Getting Started with Elixir

João Gonçalves

Section 2

Basic Types and Operators

www.packtpub.com

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

- Literal types
- Basic operators
- Collection types
- Functions to manipulate collections

Getting Started with Elixir

João Gonçalves

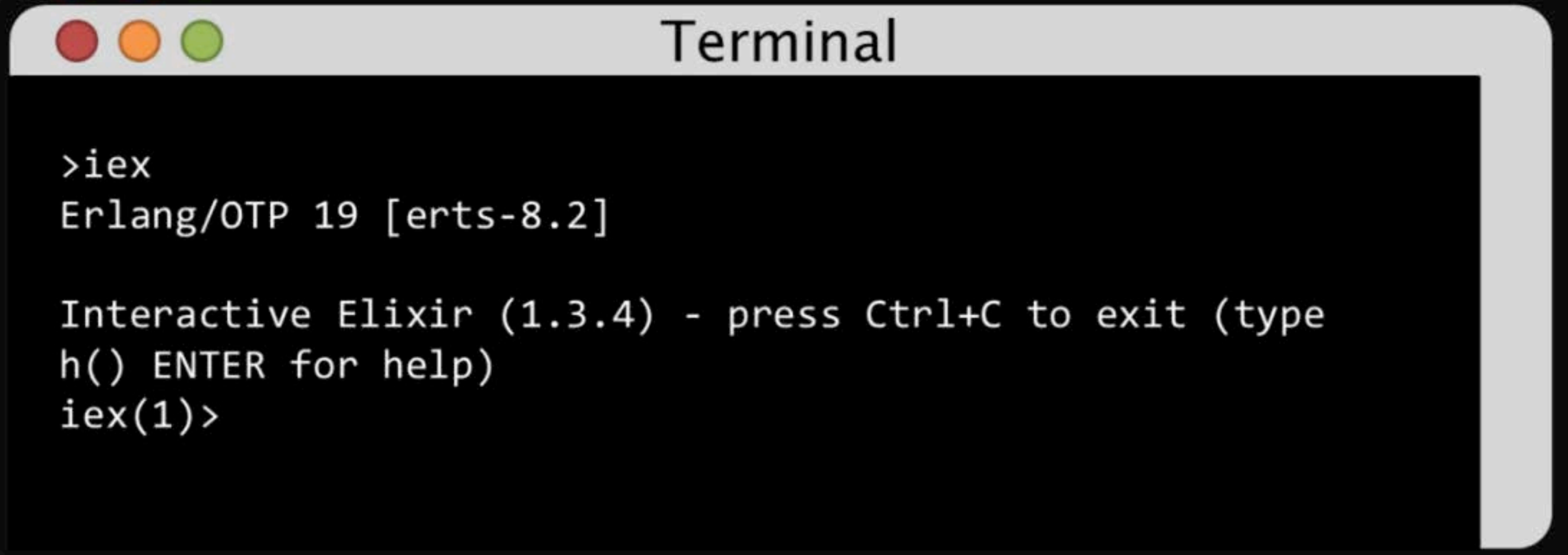
Video 2.1

Literals and Operators

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

- Observing the built-in literal types
- Exploring the basic operators

Before We Start



Numbers and Arithmetic Operators

- Integer
  - `1`
  - `-30`
- Floating point
  - `1.54932`
  - `1.5e-15`

Numbers and Arithmetic Operators

- Integer Bases

Base	Example	Decimal
Hexadecimal (16)	`0xFE`	254
Octal (8)	`0o773`	507
Binary (2)	`0b1110`	14

Numbers and Arithmetic Operators

- Integer – Separator
  - `43312209`
  - `43_312_209`

Numbers and Arithmetic Operators

- Precision

554385734895347854395348573457642348542367845236742354672345623745236745236745326745236745762345631245762345263

