Streaming Computation

Section 1

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

- Introduction
- Enumerables
- Streams
- Flow

Enumerables

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

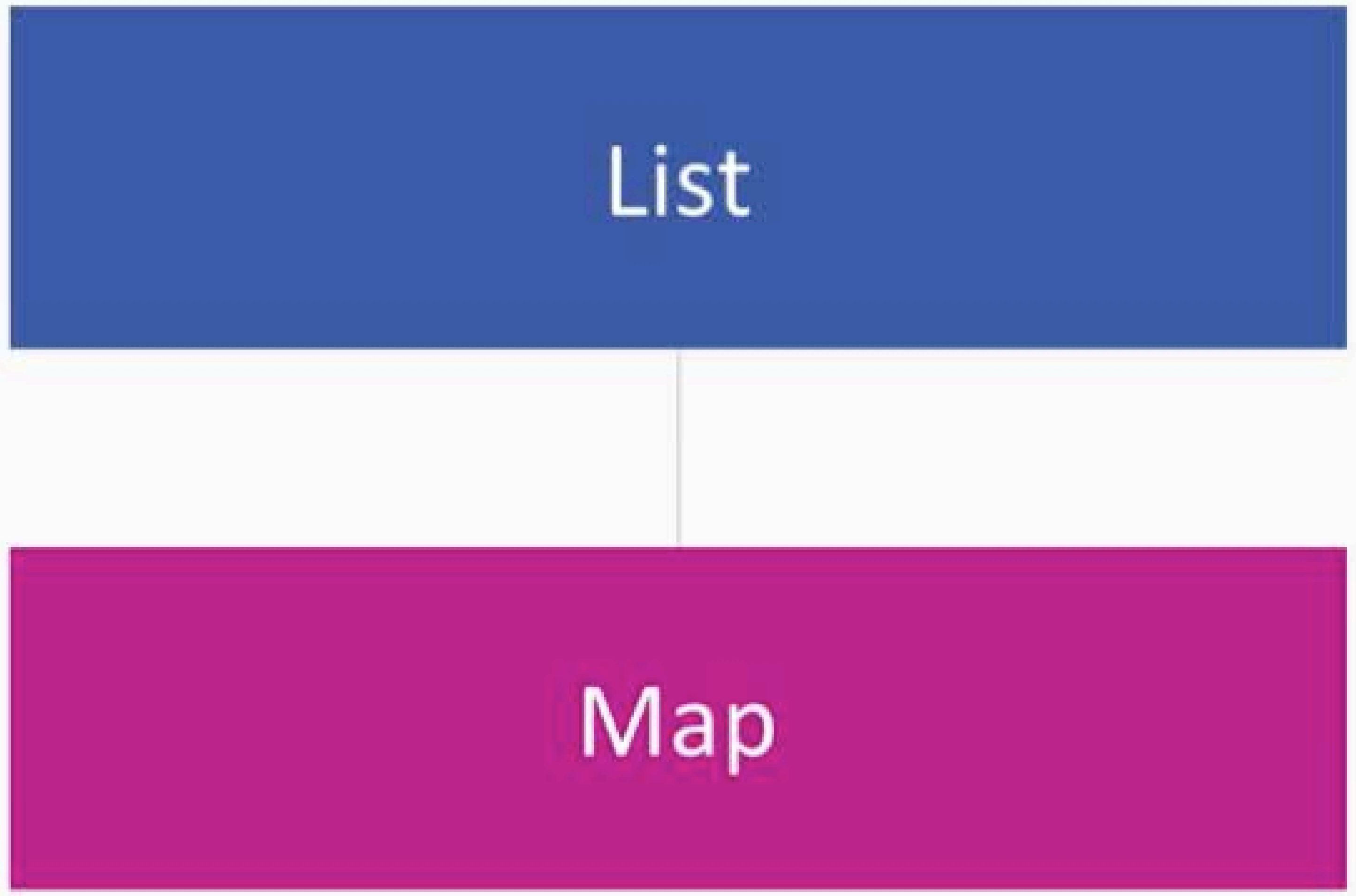
- Collection types and Enumerable
- Using Enumerable types interchangeably
- Transforming Enumerables

Collection Types

Data types that allow the composition of other data types.

