Stephen

Elixir and Phoenix

How to get help -

Official Github Repo – github.com/StephenGrider – click Repositories tab – ElixirCode

Branch – each branch is for each section

Setting up elixir

* Local computer is capable of compiling and running elixir code
* Need to install elixir
* Goto goole and type “install elixir”
* <https://elixir-lang.org/install.html>
* **Erlang/OTP 26.1.1**

Installation steps I did

1. went to this link https://elixir-lang.org/install.html#windows  
   2. went to <https://community.chocolatey.org/> and install chocolatey using powershell  
   3. in an elevated powershell shell, typed choco install elixir

A screenshot of a computer

Description automatically generated

A black background with a black screen

Description automatically generated with medium confidence

Best way to learn a new prog language is to write code as soon as possible

1. Create an array of playing cards
2. Shuffle an array of playing cards
3. Create a ‘hand’ of cards
4. Given a deck and a single card, figure out if the card is in the deck
5. Save a collection of cards to a file on the local machine
6. Load a collection of cards from the local machine

Object oriented vs functional programming

How to generate a new project?

1. When you download elixir, you get a command line command out of the box called mix, which you can use to create project, compile projects, manage dependencies etc

A black screen with a black background

Description automatically generated

If “mix ” is not found, then ensure that erlang is pointing to ver 26 of erlang

A screenshot of a computer

Description automatically generated

A computer screen shot of white text

Description automatically generated

A screenshot of a computer

Description automatically generated

In the Cards module, we create a method called hello

Now lets see how to run this code

A computer screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A black screen with a black background

Description automatically generated

A black screen with a black border

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A black background with red lines

