Polymorphism with Protocols

Section 2

In this Section, we are going to take a look at…

- Introduction
- Defining a Protocol
- Implementing a Protocol

Defining Protocols

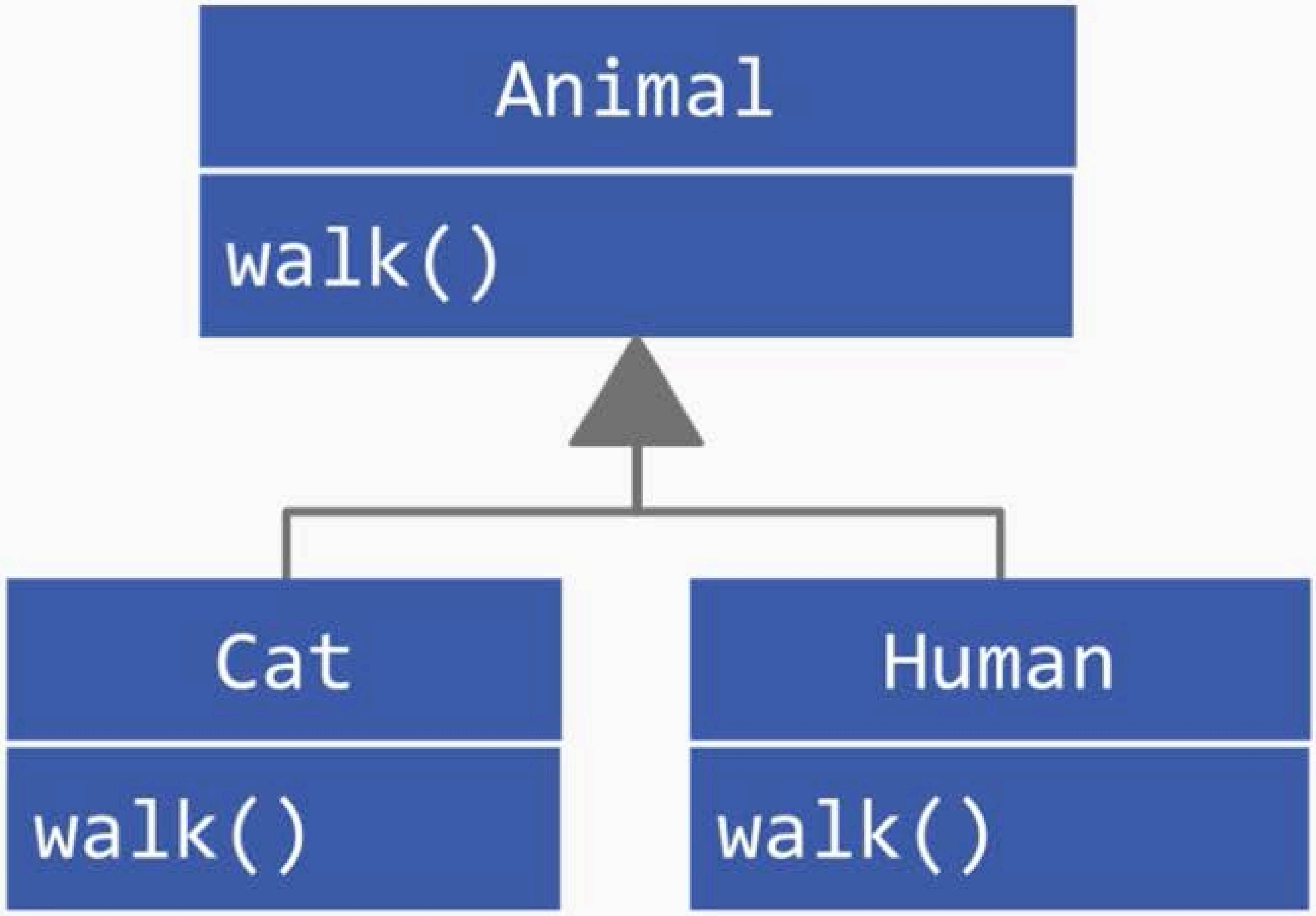
In this Video, we are going to take a look at…

- What is a Polymorphism
- Protocols and Polymorphism
- Defining Protocols

What is Polymorphism?

