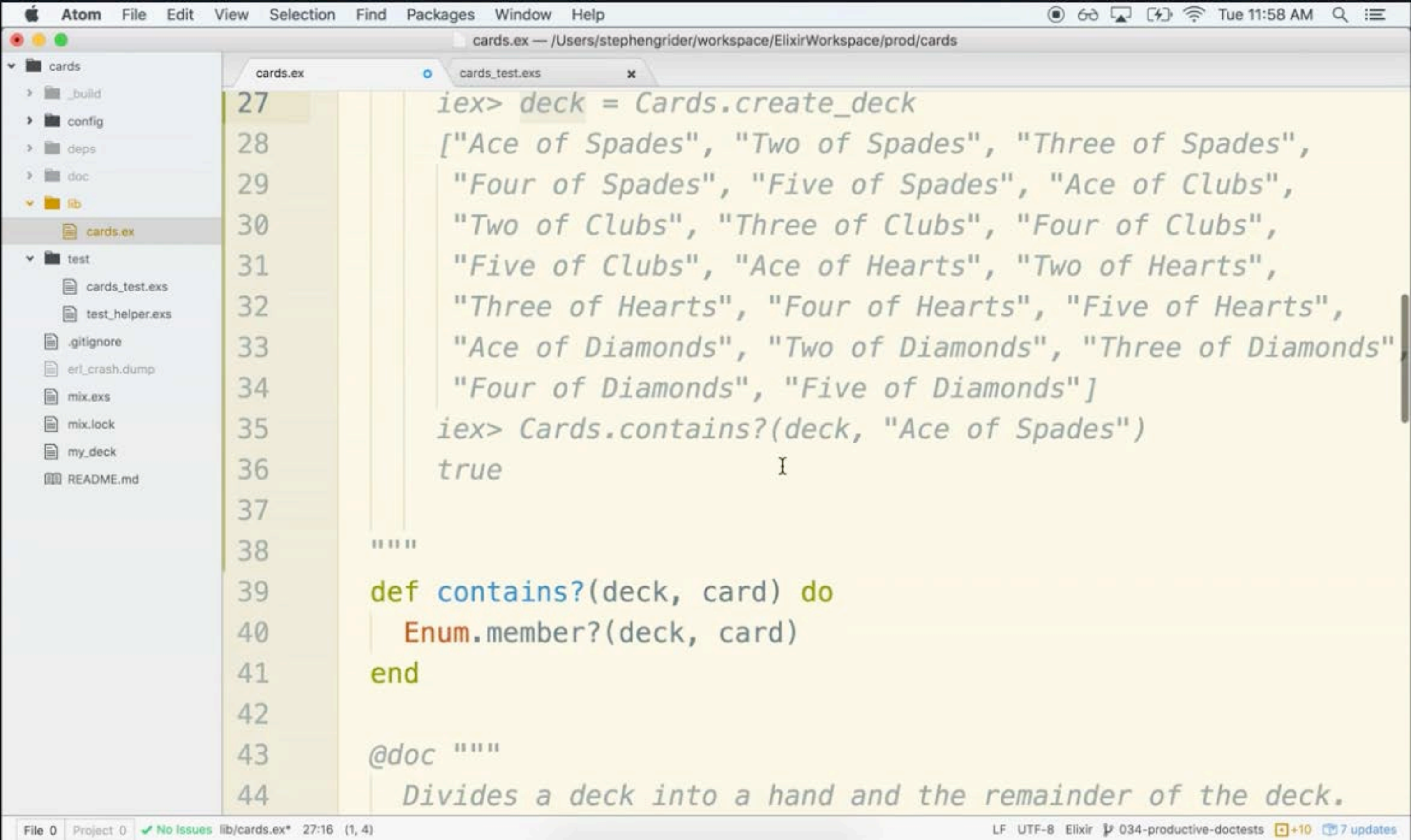
Compiling 1 file (.ex)
