

PYTHON BASICS



ADRIANO LEITE DA SILVA

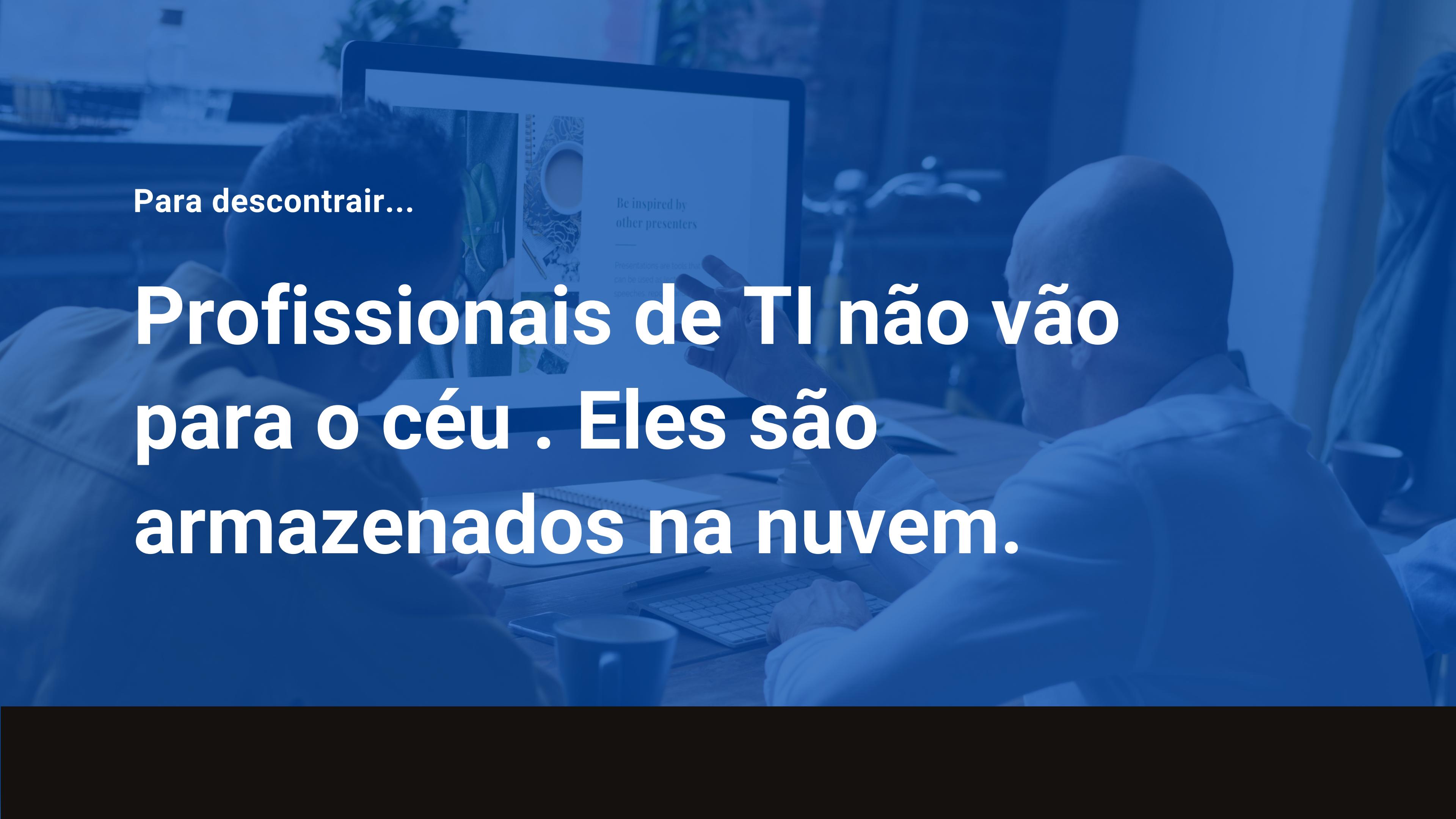
TECH LEAD MOBILE | PROFESSOR | ESCRITOR | MENTOR

32 anos, roqueiro, nerd, tatuado, barbudo, apaixonado por histórias medievais e pós apocalípticas. Fã de Senhor dos Anéis e StarWars.

