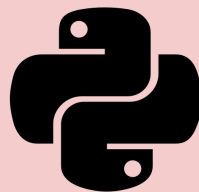


# PROGRAMAÇÃO FUNCIONAL EM PYTHON

Sobre **o que** é essa palestra:



- **Definição** de programação funcional
- **Características** e **vantagens**
- **Exemplos** de aplicação em Python





Oi :)  
Me chamo Elma

Desenvolvedora de Software na **Codeminer42**  
Membro do **Pyladies** e **WTM Natal**



@elmasantos



@elmasnts

# Paradigma de programação

Classifica linguagens

Funcionalidades

Estilo de código

# Paradigma de programação

## Programação funcional

“trata a computação como uma avaliação de funções matemáticas e que evita estados ou dados mutáveis”

# Conceitos importantes

- Funções de primeira classe (first-class functions)
- Dados imutáveis
- Independentes de estado (stateless)

# Funções puras

Não produzem efeitos colaterais

# Funções de alta ordem (high order functions)

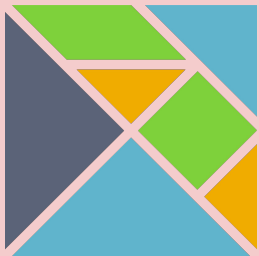
Podem aceitar outras funções como parâmetro e retornar funções como saída



# Por que funcional?

- Menos efeitos colaterais
- Melhor manutenibilidade
- Melhor testabilidade
- Facilita reuso de código

# Algumas linguagens



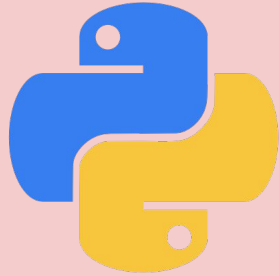
**Elm**



**Haskell**



**Clojure**



**Python - Multiparadigma**

# Dados mutáveis

```
>>> elenco_bacurau = [  
    {'nome': 'Sonia Braga', 'personagem': 'Domingas'},  
    {'nome': 'Barbara Colen', 'personagem': 'Teresa'},  
    {'nome': 'Silvero Pereira', 'personagem': 'Lunga'}  
]
```

```
>>> elenco_bacurau[0]['nome'] = 'Kleber'  
>>> elenco_bacurau[0]  
{'nome': 'Kleber', 'personagem': 'Domingas'}
```

```
>>> del elenco_bacurau[0]  
>>> elenco_bacurau  
[{'nome': 'Barbara Colen', 'personagem': 'Teresa'}, {'nome':  
'Silvero Pereira', 'personagem': 'Lunga'}]
```

# Dados imutáveis

```
import collections
Pessoa = collections.namedtuple('Pessoa', ['nome',
'personagem'])
```

```
elenco_bacurau = (
    Pessoa(nome='Sonia Braga', personagem='Domingas'),
    Pessoa(nome='Barbara Colen', personagem='Teresa'),
    Pessoa(nome='Silvero Pereira', personagem='Lunga')
)
```

```
>>> elenco_bacurau[0].nome = 'Kleber'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
```

# Dados imutáveis

```
>>> del elenco_bacurau[0]  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: 'tuple' object doesn't support item deletion
```

# Funções não puras

```
def remove_last_item(mylist):  
    mylist.pop(-1) # modifica a lista original
```

# Funções Puras

```
def butlast(mylist):  
    return mylist[:-1] # retorna uma cópia da lista
```



# Funções de alta ordem

```
def write_message(message, action):  
    action(message)
```

```
>>> write_message('Oi', print)  
Oi
```

# Funções embutidas (built-in functions) de alta ordem

Função **map**: `map(function, iterable)`

Função **filter**: `filter(function, iterable)`

# map

```
>>> integers = [1, 2, 3, 4, 5]
>>> list(map(lambda x: x * x, integers))
[1, 4, 9, 16, 25]
>>> integers
[1, 2, 3, 4, 5]
```

# filter

```
>>> integers = [1, 2, 3, 4, 5]
>>> list(filter(lambda x: x % 2 == 0, integers))
[2, 4]
>>> integers
[1, 2, 3, 4, 5]
```

# Compreensão de lista (list comprehension)

```
>>> integers = [1, 2, 3, 4, 5]
```

```
>>> [x * x for x in integers]  
[1, 4, 9, 16, 25]
```

```
>>> [x for x in integers if x % 2 == 0]  
[2, 4]
```

```
>>> integers  
[1, 2, 3, 4, 5]
```

**Concluindo...**

# Referências



- <https://docs.python.org/3/howto/functional.html>
- <https://julien.danjou.info/python-and-functional-programming/>
- <https://stackabuse.com/functional-programming-in-python/>
- <https://towardsdatascience.com/elements-of-functional-programming-in-python-1b295ea5bbe0>

**Obrigada! :)**