** (Mix) The task "tests" could not be found. Did you mean "test"?
→ cards git:(034-productive-doctests) x mix test
Compiling 1 file (.ex)
warning: variable deck is unused
  (for doctest at) lib/cards.ex:44

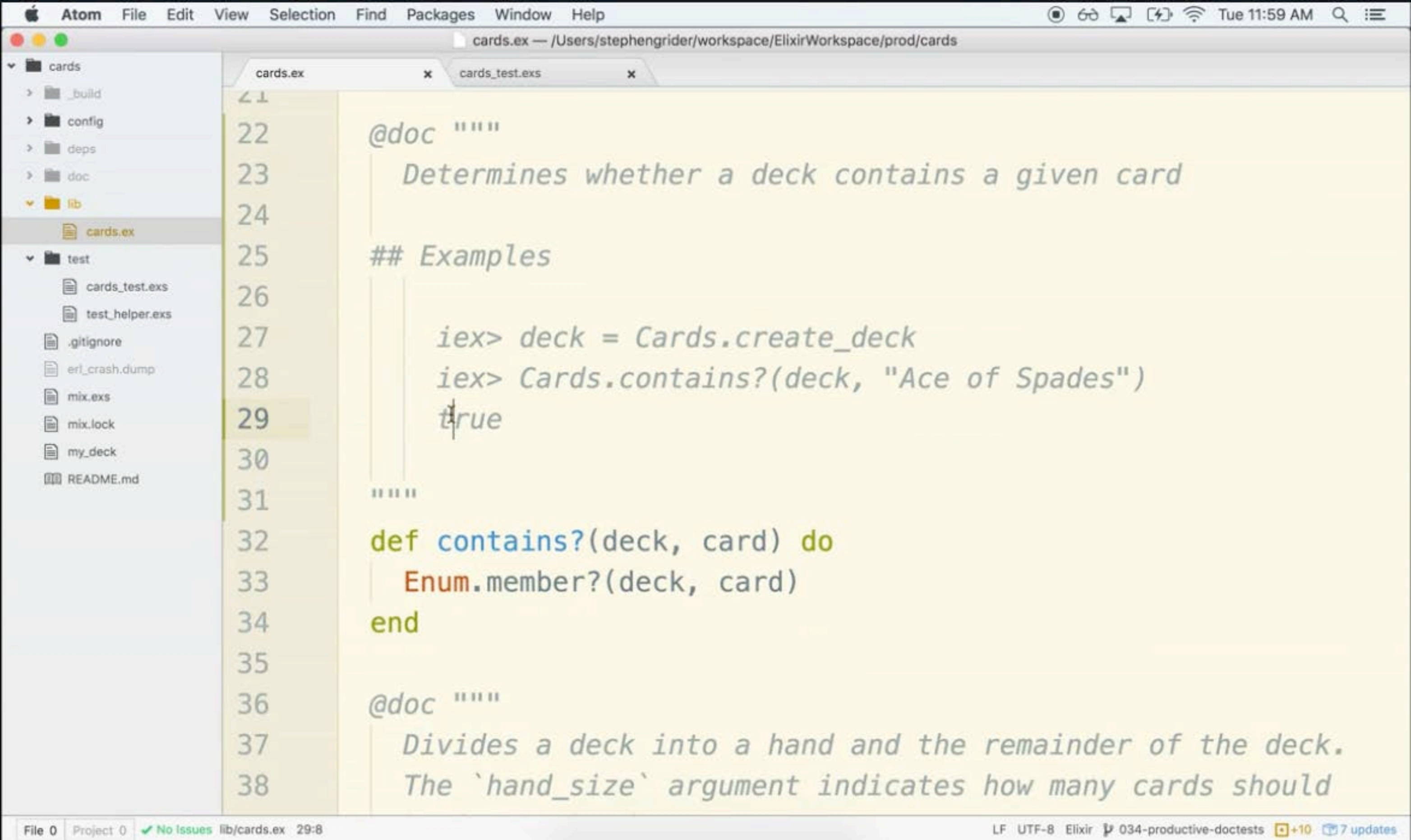
..