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated



Immutability in Elixir! – a new list is returned

A black screen with a black border

Description automatically generated

A black background with red text

Description automatically generated

A screen shot of a computer

Description automatically generated

A black screen with a black background

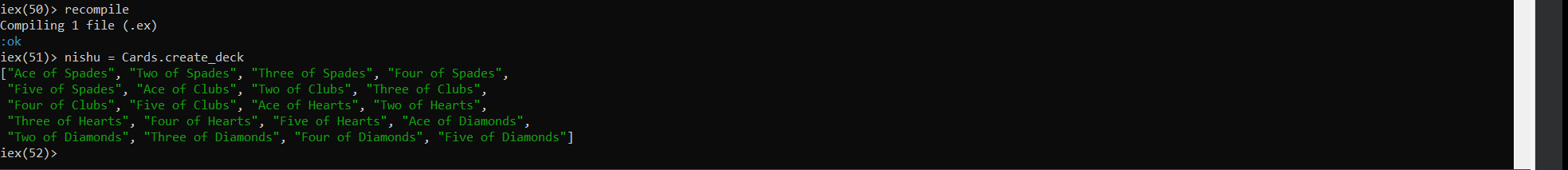
Description automatically generated

A black background with lights

Description automatically generated

A screenshot of a computer

Description automatically generated



A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

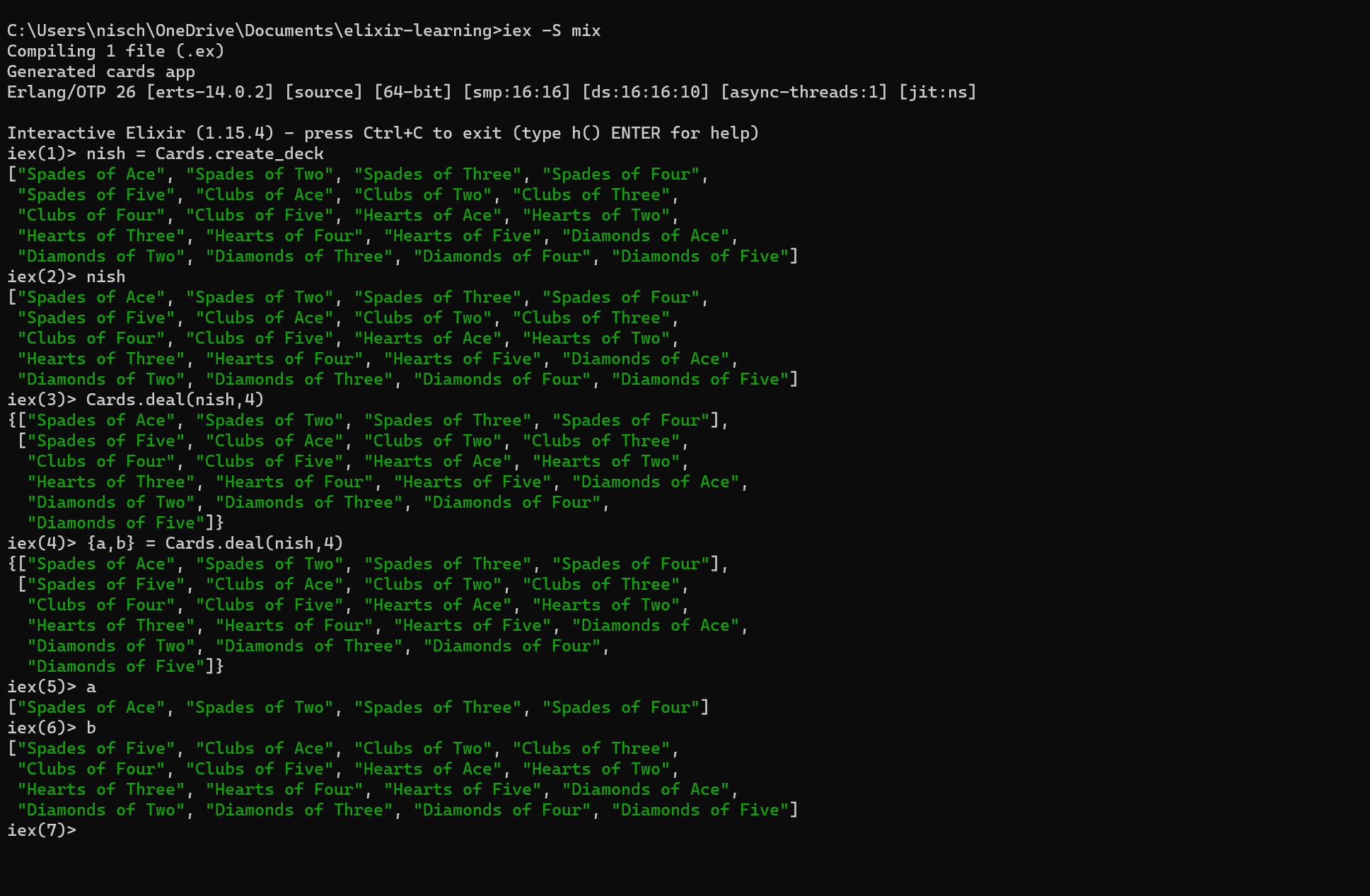
Curly branches is another data structure in elixir called a tuple

{

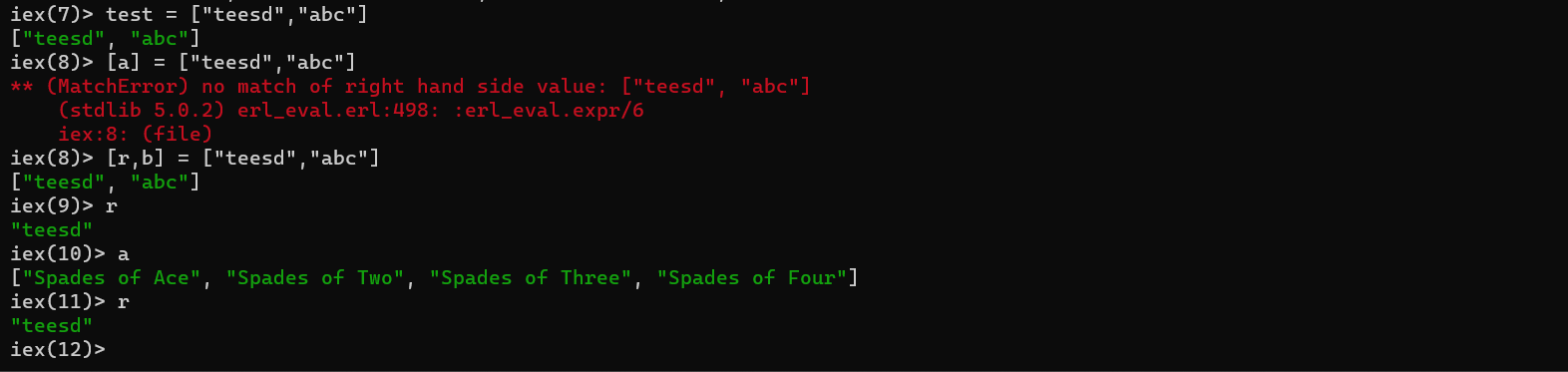
}

[] array is also a data structure in elixir

How to access the values inside of a touple? – next section



Above is called pattern matching



Code we write 🡪 gets fed into 🡪 elixir runtime 🡪 transpiled into 🡪 erlang 🡪 compiled and executed 🡪 BEAM

Erlang language is the one that compiles and executes the code in BEAM

Erlang syntax is difficult to understand – elixir reasonably straight forward syntax – so elixir is a dialect of erlang – so we use elixir to get away from the annoying parts of erlang

