Erlang Part I

Clara Benac Earle

Analysis of Concurrent and Distributed Systems

Universidad Politécnica de Madrid
Máster en Métodos Formales en Ingeniería Informática

April 5, 2022

Ericsson AXD 301

Erlang: a new language

POLITÉCNICA

Erlang: a new language

Borrows ideas from

- Prolog
- Functional programming languages

POLITÉCNICA

Erlang main characteristics

Erlang main characteristics

- Functional programming language (but not “pure” like Haskell)

POLITÉCNICA

Erlang main characteristics

- Functional programming language (but not “pure” like Haskell)
- Dynamic typing

POLITÉCNICA

Erlang main characteristics

- Functional programming language (but not “pure” like Haskell)
- Dynamic typing
- Highlights
  - Concurrency (actor model)
  - High reliability (Let it crash)

POLITÉCNICA

Erlang main characteristics

- Functional programming language (but not “pure” like Haskell)
- Dynamic typing
- Highlights
  - Concurrency (actor model)
  - High reliability (Let it crash)
- Code compiled to bytecode which runs on the Erlang Virtual Machine

POLITÉCNICA

Some companies using Erlang

POLITÉCNICA

POLITÉCNICA

Material

- www.erlang.org
- A recommended free book on Erlang `https://learnyousomeerlang.com/`
- Simon Thompson has put up a recommended set of videos on Erlang `https://www.youtube.com/watch?v=aEyQcZg-Njs` (Basic Erlang I by Simon Thompson | 1/13 of Erlang Express Course)

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats
- **Variables:** one can assign a value to a variable exactly once

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats
- **Variables:** one can assign a value to a variable exactly once
- **Atoms:** atoms are literals, constants with their own name for value

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats
- **Variables:** one can assign a value to a variable exactly once
- **Atoms:** atoms are literals, constants with their own name for value
- **Boolean Algebra & Comparison operators:** true and false are atoms.

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats
- **Variables:** one can assign a value to a variable exactly once
- **Atoms:** atoms are literals, constants with their own name for value
- **Boolean Algebra & Comparison operators:** true and false are atoms.
- **Tuples:** data structure with a fixed number of elements

POLITÉCNICA

Erlang Basics

- **Numbers:** integer and floats
- **Variables:** one can assign a value to a variable exactly once
- **Atoms:** atoms are literals, constants with their own name for value
- **Boolean Algebra & Comparison operators:** true and false are atoms.
- **Tuples:** data structure with a fixed number of elements
- **Lists:** they can contain anything!

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**
- **Guards:** additional clauses that can go in a function's head to make pattern matching more expressive

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**
- **Guards:** additional clauses that can go in a function's head to make pattern matching more expressive
- **If:** act like guards but outside of a function clause's head

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**
- **Guards:** additional clauses that can go in a function's head to make pattern matching more expressive
- **If:** act like guards but outside of a function clause's head
- **Case:** it is like the whole function head

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**
- **Guards:** additional clauses that can go in a function's head to make pattern matching more expressive
- **If:** act like guards but outside of a function clause's head
- **Case:** it is like the whole function head
- **Recursion:** a function that calls itself

POLITÉCNICA

Erlang Functions

- **Modules:** functions grouped in a file
- **Pattern matching**
- **Guards:** additional clauses that can go in a function's head to make pattern matching more expressive
- **If:** act like guards but outside of a function clause's head
- **Case:** it is like the whole function head
- **Recursion:** a function that calls itself
- **Higher Order Functions:** pass a function as a parameter to another function

POLITÉCNICA