Polymorphism

Presenting the same interface for different underlying data types.



Animal

walk()

Cat

walk()

Human

walk()

Protocols

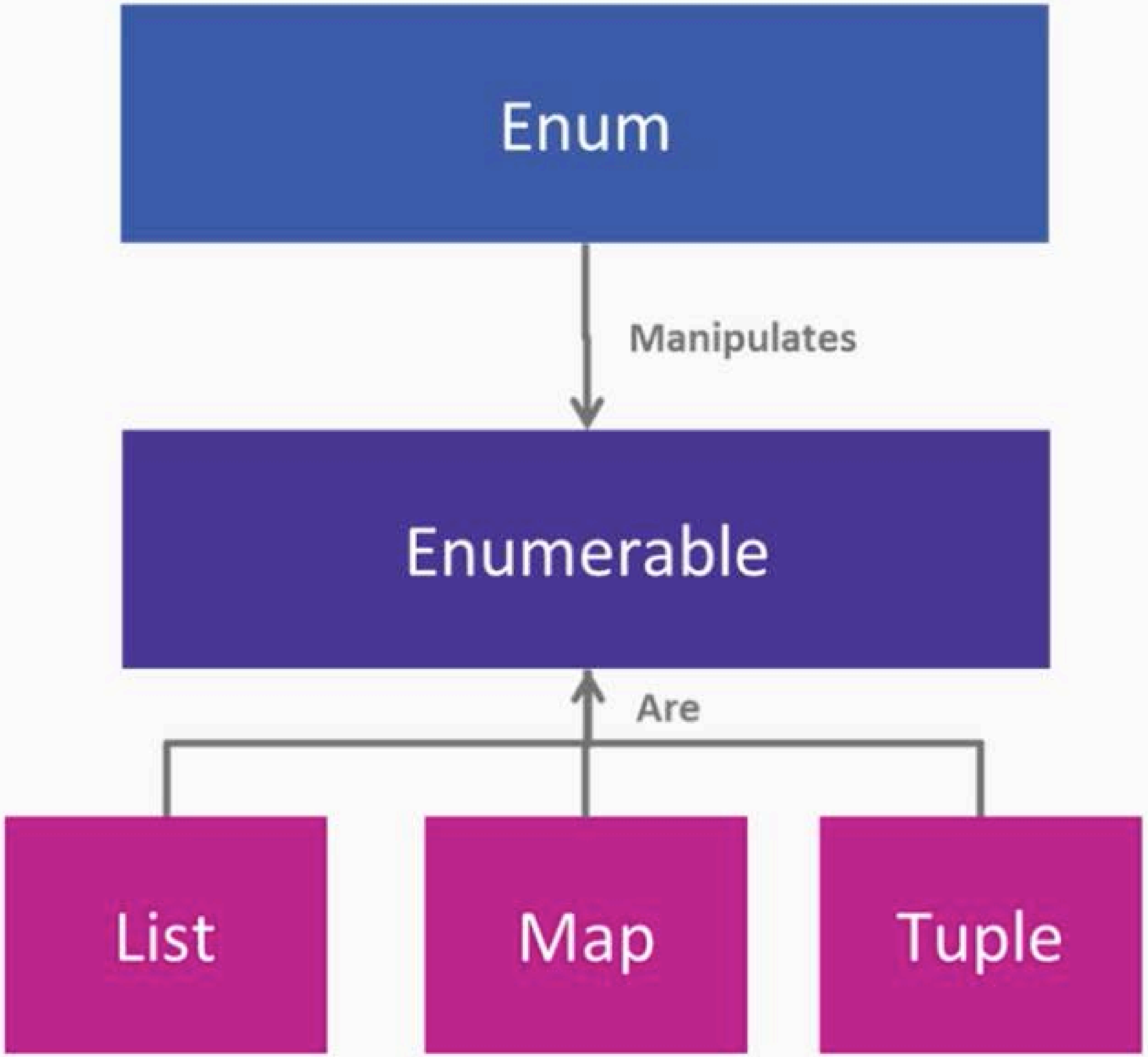
The Elixir way for type Polymorphism

Define a series of functions that must be implemented.