Finished in 0.08 seconds
2 tests, 0 failures

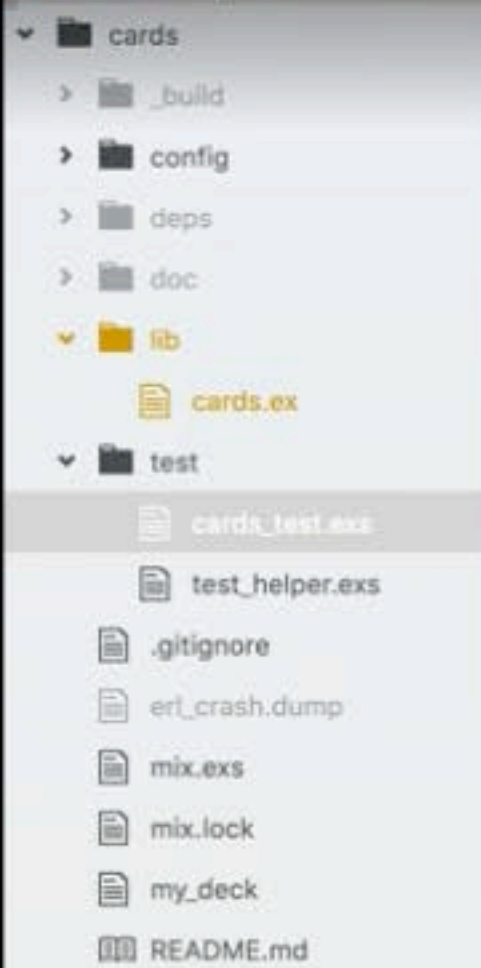
Randomized with seed 106873
```











```
1  defmodule CardsTest do
2    use ExUnit.Case
3    doctest Cards
4  end
5
```



cards

- \_build
- config
- deps
- doc
- lib
  - cards.ex
- test
  - cards\_test.exs
  - test\_helper.exs
- .gitignore
- erl\_crash.dump
- mix.exs
- mix.lock
- my\_deck
- README.md

```

1  defmodule CardsTest do
2      use ExUnit.Case
3      doctest Cards
4
5      test "create_deck makes 20 cards" do
6          deck_length = length(Cards.create_deck)
7          assert deck_length == 20
8      end
9  end
10

```

cards

\_build

config

deps

doc

lib

cards.ex

test

cards\_test.exs

test\_helper.exs

.gitignore

erl\_crash.dump

mix.exs

mix.lock

my\_deck

README.md

cards.ex | cards\_test.exs

1

2

3

4