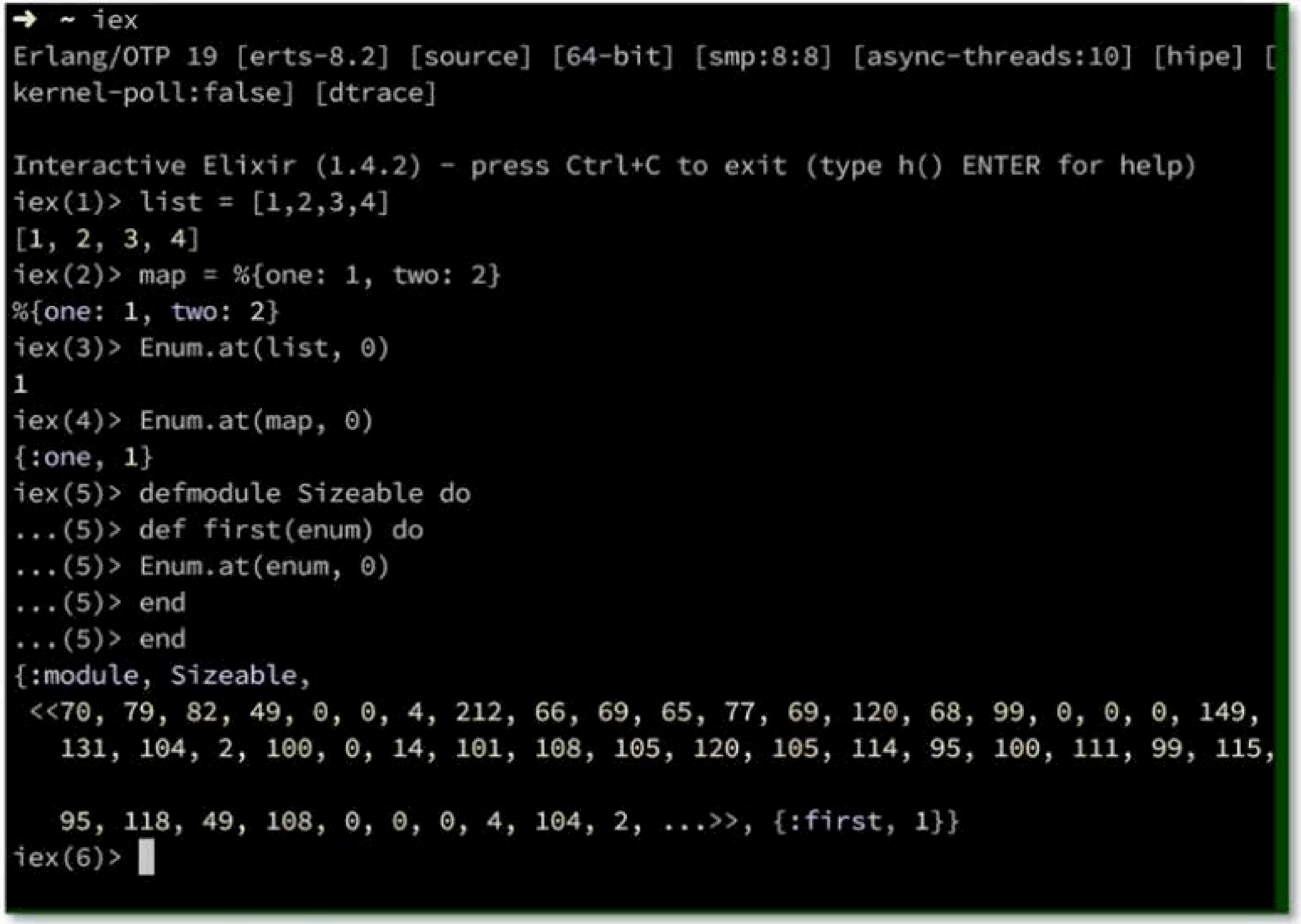
Collection Types

Data types that allow the composition of other data types.



Collection Types

Data types that allow the composition of other data types.

