Elixir

AUTOR: 8216

Índex

Introducció

- Programació funcional.
- Orientat a processos.
- Dissenyat per sistemes distribuïts.

- Aplicacions escalables i sostenibles.
- Tolerant a errors i de baixa latència.
- Concurrència i sintaxi expressiva.

La seva relació amb Erlang

- Sistemes distribuïts en temps real.
- Tolerant a errors i alta escalabilitat.

- Elixir és compatible i treballa juntament amb Erlang.
- Elixir s'executa en la VM d'Erlang (BEAM).
- Aprofita les llibreries d'Erlang i les seves eines.
- Elixir afegeix característiques addicionals per millorar la programació.

Història

- Creat l'any 2011 per José Valim.
- Nou llenguatge per BEAM.
- Va treballar amb Ruby i Erlang.

Relació amb LPs similars: Ruby vs Elixir

- Elixir influenciat amb Ruby i Erlang.
- Característiques de Ruby presents en Elixir:
  - Sistema de tipus dinàmic.
  - Paradigma de programació funcional.
  - Fort èmfasi en donar alta llegibilitat i expressivitat al codi.

```elixir
defmodule Math do
  def pow(_, 0), do: 1
  def pow(x, 1), do: x
  def pow(x, n) when rem(n, 2) == 0 do
    pow(x * x, div(n, 2))
  end
  def pow(x, n) do
    x * pow(x * x, div(n - 1, 2))
  end
end
```

Exponenciació ràpida en Elixir

```ruby
def pow(x, n)
  return 1 if n == 0
  return x if n == 1

  if n % 2 == 0
    pow(x * x, n / 2)
  else
    x * pow(x * x, (n - 1) / 2)
  end
end
```

Exponenciació ràpida en Ruby

Relació amb LPs similars: Erlang vs Elixir

- Elixir es va dissenyar per aprofitar els punts forts d'Erlang.
- Comparteixen vàries similituds:
  - Paradigma de programació funcional, concurrent i tolerant a errors.
  - Model de pas de missatges entre processos per sistemes distribuïts.
  - Pattern matching: fer matching condicions específiques.

```elixir
def pow(_, 0), do: 1
def pow(x, 1), do: x
def pow(x, n) when rem(n, 2) == 0 do
  pow(x * x, div(n, 2))
end
def pow(x, n) do
  x * pow(x * x, div(n - 1, 2))
end
```

Exemple de Pattern matching en Elixir

```erlang
-module(math_utils).
-export([fast_exp/2]).

fast_exp(Base, Exp) ->
    fast_exp(Base, Exp, 1).

fast_exp(_, 0, Result) ->
    Result;
fast_exp(Base, Exp, Result) when Exp rem 2 =:= 1 ->
    fast_exp(Base * Base, Exp div 2, Result * Base);
fast_exp(Base, Exp, Result) ->
    fast_exp(Base * Base, Exp div 2, Result).
```

Exemple de Pattern matching en Erlang

Sistema de tipus

- Sistema de tipus dinàmic.
  - Els tipus de les variables es comproven en temps d'execució.
- Sistema de tipat fort.
  - Restriccions sobre les operacions de tipus diferents.

```
# Declaracions de variables
num = 42           # variable de tipus enter
str = "Hola"       # variable de tipus string
lst = [1, 2, 3]    # variable de tipus llista

# Operacions
res1 = num + 10        # suma d'un enter amb un altre enter
res2 = str <> " món"   # concatenació d'strings
res3 = lst + 5         # error de tipus
```

Paradigmes de programació

Elixir és un llenguatge multiparadigma:

- Programació funcional.
- Programació orientada a processos.
- Metaprogramació.

```
defmodule Factorial do
  def calculate(0), do: 1
  def calculate(n), do: n * calculate(n - 1)
end
```

Càlcul del factorial en Elixir

Sistema d'execució

- Elixir s'executa en la màquina virtual d'Erlang (BEAM).
  1. Codi Elixir (.ex) → BEAM bytecode (.beam) [compilador d'Elixir]
  2. BEAM bytecode (.beam) → Fitxer binari (.exe) [BEAM]

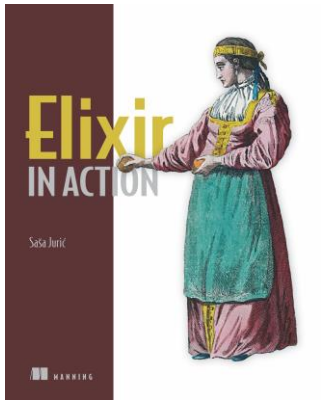
- IEx - Shell interactiva d'Elixir:
  - Executa codi interactivament.
  - Depurar i recarregar codi.

Visió personal i crítica

- Llenguatge adequat per crear pàgines web amb molts usuaris.
- Gran expressivitat i sintaxi senzilla.
  - Simplifiquen la tasca de generar codi als programadors.
- Mix: eina per crear, compilar i provar aplicacions amb Elixir.

Avaluació de les fonts

- Preferentment pàgines web oficials d'Elixir i Erlang.
- També un llibre de programació sobre Elixir i un pòdcast.
- Per dubtes puntuals, pàgines de tercers verificades.



Bibliografia i Webgrafia

- Sasa Juric. *Elixir in Action*. Manning, 2015.
- elixir-lang.github.com — elixir-lang.org. https://elixir-lang.org/.
- Index - Erlang/OTP — erlang.org. https://www.erlang.org/.
- Remote Ruby — José Valim, creator of Elixir and form Rails core contributor — remoteruby.com. https://remoteruby.com/178.
- A beginner's guide to the Elixir programming language — educative.io. https://www.educative.io/blog/elixir-functional-programming.
- Elixir (programming language) - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Elixir(programminglanguage).

Moltes gràcies per la vostra atenció!