

CodeMash 2015

Elixir For The Impatient

Abdul Habra
@ahabra

<https://github.com/ahabra/codemash15-elixir>

Agenda

- ❖ What to expect
- ❖ Language Description
- ❖ REPL: **Read Eval Print Loop**
- ❖ mix: build tool. Project structure and command line tools.
- ❖ Code:
 1. Hello World: modules, functions, anonymous functions, HereDocs, multiline strings, lists, maps, ExUnit.
 2. Sum: list processing, pattern matching, scope.
 3. FizzBuzz: if/else, cond, case, function pattern match, processes
 4. Data: struct, map, filter, reduce, pipes
- ❖ Q&A / Retro

Expectations

- ❖ You know programming
- ❖ Knowing some functional programming concepts is helpful but not required
- ❖ Will **not** debate why functional programming or Elixir are helpful
- ❖ You will learn the basics of the Elixir language
- ❖ Slides and code
- ❖ Questions and interruptions: We have a lot of material to cover, do you guys prefer waiting to the end or not?

What's Elixir

- ❖ Public in 2012
- ❖ Dynamic, functional, immutable data types
- ❖ Compiles and runs on Erlang VM: fault tolerance, lightweight processes
- ❖ Syntax inspired by Ruby
- ❖ Active community:
 - ❖ **Phoenix** web framework, by Chris McCord @ Dayton, OH

REPL

- ❖ Read Eval Print Loop
- ❖ at command line > **iex**
- ❖ `h()` or `h`, `h(ls)`, `h(List)`
- ❖ `1+1`, `:math.pow 10, 2`, `"a" <> "b"`
- ❖ `x=3` # no type specifier
- ❖ `is_number(x)`

Mix

- ❖ Command line build tool
- ❖ `mix new myproj` : look at generated directories and files
 - `.gitignore`, `README.md`, `lib /`, `test /`, ...
- ❖ `mix test`
- ❖ `mix clean`
- ❖ `mix help`

Code - Hello World

- ❖ Look at p1_hello.exs
- ❖ *.exs is a script, while *.ex gets compiled to .beam files
- ❖ Look at p2_hello.ex and p2_hello_test.exs
- ❖ Interesting things to note:
 - ❖ functions return last run statement
 - ❖ HereDocs @doc
 - ❖ Optional parenthesis for function calls
 - ❖ Unicode characters
 - ❖ Multiline strings
 - ❖ Anonymous functions and their invocation
 - ❖ Variables declaration and assignment
 - ❖ Lists and Maps

Code - Sum

- ❖ The *Enum* module
- ❖ Function capture
- ❖ Anonymous functions
- ❖ Pattern matching functions
- ❖ Using `_` to separate digits

Code - Fizz Buzz

- ❖ Kernel module: `rem`, `to_string`, `self`, `if`, `..`
- ❖ `if`, `else`, `case`
- ❖ Function pattern match with guard clause
- ❖ Process spawn and communication

Code - Data

- ❖ Struct: Like a map but with compile-time guarantees
- ❖ Map, filter, reduce
- ❖ Pipe |>

Q&A / Retro

- ❖ Questions
- ❖ How can I make this better?
- ❖ <https://github.com/ahabra/codemash15-elixir>
- ❖ @ahabra