Elixirs sole purpose is to give us a better interface to erlang

BEAM – abstract machine – virtual machine where erlang code gets executed

So the BEAM is like a jvm

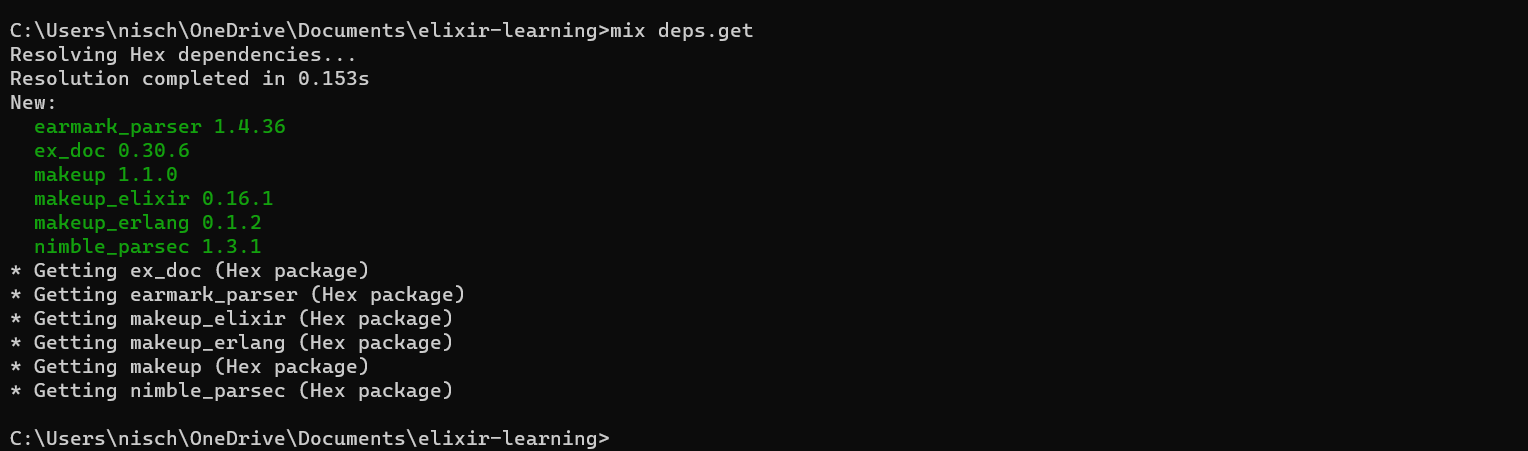
There are small bits of functionality that elixir cannot do – so we must call erlang

