Metaprogramming

Section 5

Packt

In this Section, we are going to take a look at...

- Elixir AST
- Quote and unquote
- Defining macros

Packt>

Quote and Unquote

Packt

In this Video, we are going to take a look at…

- Elixir abstract syntax tree
- Quoting expressions
- Unquote

Packt>

Why Metaprogramming?

Packt

The Abstract Syntax Tree

```
say_to(:David, “Hello!”)
```

Packt>

The Abstract Syntax Tree

```
say_to(:David, “Hello!”)
```

```
{:say_to, [], [:David, “Hello”]}
```

Packt>

The Abstract Syntax Tree

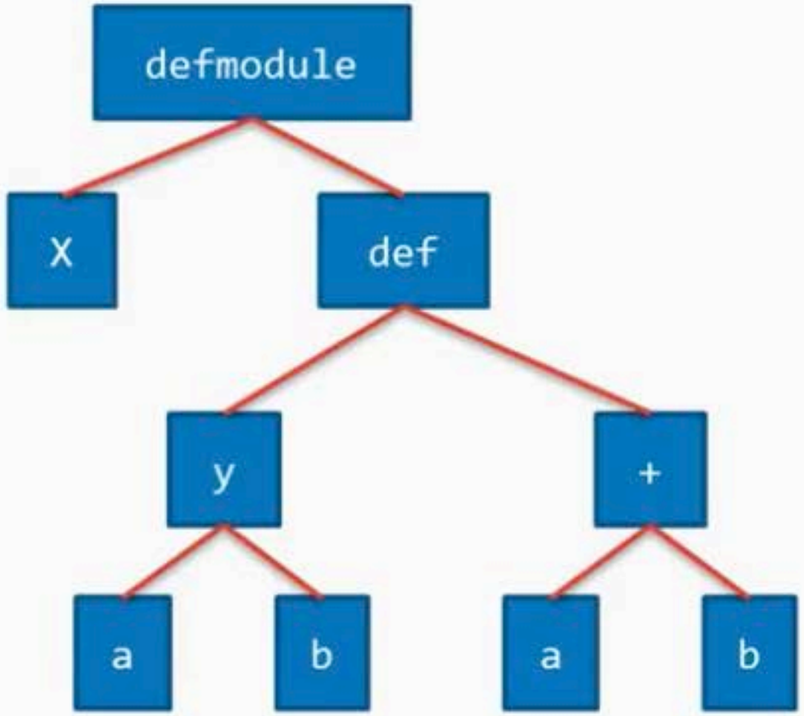
```
say_to(:David, “Hello!”)
```

```
{:say_to, [], [:David, “Hello”]}
```

Packt>

The Abstract Syntax Tree

```
defmodule X do
  def y(a, b) do
    a + b
  end
end
```



Packt>

quote

Packt

```
➜ Projects iex
Erlang/OTP 20 [erts-9.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.5.1) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> quote do: 1
1
iex(2)> quote do: "hello"
"hello"
iex(3)> quote do: [1,2,3,4]
[1, 2, 3, 4]
iex(4)> quote do: {:a, 2}
{:a, 2}
iex(5)> quote do: %{a: 2}
{:%{}, [], [a: 2]}
iex(6)> quote do: [a: 3, b: "hello"]
[a: 3, b: "hello"]
iex(7)> quote do
...(7)> hello(1, name)
...(7)> end
{:hello, [], [1, {:name, [], Elixir}]}
iex(8)> name = "Joao"
"Joao"
iex(9)> quote do
...(9)> hello(1, name)
...(9)> end
{:hello, [], [1, {:name, [], Elixir}]}
iex(10)>
```

Packt

Unquote

Evaluate expressions inside quoted expressions.

```
quote do: 1 + 2
```

```
{:+, [...], [1, 2]}
```

Packt>

Unquote

Evaluate expressions inside quoted expressions.

```
quote do: unquote(1 + 2)
```

```
3
```

Packt>

```
➜ Projects iex
Erlang/OTP 20 [erts-9.0.4] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:10] [hipe] [kernel-poll:false] [dtrace]

Interactive Elixir (1.5.1) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> name = "David"
"David"
iex(2)> quote do
...(2)> name
...(2)> end
{:name, [], Elixir}
iex(3)> quote do
...(3)> unquote(name)
...(3)> end
"David"
iex(4)> quote do
...(4)> hello(unquote(name), 20)
...(4)> end
{:hello, [], ["David", 20]}
iex(5)>
```

Packt