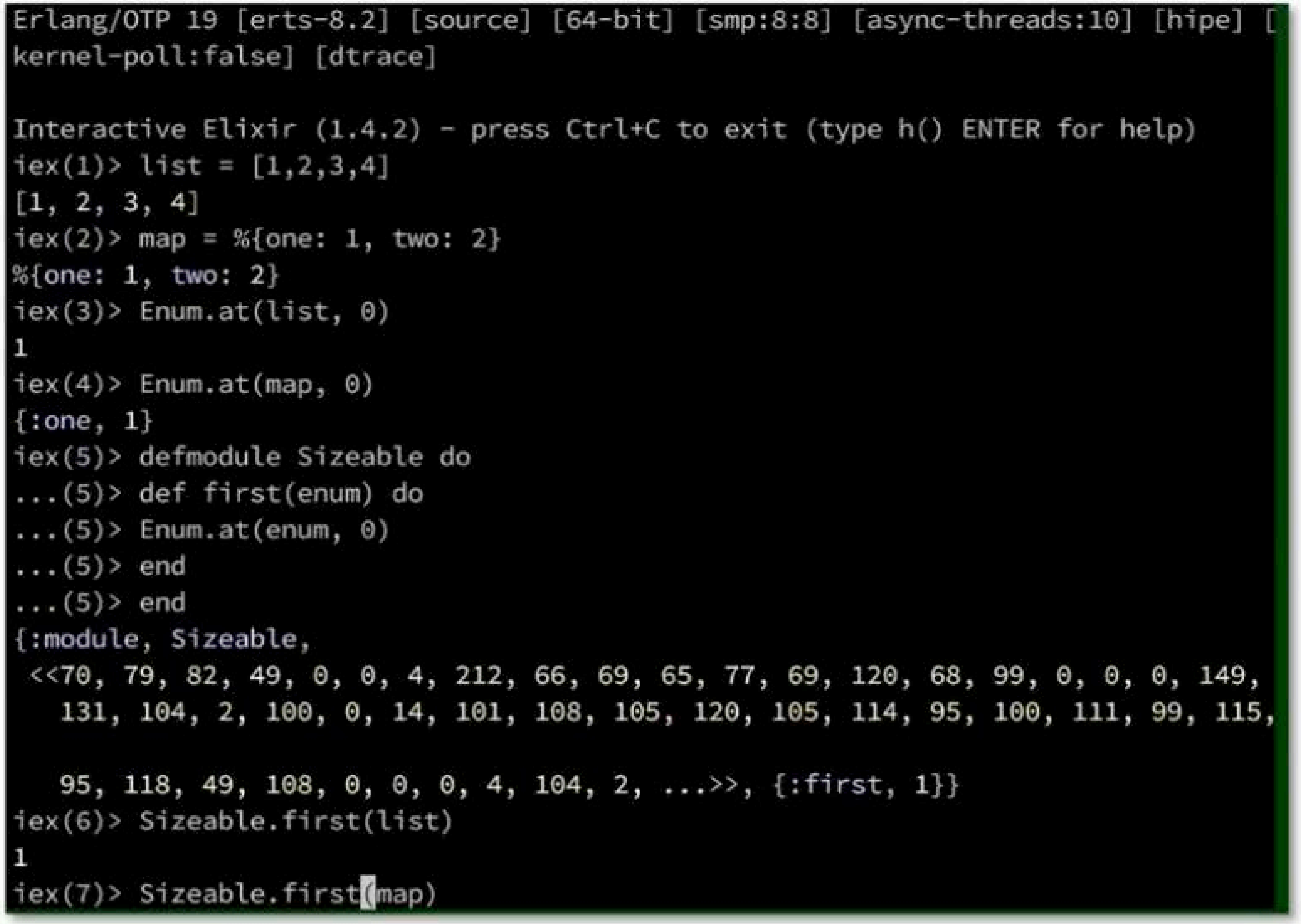


```
➜ ~ iex
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> list = [1,2,3,4]
[1, 2, 3, 4]
iex(2)> map = %{one: 1, two: 2}
%{one: 1, two: 2}
iex(3)> Enum.at(list, 0)
1
iex(4)> Enum.at(map, 0)
{:one, 1}
iex(5)> defmodule Sizeable do
...(5)> def first(enum) do
...(5)> Enum.at(enum, 0)
...(5)> end
...(5)> end
{:module, Sizeable,
 <<70, 79, 82, 49, 0, 0, 4, 212, 66, 69, 65, 77, 69, 120, 68, 99, 0, 0, 0, 149,
   131, 104, 2, 100, 0, 14, 101, 108, 105, 120, 105, 114, 95, 100, 111, 99, 115,

   95, 118, 49, 108, 0, 0, 0, 4, 104, 2, ...>>, {:first, 1}}
iex(6)> 
```

Packt>

Collection Types

Data types that allow the composition of other data types.



```
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> list = [1,2,3,4]
[1, 2, 3, 4]
iex(2)> map = %{one: 1, two: 2}
%{one: 1, two: 2}
iex(3)> Enum.at(list, 0)
1
iex(4)> Enum.at(map, 0)
{:one, 1}
iex(5)> defmodule Sizeable do
...(5)> def first(enum) do
...(5)> Enum.at(enum, 0)
...(5)> end
...(5)> end
{:module, Sizeable,
 <<70, 79, 82, 49, 0, 0, 4, 212, 66, 69, 65, 77, 69, 120, 68, 99, 0, 0, 0, 149,
   131, 104, 2, 100, 0, 14, 101, 108, 105, 120, 105, 114, 95, 100, 111, 99, 115,

   95, 118, 49, 108, 0, 0, 0, 4, 104, 2, ...>>, {:first, 1}}
iex(6)> Sizeable.first(list)
1
iex(7)> Sizeable.first(map)
```

Packt>

Enumerable

A protocol that binds Enumerable collection types

Enumerable

A protocol that binds Enumerable collection types

- Collections have a variable number of elements
- One can verify if an element belongs to a collection
- Collections can be reduced by means of a function and an accumulator