Allow implementing those functions for target data types.

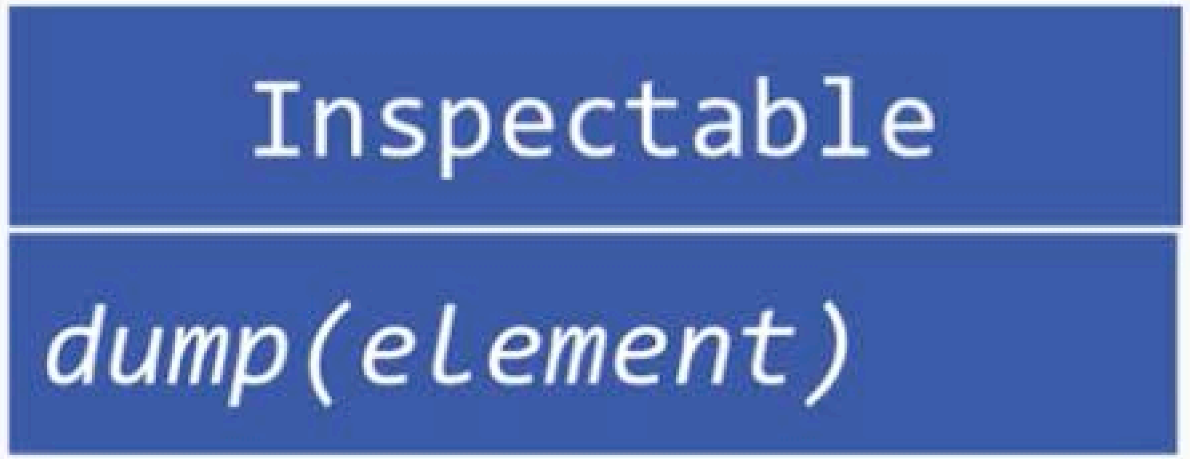
Enumerable

A Protocol that binds collections of different types through a common interface.



Inspectable

Exposes a dump function for logging type information.