Numbers and Arithmetic Operators

- Operators

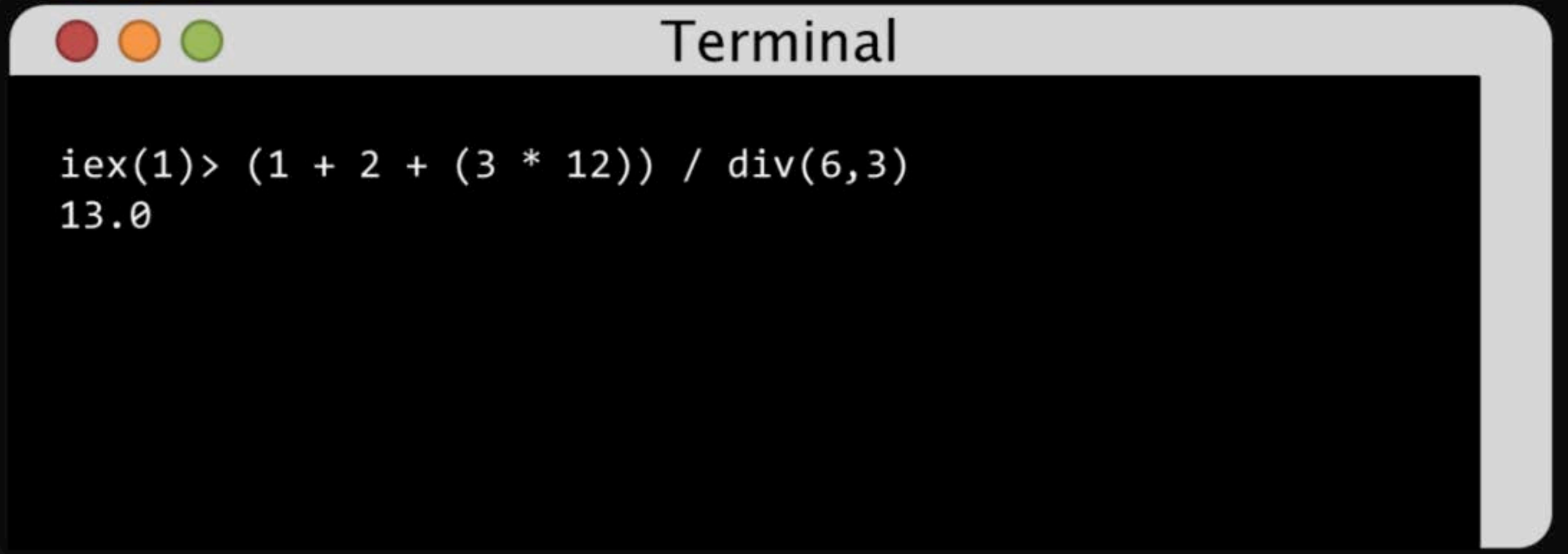
Operator	Example	Result
Addition (+)	`1.5 + 9`	`10.5`
Subtraction (-)	`1 - 2`	`-1`
Multiplication (*)	`9 * 8`	`72`
Division (/)	`10 / 4`	`2.5`

Numbers and Arithmetic Operators

- Integer Division Functions

Operator	Example	Result
Division (div)	`div(9, 7)`	1
Modulo/Remainder (rem)	`rem(9, 7)`	2