Estou na área de tecnologia há 15 anos. Professor há 14 anos. Passando por grandes instituições.

Um completo apaixonado pelos poderes que a Tecnologia e a Educação tem para mudar a vida das pessoas. Da mesma forma que mudou a minha.



A person is sitting at a desk, looking at a laptop screen. The screen displays a presentation slide with the text "Be inspired by other presenters" and "Presentations are tools that can be used as lectures, speeches, reports, etc.". The person is wearing a blue shirt and has their hands resting on the keyboard.

Para descontrair...

Profissionais de TI não vão
para o céu . Eles são
armazenados na nuvem.

O que é Python?

Python é uma linguagem de programação de **alto nível** – ou **High Level Language**. Dinâmica, interpretada, modular, multiplataforma e orientada a objetos.



A origem da Linguagem Python?

Idealizada e desenvolvida por Guido Van Rossum, matemático holandês, no início dos anos 90, o Python foi criado com o objetivo de otimizar a leitura de códigos e estimular a produtividade de quem os cria, seja este um programador ou qualquer outro profissional.

A ideia surgiu, como quase todas as boas ideias, de uma necessidade: a de economizar tempo no desenvolvimento e melhorar a eficiência em um projeto desenvolvido no instituto onde Guido era pesquisador.

Para que esta melhoria pudesse ser feita de forma mais rápida e eficaz, Guido desenvolveu uma linguagem muito descomplicada e flexível: o Python.

A origem da Linguagem Python?

Uma vez que esta linguagem passou a possibilitar a criação desde scripts muito simples até sistemas extremamente poderosos, profissionais de várias áreas começaram a progressivamente utilizá-la cada vez mais.

Hoje, além dos desenvolvedores de software, temos biólogos, contadores, físicos e outros profissionais potencializando suas habilidades através dela.

Estes são alguns motivos que tem feito o uso do Python crescer consideravelmente nos últimos anos em detrimento de outras linguagens.

Utilização do Python hoje em dia.

Scripting e Automação

Automatizar tarefas é um dos maiores objetivos de um profissional de tecnologia.

E desenvolver scripts de automação com Python é totalmente possível e rápido graças às suas várias bibliotecas nativas, disponíveis junto com sua instalação.

Desenvolvimento Web

Uma das aplicações mais comuns do Python é o desenvolvimento de aplicações para web. Desde sites simples, como hotpages para ações de marketing, quanto ERPs capazes de administrar empresas e realizar vendas de produtos e serviços.

Tudo isso com a utilização de frameworks como Django, Flask e outros.

Enquadramento de Testes

Então, utilizar o Python é poder ter a certeza que sua aplicação poderá ser testada com muita eficiência.

Devs Python adoram utilizar TDD (Test Driven Development), trazendo mais segurança e confiança aos projetos,

Utilização do Python hoje em dia.

Big Data

A análise e interpretação de grandes volumes de dados, área de conhecimento que chamados de Big Data, é um dos mais populares campos onde Python tem se tornado a linguagem favorita.

Por ser facilmente compreendida por profissionais de diversas especialidades, o Python é, antes de tudo, um facilitador.

Ciência de Dados

Acompanhando o próprio avanço da indústria tecnológica, o crescente número de bibliotecas disponíveis em Python voltadas à análise de dados oferece funções e métodos de otimização para praticamente quaisquer objetivos.

Inteligência Artificial

Quando você faz uma busca no Google e encontra exatamente a resposta que esperava, o Python desempenhou exatamente o papel que se esperava de sua atuação.

Isso porque os algoritmos e modelos de inteligência artificial presentes no buscador, elaborados para predizer a sua intenção de busca, são em sua maioria desenvolvidos nesta linguagem.

