



Programação Funcional com Elixir



Conhecendo Structs

Conhecendo Structs

- Anteriormente conhecemos os maps (mapas)...

```
iex> map = %{a: 1, b: 2}
```

```
%{a: 1, b: 2}
```

```
iex> map[:a]
```

```
1
```

```
iex> %{map | a: 3} # atualizando um map
```

```
%{a: 3, b: 2}
```

Conhecendo Structs

- Structs são mapas especiais com um conjunto definido de chaves e valores padrões. Ele deve ser definido dentro de um módulo, no qual leva o nome dele.
- É comum para um struct ser a única coisa definido dentro de um módulo.

`https://hexdocs.pm/elixir/Kernel.SpecialForms.html#%25/2`

Conhecendo Structs

```
defmodule User do
  defstruct name: "John", age: 28
end
```

- A partir desse momento, podemos usar a struct **%User{}**

```
jack = %User{name: "Jackson", age: 39}
jack.name
jack.age
```

Conhecendo Structs

- Para atualizar temos a mesma situação do mapa...

```
jack = %{jack | name: "Jackson Pires"}
```

```
jack.name
```

```
jack.age
```

Conhecendo Structs

- Se quiser criar structs sem um valor padrão use **nil**.

```
defmodule User do
  defstruct name: nil, age: nil
end
```

Conhecendo Structs

- Todas as chaves de uma struct são opcionais, então se quiser tornar alguma chave obrigatória basta usar o **@enforce_keys**

```
defmodule User do
  @enforce_keys [:name]
  defstruct name: nil, age: nil
end
```