Booleans



Booleans

- True
  - Everything is truthy
- False
  - nil is falsey

Booleans

`and, or, not` → `strict`

`&&, ||, !` → `non-strict`

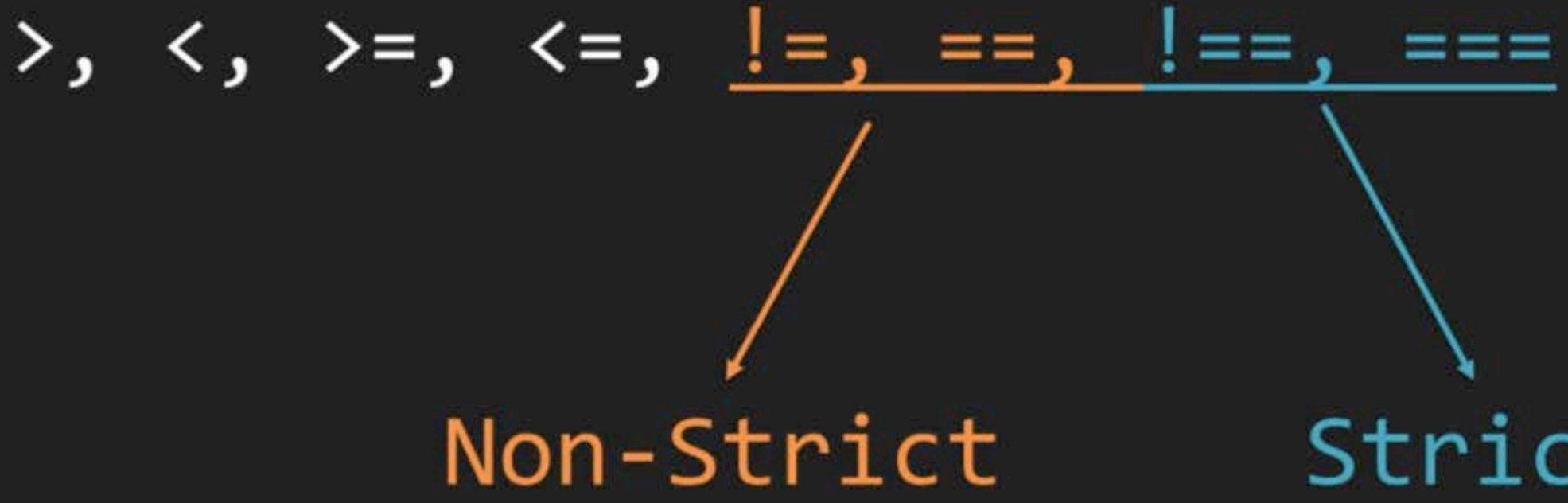
Booleans

Booleans – Comparison Operators

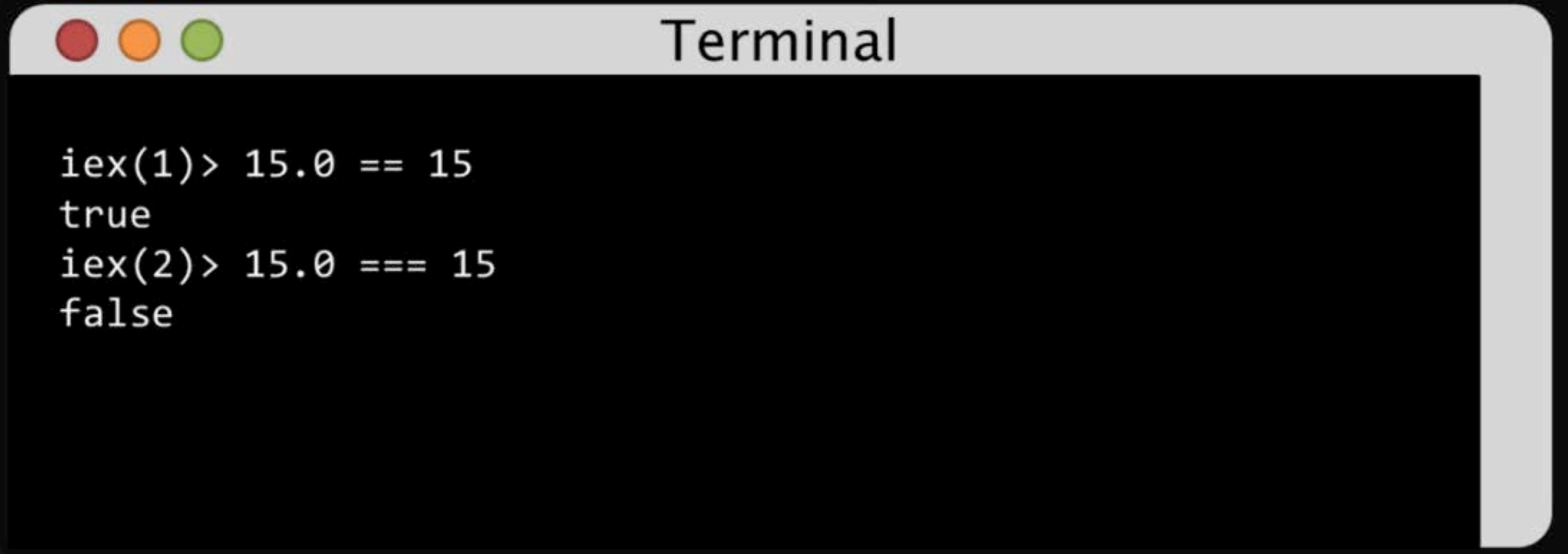
`>`, `<`, `>=`, `<=`, `!=`, `!==`, `==`, `===`

true
false

Booleans – Comparison Operators



Booleans – Comparison Operators



Booleans – Comparison Operators

- Not just for numbers
  - `number < atom < reference < function < port < pid < tuple < map < list < bitstring`
- Multiple types can be compared with one another by this order

Strings

“Hello, World!”

“γειά σου κόσμ”

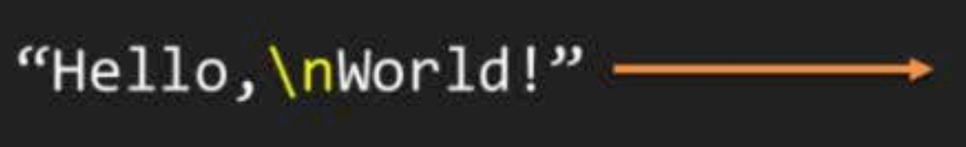
Strings

"Hello, World!"

"γειά σου κόσμ"

UTF-8

Strings



Strings

```
"I am #{2 * 15} years old."
```

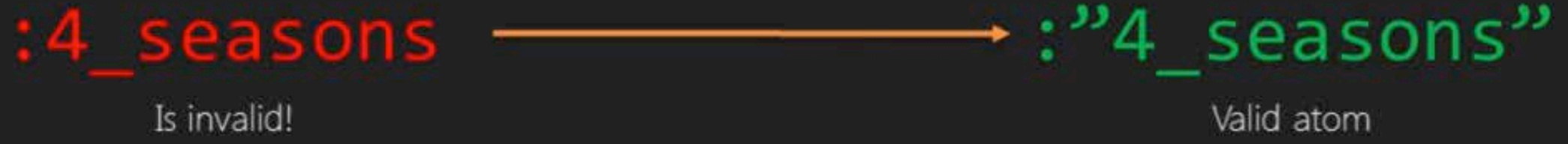
```
I am 30 years old.
```

Strings – Some fun(ctions)

Atoms

- A constant whose name is its value
  - `:hello`
  - `:Hey_you`
  - `:">_<"`

Atoms



Is invalid!

Valid atom