

```
> str = "Javascript"
```

```
> str.length
```

```
=> 10
```



```
> str = "Javascript"
```

```
> str.length
```

```
=> 10
```

```
> str = "Elixir"
```

```
> str.length
```

```
** Error!!
```



```
> str = "Javascript"
```

```
> str.length
```

```
=> 10
```

```
> str = "Elixir"
```

```
> length(str)
```

```
=> 6
```





Do objeto ao funcional

Marcelo Li Koga

TDC Porto Alegre 2018

| Por que funcional?



elixir



Aviso!



Os slides a seguir contém código com sintaxe *inspirada* em ruby, javascript e elixir, mas os conceitos podem ser aplicados na maioria das linguagens.



Dados imutáveis



```
class Checkout
```

```
  @cart = []
```

```
  def addToCart(item)
```

```
    @cart.push(item)
```

```
  end
```

```
end
```

```
defmodule Checkout
```

```
  def addToCart(cart, item)
```

```
    return cart ++ [item]
```

```
  end
```

```
end
```



| Nada de funções *void*



```
class Checkout
```

```
  @cart = []
```

```
  def addToCart(item)
```

```
    @cart.push(item)
```

```
  end
```

```
end
```

```
defmodule Checkout
```

```
  def addToCart(cart, item)
```

```
    return cart ++ [item]
```

```
  end
```

```
end
```



Tudo está nos argumentos



```
class Checkout
```

```
  def getFirstItem()  
    return @cart.first  
  end  
end
```

```
end
```

```
defmodule Checkout
```

```
  def getFirstItem(cart)  
    return List.first(cart)  
  end  
end
```

```
end
```



Fluxo de transformação



```
def capitalize(title)           # "the office"
  words = String.split(title)   # ["the", "office"]
  capWords = capitalize_all(words) # ["The", "Office"]
  return join(capWords, " ")    # "The Office"
end
```



```
def capitalize(title)
  return join(capitalize_all(String.split(title)), " ")
end
```



```
def capitalize(title)
  title           # "the office"
  |> String.split() # ["the", "office"]
  |> capitalize_all() # ["The", "Office"]
  |> join(" ")      # "The Office"
end
```



| **for => map/reduce**



```
# names = ["Jim", "Pam"]
def shout(names)
  up = []
  for(name in names)
    up.push(name.upcase())
  end
  return up # ["JIM", "PAM"]
end
```

```
def shout(names)
  return
  map(names, &upcase)
end
# ["JIM", "PAM"]
```



```
def totalFruits(items)
  sum = 0
  for(item in items)
    if(item.cat == "Fruit")
      sum += item.price
    end
  end
  return sum
end
```

```
def isFruit(item)
  item.cat == "Fruit"
end
```

```
def totalFruits(items)
  items
  |> filter(&isFruit)
  |> map(fn item -> item.price)
  |> sum()
end
```



```
def sumDoubleOfOdds(numbers)
  sum = 0
  for(n in numbers)
    if isOdd(n)
      sum += n*2
    end
  end
  return sum
end
```

```
def sumDoubleOfOdds(numbers)
  numbers           # [1,2,3]
  |> filter(&isOdd) # [1,3]
  |> map(fn n -> n*2) # [2,6]
  |> sum()           # 8
end
```



Não controle, só declare



```
def sumDoubleOfOdds(numbers)
  sum = 0
  for(i=0; i < numbers.length;
i++)
    if isOdd(numbers[i])
      sum += numbers[i]*2
    end
  end
  return sum
end
```

```
def sumDoubleOfOdds(numbers)
  numbers           # [1,2,3]
  |> filter(&isOdd)  # [1,3]
  |> map(fn n -> n*2) # [2,6]
  |> sum()           # 8
end
```



| **if => pattern matching**



```
def isPaid(p)
  return p.status == 'paid'
end
```




```
def isPaid(p)
  if(p == nil)
    return false

    return p.status == 'paid'
  end
```

```
def isPaid(nil)
  return false
end
```