Dev Python

Pesquisa feita em
17/06/21

Para região de SP

 CI&T Desenvolvedor Python: mensal 4.6 ★ 2 salários Ver 2 salários de todas as localizações	Cerca de R\$ 6 mil -R\$ 7 mil 
 Storm Security Desenvolvedor Python: mensal 3.3 ★ 2 salários Ver 2 salários de todas as localizações	Cerca de R\$ 5 mil -R\$ 6 mil 
 Plugar Desenvolvedor Python: mensal 3.5 ★ 2 salários Ver 2 salários de todas as localizações	Cerca de R\$ 3 mil -R\$ 3 mil 
 Nexus Edge Desenvolvedor Python: prestador de serviços mensal 1 ★ 2 salários Ver 2 salários de todas as localizações	Cerca de R\$ 3 mil -R\$ 3 mil 
 Accenture Desenvolvedor Python: mensal 4 ★ 1 salários Ver 1 salários de todas as localizações	Cerca de R\$ 4 mil -R\$ 4 mil 
 Stefanini Desenvolvedor Python: mensal 3.7 ★ 1 salários Ver 1 salários de todas as localizações	Cerca de R\$ 8 mil -R\$ 9 mil 
 Stone Desenvolvedor Python: mensal 4 ★ 1 salários Ver 1 salários de todas as localizações	Cerca de R\$ 3 mil -R\$ 3 mil 



Fundamentos de Python



```
class Main {  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 10;  
  
        System.out.println(a);  
        System.out.println(b);  
    }  
}
```

```
a = 7  
b = 10  
  
print(a,b)
```

Conceitos Básicos

Variáveis

Como o próprio nome diz, são informações que variam.

São espaços reservados na memória para armazenar informações que serão utilizadas na execução do programa.

```
a = 32  
b = 2021  
c = 10.99
```

```
caractere = 'C'  
textoA = 'Olá Mundo!'
```

```
vaiChover = 'True'
```

Conceitos Básicos

Tipos de Dados

Quando falamos de dados, podemos trabalhar com os tipos:

String = Texto

Int = Inteiros

Float = Números fracionais

Bool ou **Boolean** = Lógico

```
a = 32  
b = 2021  
c = 10.99
```

```
caractere = 'C'  
textoA = 'Olá Mundo !'  
vaiChover = 'True'
```

Conceitos Básicos

Concatenando Informações

Quando falamos de informações em uma linguagem de programação, as vezes, vamos nos deparar com um problema, apresentar informações pré-determinadas e as variáveis. Para isso chamamos a concatenação para apresentar da forma que desejamos.

```
nome = 'Adriano'  
sobrenome = 'Leite da Silva'  
  
print(nome)  
print(sobrenome)  
print(nome, sobrenome)  
print(nome + sobrenome)  
print(nome + ' ' + sobrenome)
```

Neste exemplo, estamos trabalhando com informações que foram previamente armazenadas em variáveis e concatenadas com textos pré-definidos de acordo com a necessidade do que desejamos apresentar na tela.

```
nome = 'Adriano'  
sobrenome = 'Leite da Silva'  
idade = 32  
email = 'adrianoleitedasilva@outlook.com'  
  
print('Olá, meu nome é ' + nome + ', tenho 32 anos de idade! E meu e-mail é ' + email)
```

Conceitos Básicos

Operações Aritméticas

Em Python, podemos fazer cálculos de forma simples.

```
a = 10
b = 11

print('A soma resultou em:', a + b)
print('A subtração resultou em:', a - b)
print('A multiplicação resultou em:', a * b)
print('A divisão resultou em:', a / b)
print('O quociente da divisão resultou em:', a % b)
```

Conceitos Básicos

Operadores Relacionais

Podemos trabalhar também com Operações que para validar se uma informação é verdadeira ou falsa.

```
10 < 11 # 10 é menor que 11?  
1 > 2 # 1 é maior que 2?  
10 >= 10 # 10 é maior ou igual a 10?  
3 != 2 # 3 é diferente de 2?  
2 == 2 # 2 é igual a 2?
```

