***Practice***

ex(1)> "hello world!"

"hello world!"

iex(2)> 24\*60

1440

iex(3)> 1440 \* 60

86400

iex(4)> 24 \*60\*60

86400

iex(5)> (4+8+15+16+23+42)/6

18.0

iex(6)> sentence = " A really long and complex sentence we'd rather not repeat."

" A really long and complex sentence we'd rather not repeat."

iex(7)> score = 2/5 \*100

40.0

iex(8)> x= " not for multiplication "

" not for multiplication "

iex(9)> sentence

" A really long and complex sentence we'd rather not repeat."

iex(10)> score

40.0

iex(11)> x

" not for multiplication "

iex(12)> place = "World"

"World"

iex(13)> "hello! #{place}!"

"hello! World!"

iex(14)> seconds = 86400

86400

List

iex(15)> shopping\_list = ["fish","ham","eggs","bread"]

["fish", "ham", "eggs", "bread"]

iex(17)> those\_who\_are\_assembled =[["Izzay","30ish","Female"],["The Author","30ish","Male"],]

[["Izzay", "30ish", "Female"], ["The Author", "30ish", "Male"]]

Maps

iex(18)> person = %{"name" => "Roberto", "age" =>"56","gender" =>"male"}

%{"age" => "56", "gender" => "male", "name" => "Roberto"}

iex(19)> person["name"]

"Roberto"

List of Maps

iex(22)> those\_who\_are\_assembled = [%{"name" => "Izzy", "age" => "30ish", "gender" => "Female"},%{"name" => "The Author", "age" => "30ish", "gender" => "Male"},%{"name" => "The Reader", "age" => "Unknowable", "gender" => "Unknowable"},]

[

%{"age" => "30ish", "gender" => "Female", "name" => "Izzy"},

%{"age" => "30ish", "gender" => "Male", "name" => "The Author"},

%{"age" => "Unknowable", "gender" => "Unknowable", "name" => "The Reader"}

]

iex(23)> person[:name]

nil

iex(24)> person["name"]

"Roberto"

iex(25)> person.name

\*\* (KeyError) key :name not found in: %{"age" => "56", "gender" => "male", "name" => "Roberto"}

iex:25: (file)

iex(25)> person = %{

...(25)> :name => "Izzy",

...(25)> :age => "30ish",

...(25)> :gender => "Female"

...(25)> }

%{age: "30ish", gender: "Female", name: "Izzy"}

iex(26)> person.name

"Izzy"

iex(27)> person[:name]

"Izzy"

iex(28)> person["name"]

Nil

Functions

iex(29)> greeting = fn(place) -> "Hello, #{place}!" end

#Function<42.3316493/1 in :erl\_eval.expr/6>

iex(30)> greeting.("World")

"Hello, World!"

iex(31)> greeting.("Mars")

"Hello, Mars!"

iex(32)> place

"World"

iex(33)> c\_to\_f = fn(c)-> c\*1.8+23 end

#Function<42.3316493/1 in :erl\_eval.expr/6>

iex(34)> c\_to\_f.(20)

59.0

iex(35)> c\_to\_f = fn(c)-> c\*1.8+32 end

#Function<42.3316493/1 in :erl\_eval.expr/6>

iex(36)> c\_to\_f.(20)

68.0

iex(37)> c\_to\_f.(24)

75.2

iex(38)> c\_to\_f.(40)

104.0

Multiple Arguments Functions

iex(39)> greeting = fn(name, gender, age)-> "Hello, #{name}! I see you're #{gender} and you're #{age} years old." end

#Function<40.3316493/3 in :erl\_eval.expr/6>

iex(40)> greeting.("Izzy","Female",20)

"Hello, Izzy! I see you're Female and you're 20 years old."

iex(41)> greeting.("izzay")

\*\* (BadArityError) #Function<40.3316493/3 in :erl\_eval.expr/6> with arity 3 called with 1 argument ("izzay")

(stdlib 4.1.1) erl\_eval.erl:714: :erl\_eval.do\_apply/6

iex:41: (file)

Capture operator  
The only differences are that the capture operator syntax is shorter, and you refer to arguments by their position rather than a name

***Exercise:***

* Make a function which turns Fahrenheit temperatures into celsius.

iex(51)> f\_to\_c = fn(f)->(f - 32) \* 5/9 end

#Function<42.3316493/1 in :erl\_eval.expr/6>

iex(52)> f\_to\_c.(32)

0.0

iex(53)> f\_to\_c.(96)

35.55555555555556

iex(54)>

* Make a function which returns the number of seconds in the specified amount of days. For example, seconds.(2) should tell us how many seconds there are in 2 days.

iex(54)> second =&(&1\* (24\*60\*60))

#Function<42.3316493/1 in :erl\_eval.expr/6>

iex(55)> second.(1)

86400

iex(56)> second.(2)

172800

* Make a function which takes two maps with "age" keys in them and returns the average age.

iex(57)> person\_age = fn( %{"age" => age\_1},%{"age" => age\_2}) ->(age\_1+age\_2)/2 end

#Function<41.3316493/2 in :erl\_eval.expr/6>

iex(58)> person\_age.(%{"age" => 20},%{"age" => 20})

20.0

iex(59)> person\_age.(%{"age" => 20},%{"age" => 30})

25.0