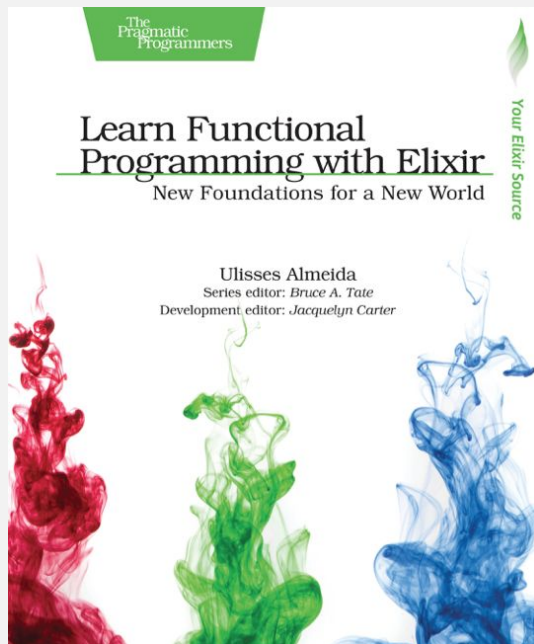
```
def isPaid(p)
  return p.status == 'paid'
end
```



```
def isOrderPaid(order)
  order
  |> getPayment()
  |> isPaid()
end
```



Referências



Elixir-koans

Elixir learning exercises

elixirkoans.io





**QUERO
BOLSA**

Assim é **fácil** entrar na faculdade.



QUERO
EDUCAÇÃO

- **querobolsa.com.br**
- **+300 mil** vagas no site
- Desde 2012, **+420 mil** alunos matriculados



Obrigado!



@mlikoga

que.bo/vagas-dev



Assim é **fácil** entrar na faculdade.



Um resuminho de 2018



Chat

87mi

Chats atendidos

800

A média contatos efetivos por Guia semanalmente

Facebook

1.5mi

Interações no Facebook por semana, por Guia

WhatsApp

66mi

Conversas de Whatsapp atendidas

2mil

Contatos efetivos por semana, por Guia

30%

Em média maior que a conversão pelos canais de Ligações e Chat





QueroWhats

PAINEL DO OPERADOR

Ativas

12 +

Marcelo Li Koga 20:25
Vamos fechar então? 🙌

Maria de Fatima Granja... 22/11/18
QueroBolsista, já conferiu a...

Plínio Henrique 21/11/18
Veja o que acha dela

José Augusto Neto 08/10/18
Oi José, tudo bem? Sou Gul...

Márcio André Ferreira ... 27/09/18
QueroBolsista, você quer co...

50 mil mensagens por dia

Marcelo Li Koga 5511982250439

Oi, gostaria de estudar Design em São José dos Campos
29/11/18 - 20:23

Vocês têm bolsa?
29/11/18 - 20:23

MARCELO KOGA
oi Marcelo! Vou procurar aqui pra vc
29/11/18 - 20:24

MARCELO KOGA
Achei essa bolsa incrível de 67%
29/11/18 - 20:24

MARCELO KOGA
UNIP - Design
São José Dos Campos

Início 1º semestre de 2019
Noite
6 semestres
Bacharelado (graduação)
Presencial
Jardim das Indústrias , Rod. Presidente Dutra, km 157,5, São José dos Campos - SP

Mensalidade sem o desconto: R\$ 1.285,72
Bolsa: 67.24%
Mensalidade *com o desconto*: R\$ 421,20
Valor da pré-matrícula: R\$ 842,40

<https://querobolsa.com.br/o/d5279e84-7d22-4e59-8425-f93f2069ca95>
29/11/18 - 20:25

Opa, q beleza!! 😄😄
29/11/18 - 20:25

Nova mensagem

Matrículas

Design

Matrícula
ID 5697779

Situação: **INICIADO**

29/11/2018 - 20:24

UNIP
São José Dos Campos

- 🕒 2019.1
- ★ Bacharelado (graduação)
- 📖 Presencial
- 🕒 Noite
- 📈 6 semestres
- 📍 Jardim das Indústrias , Rod. Presidente Dutra, km 157,5, São José dos Campos - SP
- 📞 0800 010 9000

5 BOLSAS RESTANTES

Valor da Pré-matrícula R\$ 842,40

DESCONTO COMERCIAL

Mensalidade sem desconto R\$ 1.285,72

Desconto Quero Bolsa 67.24%



Marcelo Li Koga

- Eng. de Computação USP
- Java: 4 anos
- C++: 4 anos
- Elixir: 6 meses



Anotações

<https://dev.to/allanmacgregor/you-should-learn-functional-programming-in-2018-4nff>