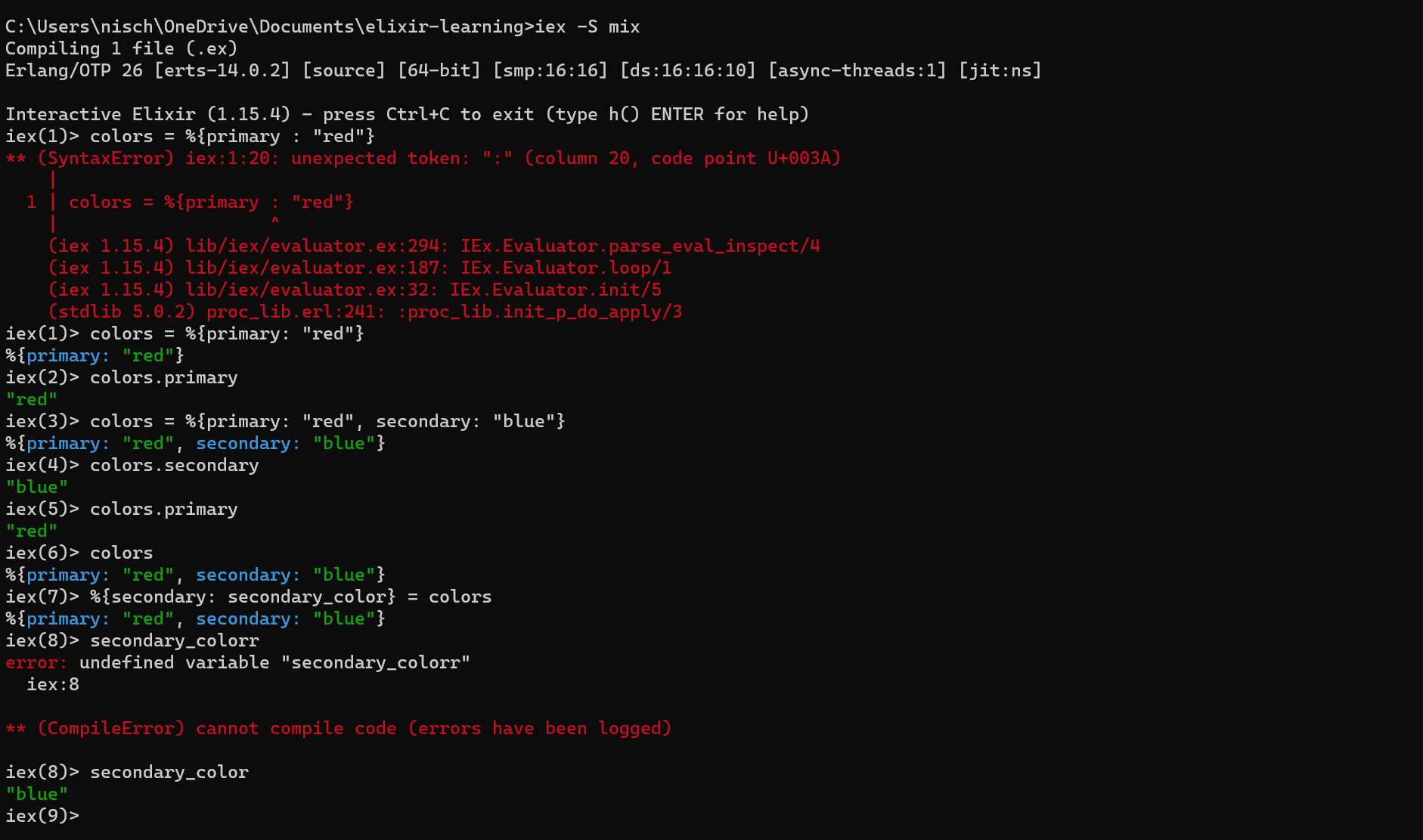
Eg. Filesystem

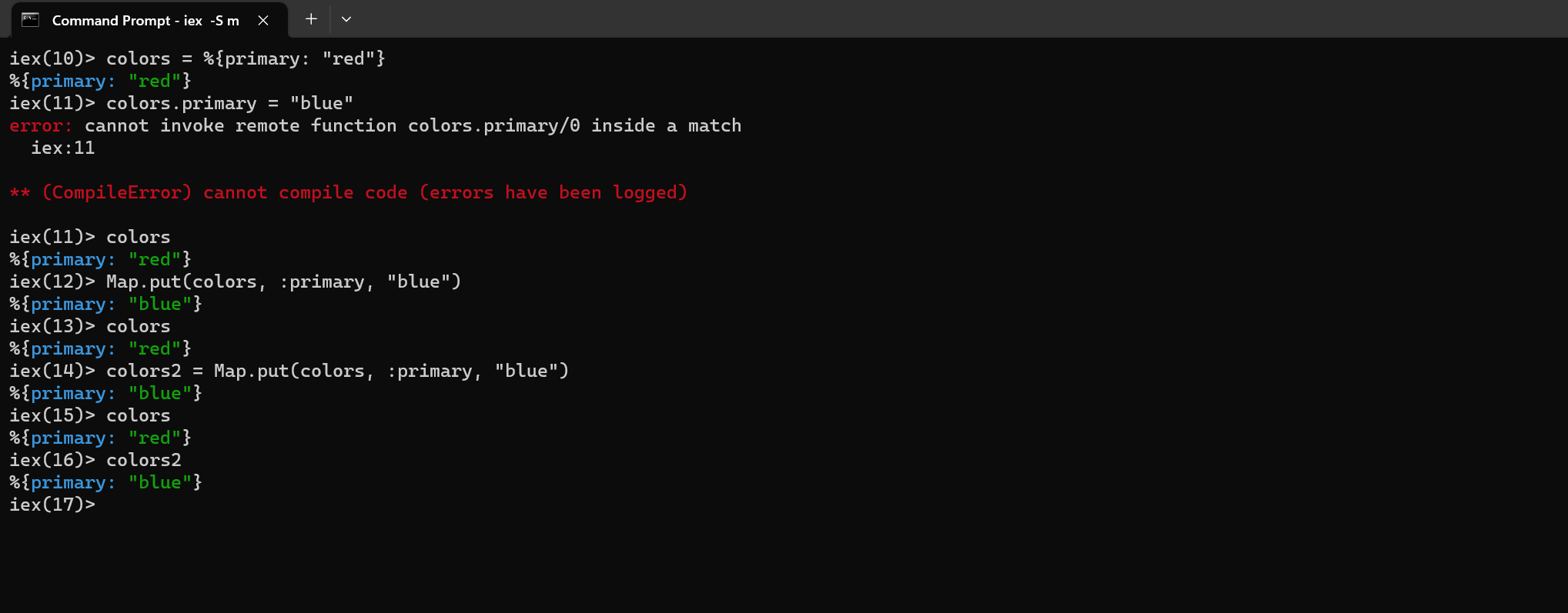
A screen shot of a computer

Description automatically generated

If you want to tell the compiler that yes, I have a variable that I am not gonna use and I know it is gonna be unused, just add a underscore … i.e. \_reason , and there wont be warnings in the compiler anymore







A black screen with a black border

Description automatically generated

