



# **Programação Funcional com Elixir**



# **Criando Módulos e Funções**

# Criando Módulos e Funções

- Como já vimos anteriormente, o Elixir trabalha separando funções em módulos, inclusive já criamos nosso primeiro módulo e função aulas atrás.

- `defmodule Say do`
- `def hello do`
- `"Olá Mundo!!!"`
- `end`
- `end`

# Criando Módulos e Funções

- Conforme podemos observar, para criar um módulo usamos a palavra chave `defmodule` e para criar funções usamos a palavra chave `def`
  - `defmodule` Say do
  - `def` hello do
  - `"Olá Mundo!!!"`
  - `end`
  - `end`

# Criando Módulos e Funções

- Por convenção, os nomes dos módulos devem começar com letra maiúscula, seguindo o padrão **CamelCase** para nomes compostos.

- `defmodule SaySomething do`
- `def hello do`
- `"Olá Mundo!!!"`
- `end`
- `end`
-

# Criando Módulos e Funções

- Já as funções devem seguir a convenção **snake\_case**, ou seja, devem ser sempre escritas em minúsculo e para nomes compostos separá-los com um underscore.

```
○ defmodule SaySomething do
○   def hello_world do
○     "Olá Mundo!!!"
○   end
○ end
○
```

# Criando Módulos e Funções

- Podemos usar namespaces para facilitar e evitar confusões em nossas aplicações. Para isso, podemos usar um ponto (.) para separar os namespaces em nossos módulos, veja:

- `defmodule MyModule.SaySomething do`
- `def hello_world do`
- `"Olá Mundo!!!"`
- `end`
- `end`