

Welcome to Elixir

“This is good shit”

— Joe Armstrong

June 19, 2016

Table of Contents

Overview

Power of Erland

Syntax

Macros

Tooling and Abstractions

Motivation

- ▶ Power of Erland
- ▶ Syntax
- ▶ Macros
- ▶ Tooling and Abstractions

Table of Contents

Overview

Power of Erland

Syntax

Macros

Tooling and Abstractions

Erlang

- ▶ created in mid-1980s
- ▶ designed for telecom
- ▶ connect multiple systems
- ▶ minimal impact of errors
- ▶ entire system should never go down

High availability

- ▶ fault tolerance
- ▶ scalability
- ▶ distribution
- ▶ responsiveness
- ▶ live update

How do they do it?

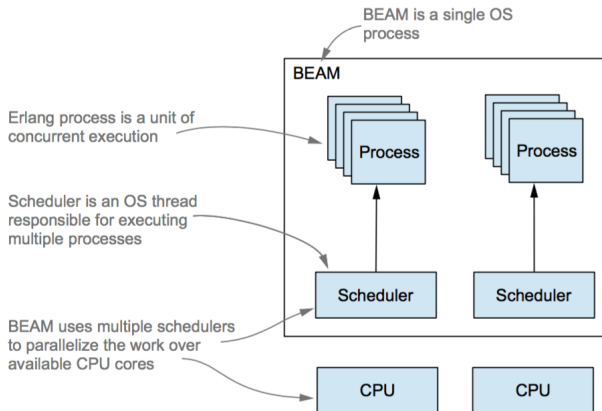


Table of Contents

Overview

Power of Erland

Syntax

Macros

Tooling and Abstractions

Syntax

“Elixir syntax is like a marriage of DSL friendly Ruby and the powerful hygienic macros of Clojure.”

– Devin Torres

The basics you know

```
1 + 1          # => 2
2 * (3 + 1) / 4 # => 2.0
1 + 2; 1 + 3    # => 4
```

```
greeting = "Hello _World!"
IO.puts(greeting)
# => Hello World!
# => :ok
```

Modules and function, oh my!

```
defmodule Geometry do  
  def rectangle_area(a, b) do  
    a * b  
  end  
end
```

Composing functions

```
def process_xml(model, xml) do  
  model  
  |> update(xml)  
  |> process_changes  
  |> persist  
end
```

Function arity

```
defmodule Rectangle do
  def area(a), do: area(a, a)
  def area(a, b), do: a * b
end
```

Table of Contents

Overview

Power of Erland

Syntax

Macros

Tooling and Abstractions

Table of Contents

Overview

Power of Erland

Syntax

Macros

Tooling and Abstractions