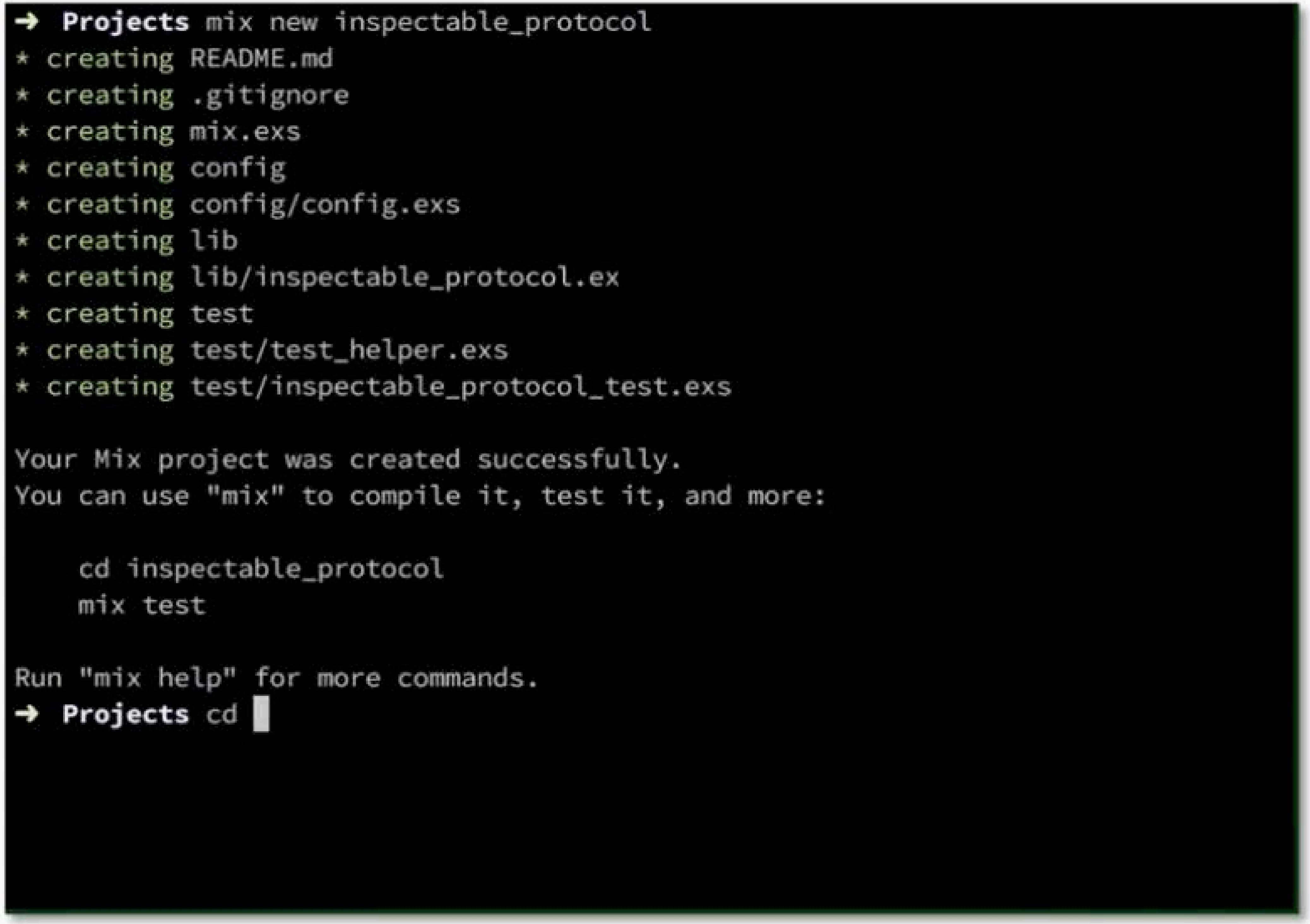


Inspectable

dump(element)