Conceitos Básicos

Operadores de Atribuição

Podemos atribuir valores para uma variável, de formas diferentes.

```
a = 3  
a = a + 7  
print(a)  
  
a += 5 # a = a + 5  
print(a)  
  
a -= 3 # a = a - 3  
print(a)  
  
a *= 2 # a = a * 2  
print(a)  
  
a /= 4 # a = a / 4  
print(a)  
  
a %= 4 # a = a % 4  
print(a)  
  
a **= 8 # a = a ** 8  
print(a)
```

Conceitos Básicos

Strings

```
# Trabalhando com Strings(Texto)
"Adriano Leite da Silva"
'Adriano Leite da Silva'
```

```
# Quando Trabalhar com Aspas Simples e Aspas Duplas
# Ambas servem para textos, mas quando tenho algo como o caso abaixo
# 'Caixa D'agua', tenho duas opcoes
"Caixa D'agua"
'Caixa D\'agua'
```

Conceitos Básicos

Fatiando Strings

```
# Fazendo o Fatiamento de uma String(Texto)
nome = 'Adriano Leite da Silva'

# Aqui sempre farei a chamada a var
# no caso nome
# e entre [] passarei as posições do texto que eu quero que mostre

nome[0] # Mostro a letra que está na posição 0, ou seja, A
nome[3] # Se eu mostrar a posição 3, será i
```

Conceitos Básicos

Fatiando Strings

```
# Fazendo o Fatiamento de uma String(Texto)
nome = 'Adriano Leite da Silva'

nome[3:] # Mostro a partir de uma posicao especifica
# No caso acima, pegamos a partir da posicao 3 e mostramos todo o restante

nome[:9] # Já nesse caso, pegamos todas as letras até a posição 9

nome[:-1] # Quando colocamos o :-1 pegamos todo o conteúdo menos a última posicao
# nome[-1:] serve para pegar a última posicao tbm

nome[-1] # -1 apenas, serve para pegar a última posicao
```

```
# Fazendo o Fatiamento de uma String(Texto)
numeros = '1234567890'

# Como imprimir na tela toda a minha string
numeros
numeros[:]

# Posso fazer a impressao dos valores com um passo, ou seja,
# pulando a quantidade de casas informadas
numeros[::2]
# Nesse caso ele deve imprimir 13579

# outro exemplo
numeros[1::2]
#nesse exemplo, ele deve comecar pelo numero 2 e imprimir 2468
```

```
frase = 'Python é legal demais'

# Fazendo verificações em um texto
'py' not in frase # Verifica se não temos isso na frase | Lembrando que Py é diferente de py
'ing'in frase # Deve retornar True

len(frase) # Para verificar o tamanho de uma var
frase.lower() # transformando meu texto em minúsculo
frase.upper() # transformando meu texto em maiúsculo

fraseMius = frase.lower()
fraseMaius = frase.upper()

fraseMaius
fraseMius
```

```
numeros = '1234567890'

numeros[::-1]
numeros[::-2]

nome = 'adriano'
nome[::-1]

frase =
'Python é uma linguagem excelente!'
frase.split('e')
frase.split('y')
```

Conceitos Básicos

Listas

```
lista = ['Banana', 'Abacaxi', 'Melancia', 'Uva', 'Pera', 'Kiwi']

print(lista) # Imprimindo todo o valor da lista
print(lista[0]) # Imprimindo a posicao 0
print(lista[2:]) # Imprimindo a partir de Melancia
```

Conceitos Básicos

Listas

```
# Para adicionar um novo item  
lista.append('Manga')  
print(lista)
```

```
# Para remover um item  
lista.remove('Manga')  
print(lista)
```

Conceitos Básicos

Listas

```
lista = ['Banana', 'Abacaxi', 'Melancia', 'Uva', 'Pera', 'Kiwi']

lista.index('Pera')
```



Obrigado!