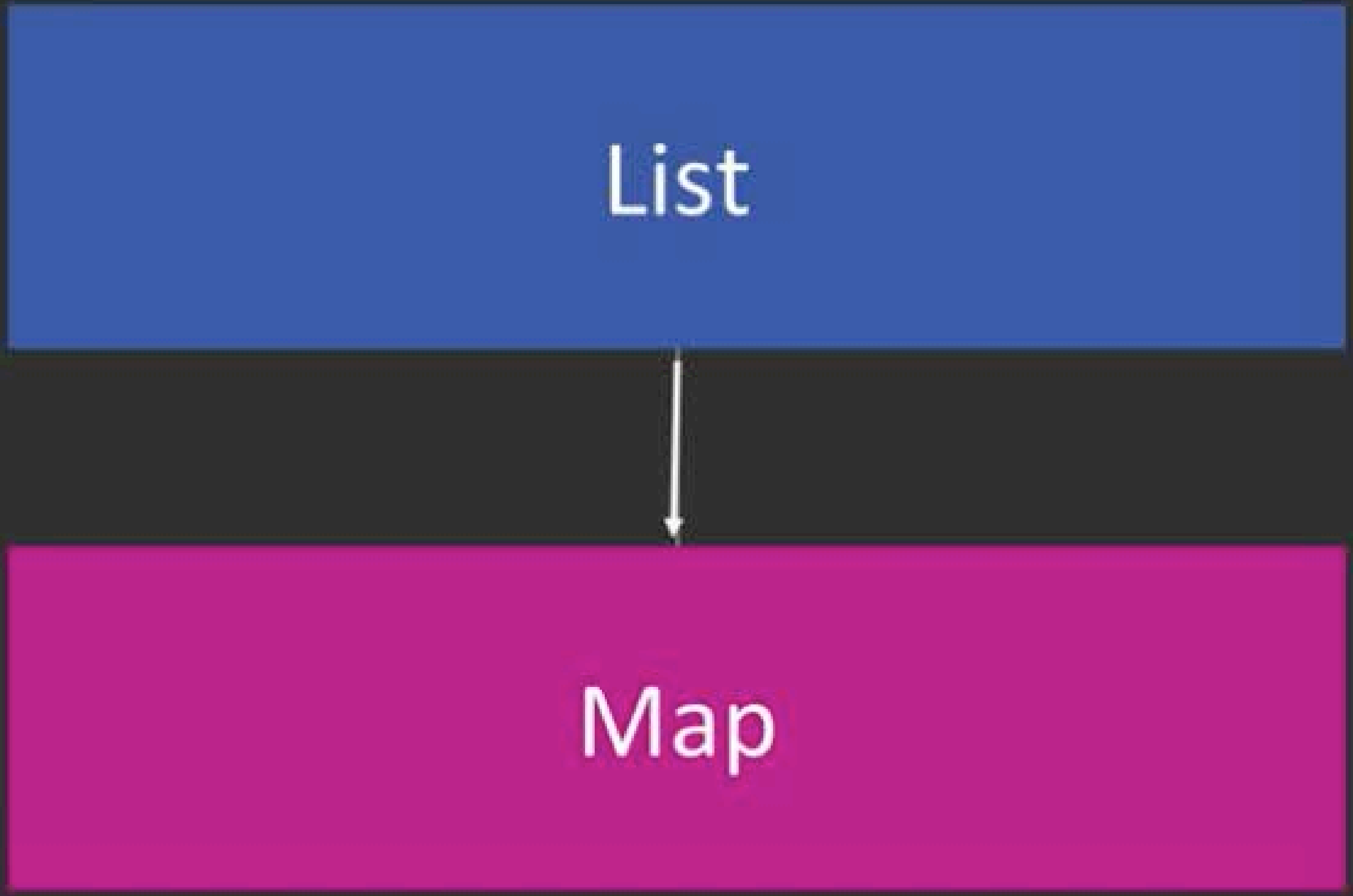
Enumerable

Enum

- Collections have a variable number of elements
- One can verify if an element belongs to a collection
- Collections can be reduced by means of a function and an accumulator