Defining Protocols

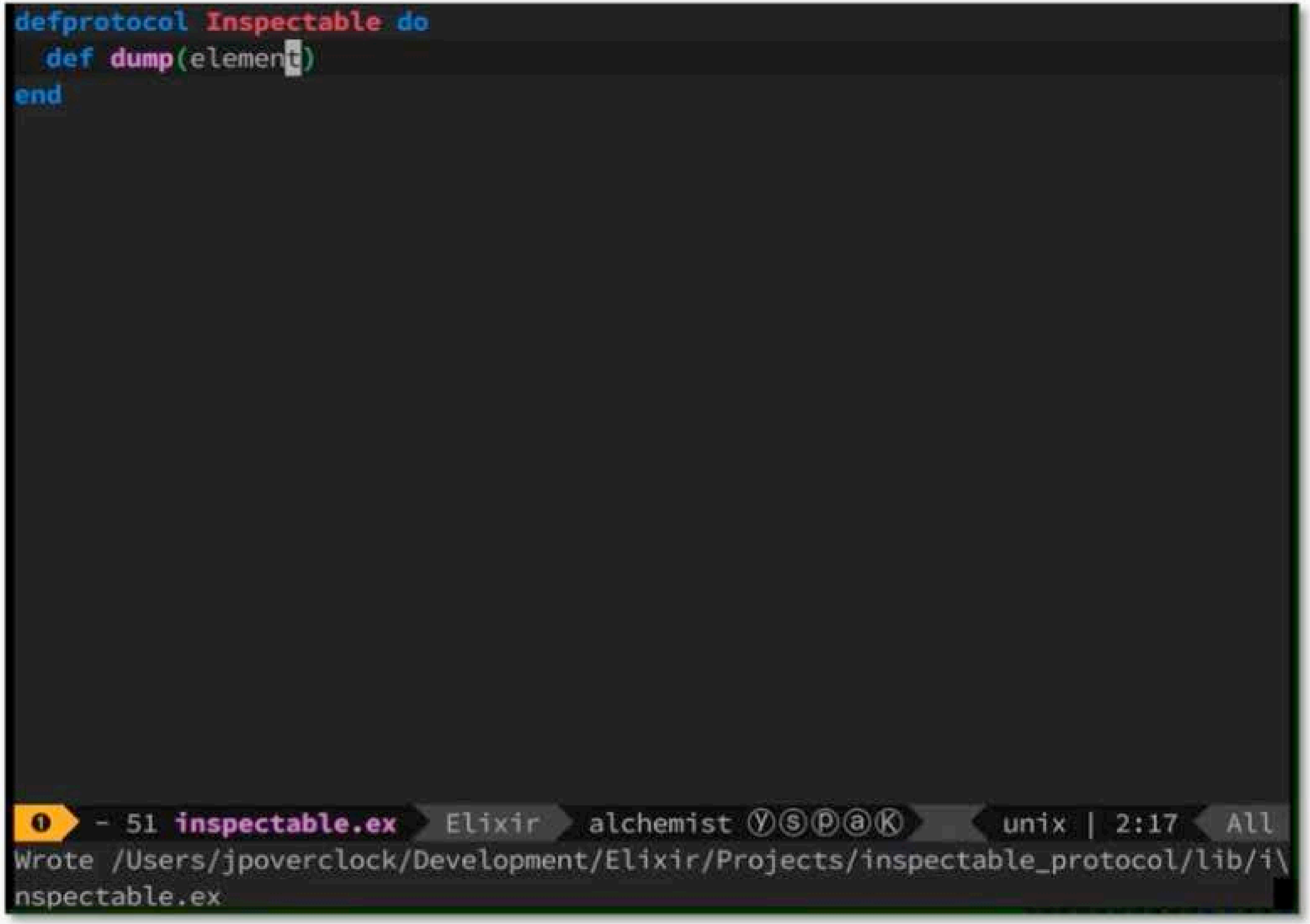
Define a Protocol – Inspectable - that exposes a dump function for logging type information.



```
➜ Projects mix new inspectable_protocol
* creating README.md
* creating .gitignore
* creating mix.exs
* creating config
* creating config/config.exs
* creating lib
* creating lib/inspectable_protocol.ex
* creating test
* creating test/test_helper.exs
* creating test/inspectable_protocol_test.exs

Your Mix project was created successfully.
You can use "mix" to compile it, test it, and more:

    cd inspectable_protocol
    mix test

Run "mix help" for more commands.
➜ Projects cd
```

Defining Protocols

Define a Protocol – Inspectable - that exposes a dump function for logging type information.

