

MAKING SENSE OF ELIXIR DIRECTIVES

WHAT ARE DIRECTIVES?

- > WAY TO DECLARE, IMPORT AND REFERENCE OTHER MODULES
 - > ELIXIR HAS **FOUR** OF THEM
 - > BUT, BUT, WHY?

ELIXIR'S DIRECTIVES

- > ALIAS
- > IMPORT
- > REQUIRE
- > USE

ALIAS

USED TO REFERENCE A MODULE IN SHORT-HAND

```
alias Foo.Bar, as: Bar
```

```
# Can now do  
Bar.function
```

IMPORT

USED TO EASILY ACCESS FUNCTIONS OR MACROS FROM OTHER
MODULES

```
import List, only: [duplicate: 2]
```

```
# Can now do
```

```
duplicate("hello", 3)
```

REQUIRE

USED TO MAKE SURE MODULE IS COMPILED AND AVAILABLE

```
require Foo
```

```
Foo.a_macro
```

USE

CALLBACK THAT INJECTS CODE INTO CURRENT CONTEXT

```
use ExUnit.Case, async: true
```

```
test "now have testing macros" do  
  assert true
```

```
end
```

SUMMARY

- > `alias` TO MORE CONVENIENTLY REFERENCE MODULES
- > `import` TO MORE CONVENIENTLY REFERENCE FUNCTIONS
- > `require` TO ENSURE MACROS ARE AVAILABLE AND COMPILED
- > `use` TO GIVE MODULE OPPORTUNITY TO 'HOOK' INTO YOUR CODE

BACKGROUND

- > CURRENT ELIXIR DIRECTIVE DOCS
- > IRC DESIGN DISCUSSION BETWEEN ME, JOSE VALIM, AND CHRIS MCCORD ABOUT THIS VERY TOPIC
- > MAILING LIST DISCUSSION ABOUT DIRECTIVES FROM LONG AGO
 - > MY PROPOSED CHANGES