5

6

7

8

9

10

defmodule CardsTest do
 use ExUnit.Case
 doctest Cards

 test "create\_deck makes 20 cards" do
 deck\_length = length(Cards.create\_deck)
 assert deck\_length == 20
 end
end

→ cards git:(035-writing-effective-doctests) x mix test

warning: variable deck is unused  
(for doctest at) lib/cards.ex:44

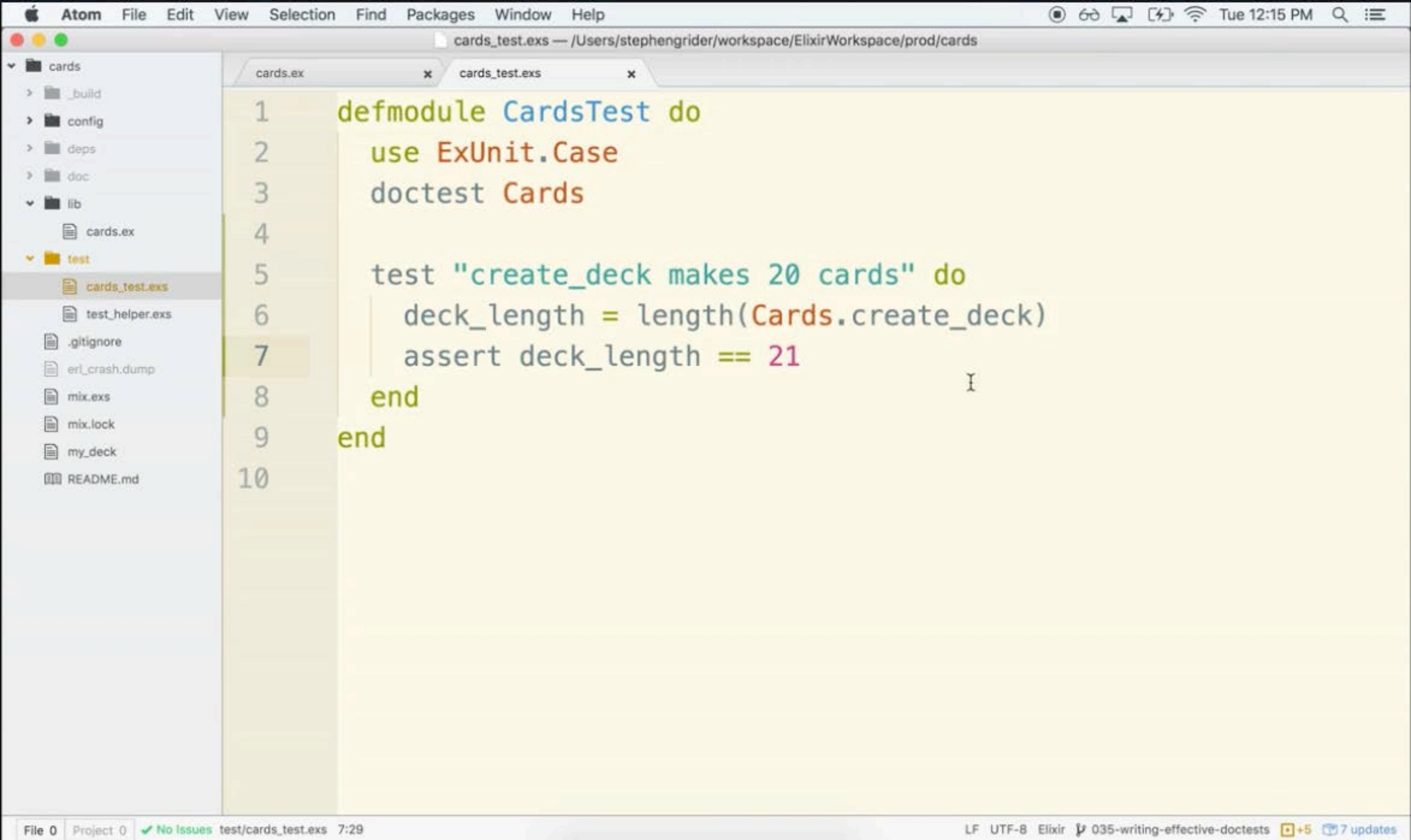
...

Finished in 0.1 seconds  
3 tests, 0 failures

Randomized with seed 30567

→ cards git:(035-writing-effective-doctests) x █





cards

- \_build
- config
- deps
- doc
- lib
  - cards.ex
- test
  - cards\_test.exs
  - test\_helper.exs
- .gitignore
- erl\_crash.dump
- mix.exs
- mix.lock
- my\_deck
- README.md

```
1 defmodule CardsTest do
2   use ExUnit.Case
3   doctest Cards
4
5   test "create_deck makes 20 cards" do
6     deck_length = length(Cards.create_deck)