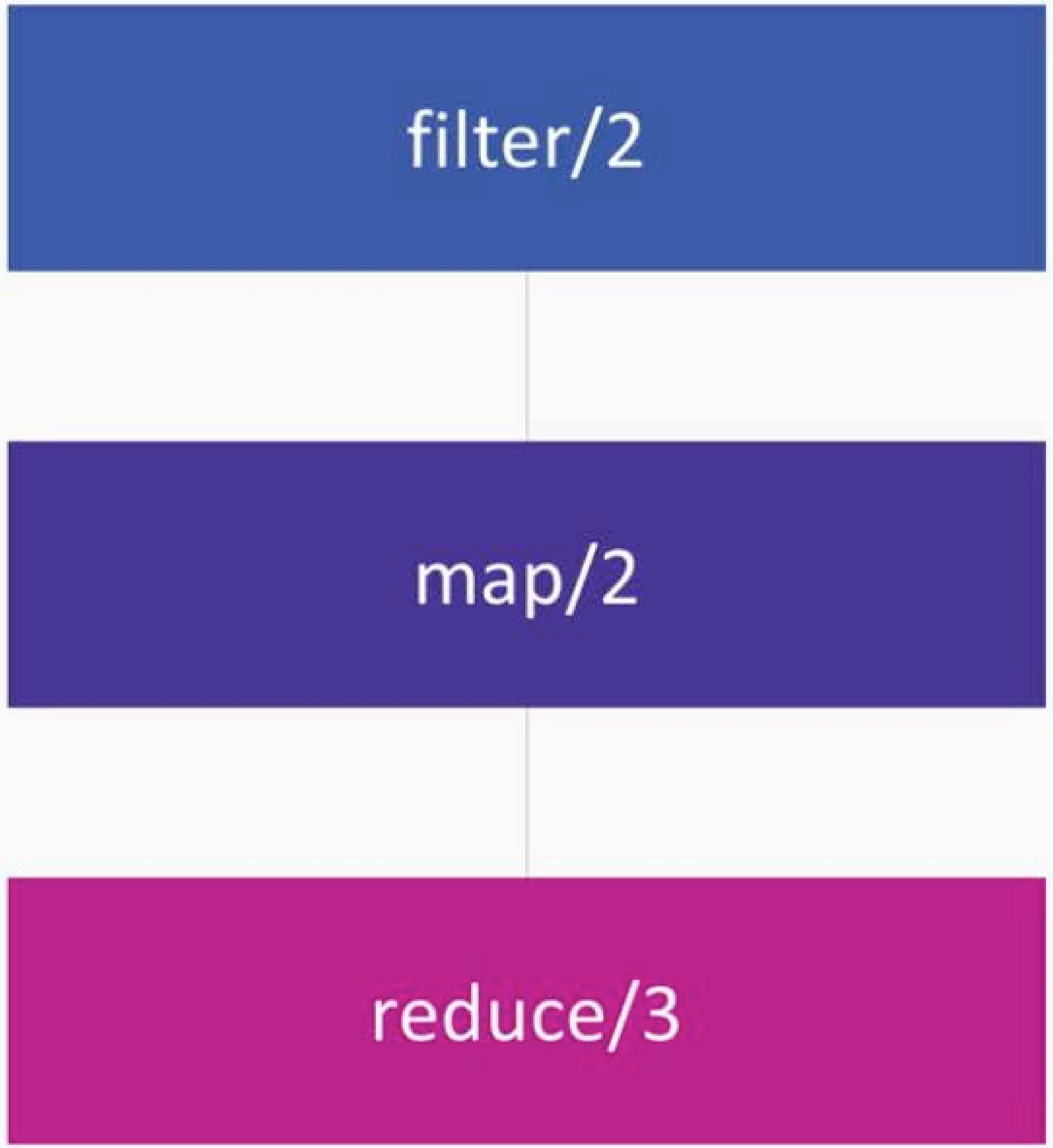
Enumerable

Enum



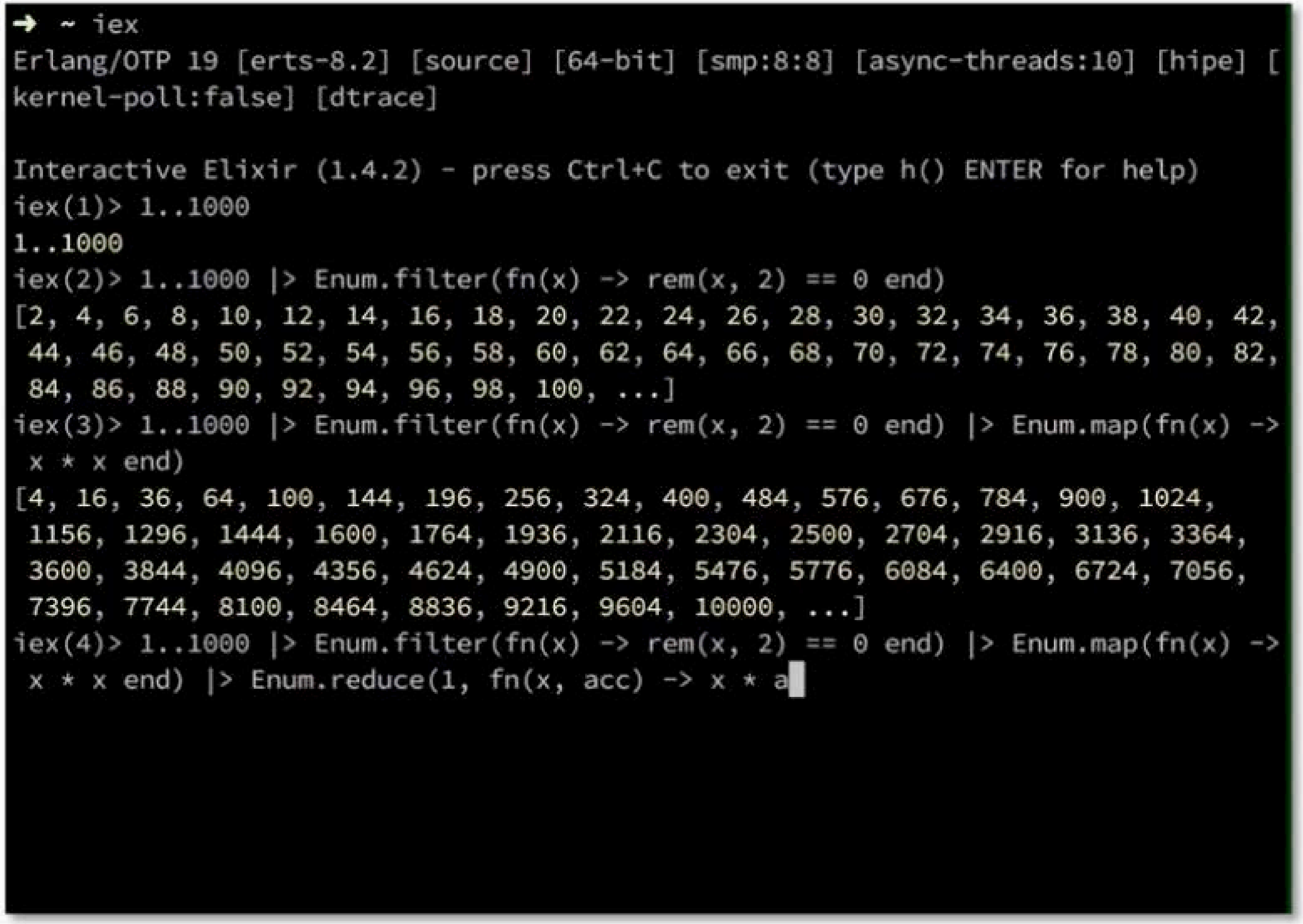
Transformations

Transforming collections by filtering, mapping, and reducing their elements.



