

---

# Python

## Programando de forma idiomática

Eliakin Costa

Instituto Federal da Bahia  
campus Salvador  
eliakincosta@ifba.edu.br

26 de abril de 2017



# Agenda

---

- 1 Introdução
- 2 Tipagem Dinâmica, mas nem tanto
- 3 Tipos, Instruções e Estilo
- 4 Funções e Classes
- 5 Ainda Tem muito mais



# Quem sou eu?

---

- Meu nome é Eliakin Costa,
- Sou aluno de ADS (IFBA),
- Apaixonado pelo software livre \o/
- Membro e Colaborador do OPAI e KDE
- **Contatos:**
  - **E-mail:** [eliakim170 at gmail.com](mailto:eliakim170@gmail.com)
  - **GitHub:** [github.com/eliakincosta](https://github.com/eliakincosta)



# Introdução



# O começo de tudo

---

- CWI - Centro Tecnológico
- Guido Van Rossum
- Linguagem ABC
- Guido precisava resolver um problema relacionado com o Amoeba xD



# Python!? Que nome é esse?

---



# Python!? Que nome é esse?

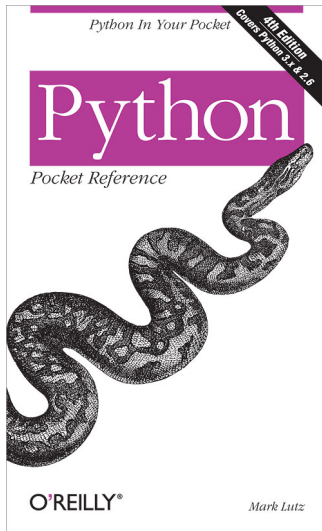
---

- Python Flying in Circus
- Episódio do spam xD
- Guido não gostava da relação do python com uma cobra



# Python!? Que nome é esse?

---





# Guido Van Rossum - O ditador benevolente

---

- Está vivo!!!!
- Google de 2005-2013
- Dropbox de 2013-hoje



# Por que eu devo aprender Python?

---

- Linguagem de Propósito geral
- Simples e intuitivo
- Roda em qualquer sistema
- Você consegue fazer quase tudo apenas com os recursos da própria linguagem
- Multiparadigma



# Quem está usando python?

---



# Quem está usando python?

---



## Tipagem Dinâmica, mas nem tanto



# Características da Tipagem Dinâmica

---

- Verificações de tipo feitas em runtime
- Verificações feita sobre o valor(ex: `class int`)
- Bob Harper, um dos criadores da linguagem Standard ML, "uma linguagem dinamicamente tipada é uma linguagem estaticamente tipada com apenas um tipo estático".



# Como é a tipagem estática?

---

## JAVA

---

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        int a = 2;
        String b = "teste";
        a = b; //ERRADO, FALHA NA COMPILACAO
    }
}
```

---



# Como é a tipagem dinâmica?

---

## Python

---

```
a = 2
```

```
a = "texto"
```

```
#Variaveis funcionam como referencias polimorficas.
```

```
#Verificacao de tipo em runtime. Feita sobre o dado.
```

---





# Tipagem dinâmica, mas forte

---

- Observe o seguinte código em php:

```
echo 1 + '13 bananas';
```

- Vamos testar o mesmo código em python.



- Código mais robusto
- Tipos inválidos ocasionarão exceções
- Explícito é melhor do que implícito



# Tipos, Instruções e Estilo



## ■ Números:

- inteiro (int)
- ponto flutuante (float)
- booleano (bool) 1 ou 0
- complexo (complex)

## ■ Obs:

- Suportam as operações básicas, mod, etc (+, -, \*, /, %, \*\*, +=, -=, \*=, /=, %=, \*\*=).
- Operações entre diferentes tipo irão resultar no tipo mais complexo
- Cuidado, números inteiros tem precisão infinita
- Controle de fluxo e bool



# Tipos Básicos do python

---

## ■ Iteráveis:

Tabela: Iteráveis

	Ordenada	Modificável	Unicidade
strings	X		
listas	X	X	
tuplas	X		
sets		X	X

## ■ Vamos tentar um pouco de código

**Obs: Iremos focar apenas em listas**



## Cópia Rasa de listas

---

```
a = [1,2,3]
>>> b = a
>>> b is a
True
>>> a.append(4)
>>> b
[1, 2, 3, 4]
```

---

**Como resolver isso de uma maneira mais fácil?**



- Podemos usar fatiamento de iteráveis (slicing)
- O fatiamento faz uma cópia de valor do iterável
- Essa cópia é feita no primeiro nível
- **Python aceita níveis arbitrários em estrutura modificáveis**
- Vamos entender isso!!!



- O slicing muda a forma como percorremos uma lista
- Mas e se precisarmos fazer verificações para filtrar a lista?





# Hackeando Listas com Python

---

Obtendo Números pares de uma lista

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]

nova_lista = []

for elemento in lista:
    if elemento%2==0:
        nova_lista.append(elemento)

print(nova_lista)
```



# Hackeando Listas com Python

---

Fazendo o mesmo de uma forma mais pythonica

---

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]
```

```
print([elemento for elemento in lista if elemento%2==0])
```

---



# Funções e Classes



**Ainda Tem muito mais**



# Finalizando...

---



# Referências

---