7     assert deck_length == 21
8   end
9 end
10
```

1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

test/cards\_test.exs:5

Assertion with == failed

code: deck\_length == 21

lhs: 20

rhs: 21

stacktrace:

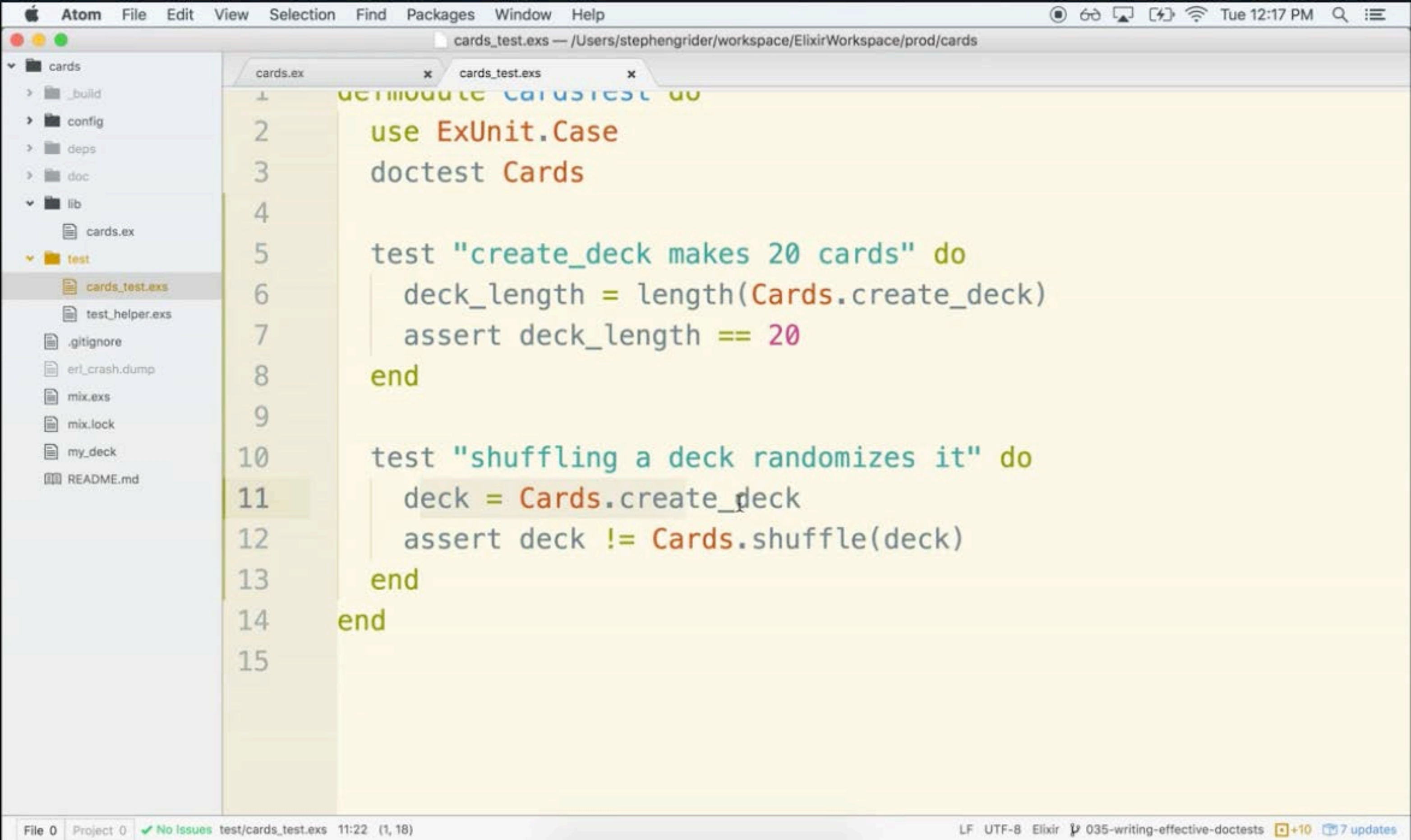
test/cards\_test.exs:7: (test)

Finished in 0.09 seconds

3 tests, 1 failure

Randomized with seed 242578

→ cards git:(035-writing-effective-doctests) x





1. stephengrider@stephens-MacBook-Pro: ~/workspace/ElixirWorkspace/prod/cards (zsh)

Finished in 0.09 seconds

3 tests, 1 failure

Randomized with seed 242578

→ cards git:(035-writing-effective-doctests) x mix test

warning: variable deck is unused  
(for doctest at) lib/cards.ex:44

....

Finished in 0.1 seconds

4 tests, 0 failures

Randomized with seed 918666

→ cards git:(035-writing-effective-doctests) x