Transformations

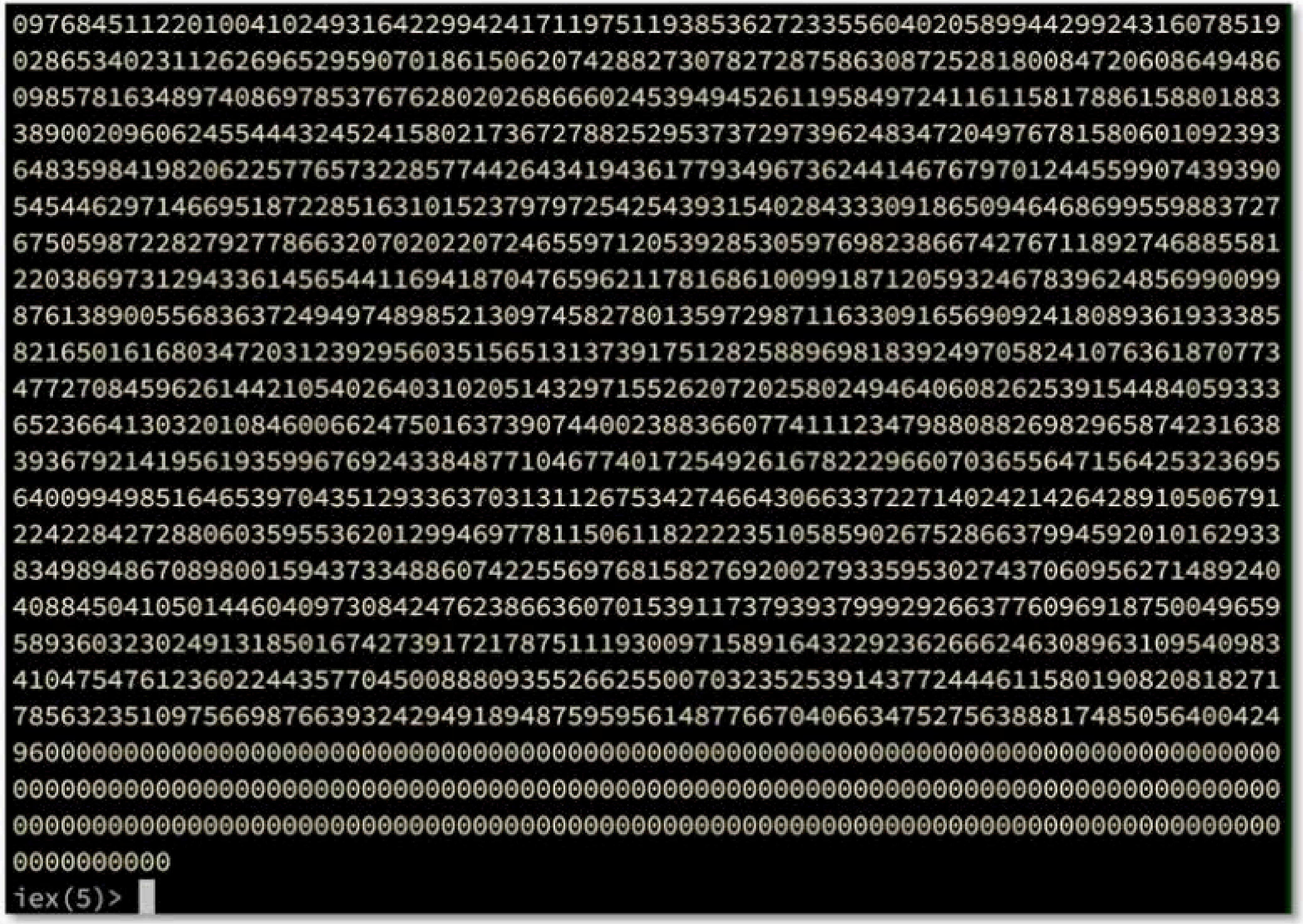
Transforming collections by filtering, mapping, and reducing their elements.



```
➜ ~ iex
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> 1..1000
1..1000
iex(2)> 1..1000 |> Enum.filter(fn(x) -> rem(x, 2) == 0 end)
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82,
 84, 86, 88, 90, 92, 94, 96, 98, 100, ...]
iex(3)> 1..1000 |> Enum.filter(fn(x) -> rem(x, 2) == 0 end) |> Enum.map(fn(x) ->
 x * x end)
[4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676, 784, 900, 1024,
 1156, 1296, 1444, 1600, 1764, 1936, 2116, 2304, 2500, 2704, 2916, 3136, 3364,
 3600, 3844, 4096, 4356, 4624, 4900, 5184, 5476, 5776, 6084, 6400, 6724, 7056,
 7396, 7744, 8100, 8464, 8836, 9216, 9604, 10000, ...]
iex(4)> 1..1000 |> Enum.filter(fn(x) -> rem(x, 2) == 0 end) |> Enum.map(fn(x) ->
 x * x end) |> Enum.reduce(1, fn(x, acc) -> x * a
```