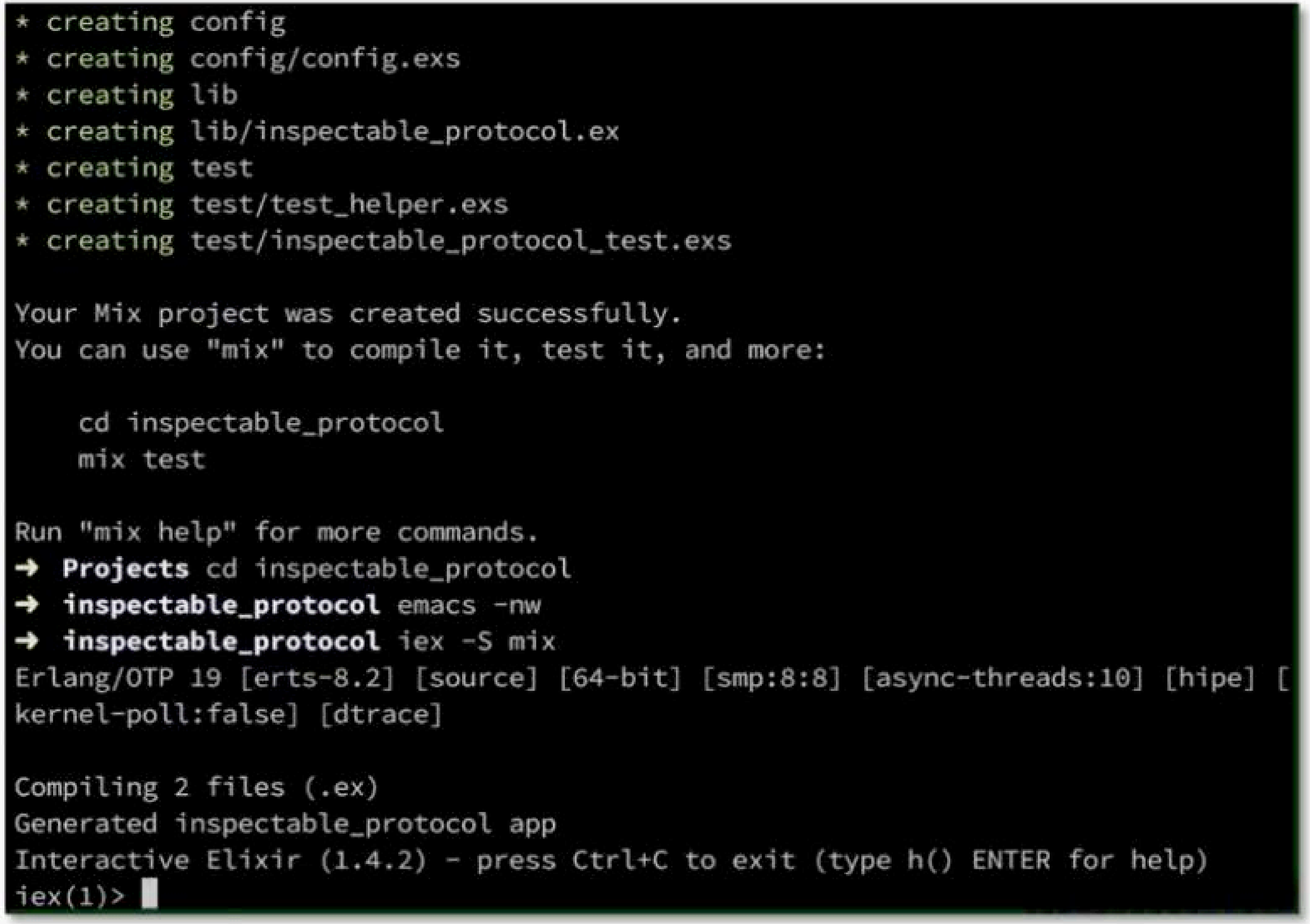


```
defprotocol Inspectable do
  def dump(element)
end
```

```
❶ - 51 inspectable.ex  Elixir  alchemist ⓨⓢⓟⓐⓚ  unix | 2:17  All
Wrote /Users/jpoverclock/Development/Elixir/Projects/inspectable_protocol/lib/i\
nspectable.ex
```

Defining Protocols

Define a Protocol – Inspectable - that exposes a dump function for logging type information.



```
* creating config
* creating config/config.exs
* creating lib
* creating lib/inspectable_protocol.ex
* creating test
* creating test/test_helper.exs
* creating test/inspectable_protocol_test.exs

Your Mix project was created successfully.
You can use "mix" to compile it, test it, and more:

    cd inspectable_protocol
    mix test

Run "mix help" for more commands.
➜ Projects cd inspectable_protocol
➜ inspectable_protocol emacs -nw
➜ inspectable_protocol iex -S mix
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Compiling 2 files (.ex)
Generated inspectable_protocol app
Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> 
```