Transformations

Transforming collections by filtering, mapping, and reducing their elements.



```
09768451122010041024931642299424171197511938536272335560402058994429924316078519
02865340231126269652959070186150620742882730782728758630872528180084720608649486
09857816348974086978537676280202686660245394945261195849724116115817886158801883
38900209606245544432452415802173672788252953737297396248347204976781580601092393
64835984198206225776573228577442643419436177934967362441467679701244559907439390
54544629714669518722851631015237979725425439315402843330918650946468699559883727
67505987228279277866320702022072465597120539285305976982386674276711892746885581
22038697312943361456544116941870476596211781686100991871205932467839624856990099
87613890055683637249497489852130974582780135972987116330916569092418089361933385
82165016168034720312392956035156513137391751282588969818392497058241076361870773
47727084596261442105402640310205143297155262072025802494640608262539154484059333
65236641303201084600662475016373907440023883660774111234798808826982965874231638
39367921419561935996769243384877104677401725492616782229660703655647156425323695
64009949851646539704351293363703131126753427466430663372271402421426428910506791
22422842728806035955362012994697781150611822223510585902675286637994592010162933
83498948670898001594373348860742255697681582769200279335953027437060956271489240
40884504105014460409730842476238663607015391173793937999292663776096918750049659
58936032302491318501674273917217875111930097158916432292362666246308963109540983
41047547612360224435770450088809355266255007032352539143772444611580190820818271
78563235109756698766393242949189487595956148776670406634752756388817485056400424
96000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000
iex(5)>
```

Project Euler

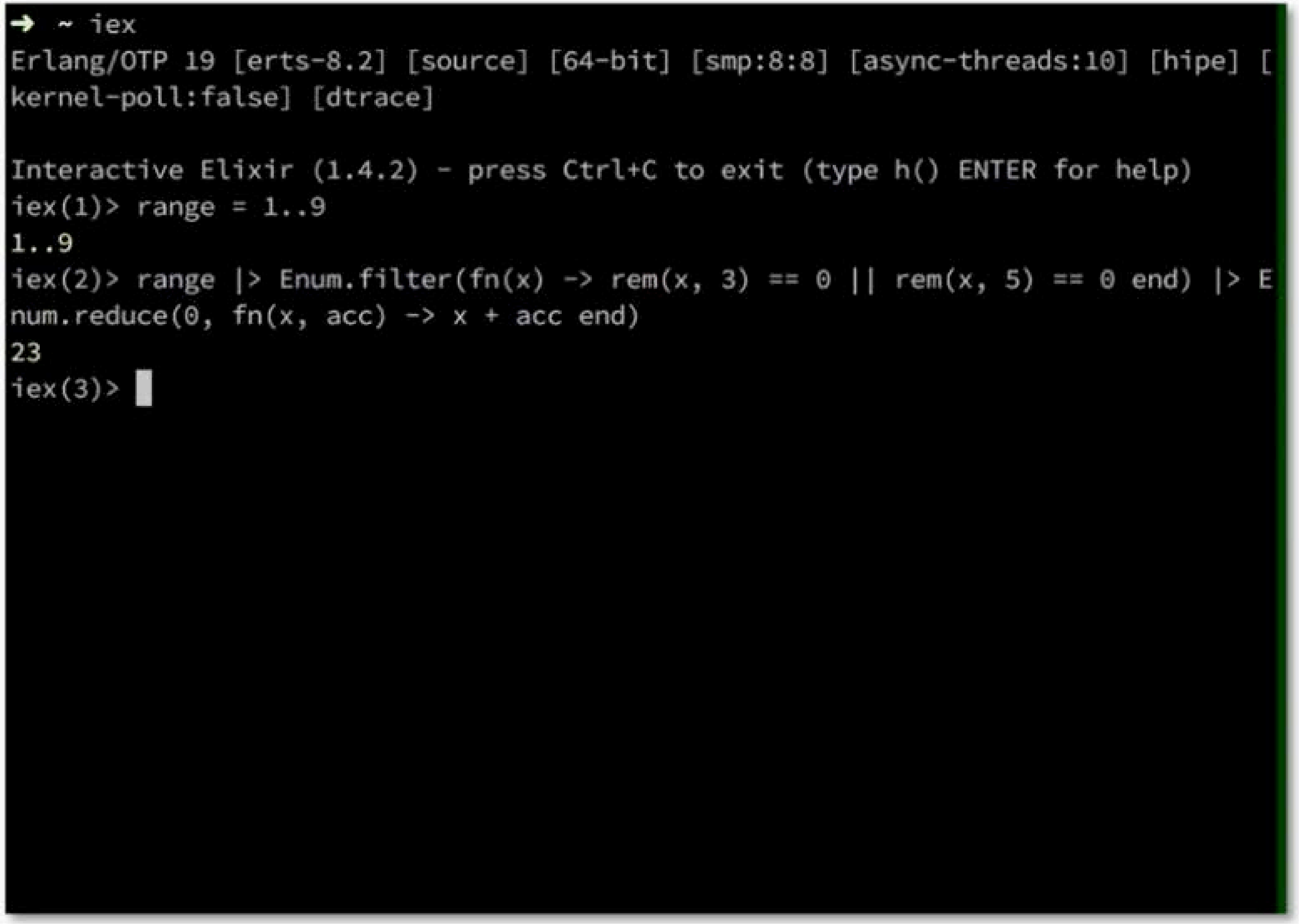
- Exercise 1:
  - If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6, and 9
  - The sum of these multiples is 23
  - Find the sum of all the multiples of 3 or 5 below 1000

https://projecteuler.net/

Project Euler — 1

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6, and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

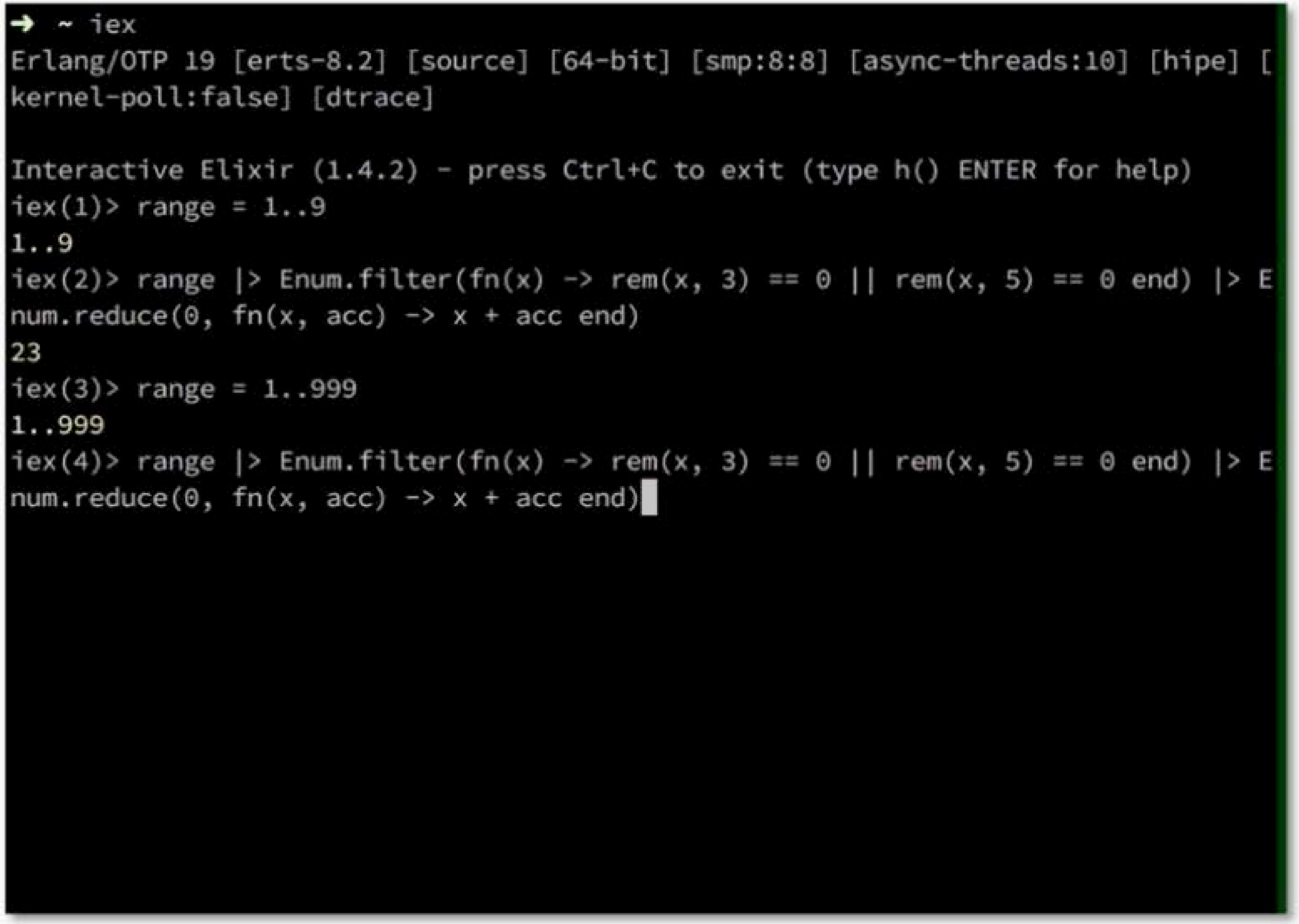


```
➜  ~ iex
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> range = 1..9
1..9
iex(2)> range |> Enum.filter(fn(x) -> rem(x, 3) == 0 || rem(x, 5) == 0 end) |> E
num.reduce(0, fn(x, acc) -> x + acc end)
23
iex(3)> 
```

Project Euler — 1

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6, and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.



```
➜  ~ iex
Erlang/OTP 19 [erts-8.2] [source] [64-bit] [smp:8:8] [async-threads:10] [hipe] [
kernel-poll:false] [dtrace]

Interactive Elixir (1.4.2) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> range = 1..9
1..9
iex(2)> range |> Enum.filter(fn(x) -> rem(x, 3) == 0 || rem(x, 5) == 0 end) |> E
num.reduce(0, fn(x, acc) -> x + acc end)
23
iex(3)> range = 1..999
1..999
iex(4)> range |> Enum.filter(fn(x) -> rem(x, 3) == 0 || rem(x, 5) == 0 end) |> E
num.reduce(0, fn(x, acc) -> x + acc end)
```