PYTHON ACADEMY

GUIA COMPLETO SOBRE LISTAS NO PYTHON

Esse ebook trata de um dos principais tipos de dados usados no dia a dia do Pythonista: as famosas LISTAS!

PYTHONACADEMY.COM.BR

Salve salve Pythonista!

Nesse post, vamos tratar de um tipo de dados muito importante da linguagem Python: as **Listas**!

Ela está presente no dia a dia do desenvolvedor Python e é muito importante para construção de códigos utilizando nossa querida linguagem.

Se prepare que o post está **C-O-M-P-L-E-T-O**!

Então já abre seu terminal para acompanhar o post e **vamos nessa**!

Ops, Fallout 😆

Introdução

Uma **Lista** (`list`) em Python, nada mais é que uma coleção ordenada de valores, separados por vírgula e dentro de colchetes `[]`.

Elas são utilizadas para armazenar diversos itens em uma única variável. Entender este conteúdo é de extrema importância para dominar a linguagem por completo!

Abaixo temos um exemplo de uma lista:

```
# Exemplo de lista:
lista = ['Python', 'Academy']

print(lista)
```

Saída do código acima:

```
['Python', 'Academy']
```

Podemos observar a classe de uma lista com `type()`:

```
lista = []

print(type(lista))
```

Saído do código acima:

```
<class 'list'>
```

Criando listas

Existem várias maneiras de se criar uma lista.

A maneira mais simples é envolver os elementos da lista por colchetes, por exemplo:

```
# Lista com apenas um elemento
lista = ["PythonAcademy"]
```

Também podemos criar uma lista vazia:

```
lista = []
```

Para criar uma lista com diversos itens, podemos fazer:

```
lista = ['Python', 'Academy', 2021]
```

Também podemos utilizar a função `list` do próprio Python (*built-in function*):

```
lista = list(["Python Academy"])
```

Outra forma é criar listas resultantes de uma operação de *List Comprehensions*!

> *List Comprehensions Não domina List Comprehensions? Então já clica aqui para ler nosso post completo sobre esse assunto!* 😉

```
[item for item in iteravel]
```

Podemos ainda criar listas através da função `range()`, dessa forma:

```
list(range(10))
```

O que resultará em:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Acessando dados da lista

Todos os itens de uma lista são indexados, ou seja para cada item da lista um índice é atribuído da seguinte forma: `lista[indice]`.

Exemplo com itens:

```
frutas = ['Maça', 'Banana', 'Jaca', 'Melão', 'Abacaxi']
```

E assim ficaria a sequência de índices:

Índice	0	1	2	3	4
Valores	Maçã	Banana	Jaca	Melão	Abacaxi

Em Python os índices são iniciados em 0.

Ou seja, como podemos acessar o primeiro item da lista que é o índice 0? **Veja abaixo**:

```
print(frutas[0])
```

A saída como previsível foi a string com a palavra `Maça` por ocupar o índice 0:

```
Maça
```

Agora vamos ver sobre **Indexação Negativa**!

Agora que se inscreveu, podemos seguir! 😉

Indexação negativa

> *E se o desejado for o último item?*

Neste momento entramos no conceito de **indexação negativa**, que significa começar do fim.

`-1` irá se referir ao último item. Por exemplo:

Índice	-5	-4	-3	-2	-1
Valores	Maçã	Banana	Jaca	Melão	Abacaxi

Dessa forma, para buscar pelo último item da lista:

```
print(frutas[-1])
```

Resultando em:

```
Abacaxi
```

Lista dentro de lista

Suponha que exista uma lista dentro de uma lista, assim:

```
lista = ['item1', ['python', 'Academy'], 'item3']
```

Como podemos acessar o primeiro índice do item que é uma lista?

A resposta é simples, basta selecionar a posição em que se localiza a lista para ter acesso a ela, assim:

```
sublista = lista[1]
print(sublista[0])
```

Ou ainda:

```
print(lista[1][0])
```

Ambos obtém mesmo resultado:

```
'python'
```

> 💡 *Estou construindo o **DevBook**, uma plataforma que usa IA para criar ebooks técnicos — com código formatado e exportação em PDF. Depois de ler, dá uma passada lá!*

Fatiando uma lista (*slicing*)

O fatiamento de listas, do inglês *slicing*, é a extração de um conjunto de elementos contidos numa lista. Ele é feito da seguinte forma:

```
lista[ inicio : fim : passo ]
```

Explicando cada elemento:

- `início` se refere ao índice de início do fatiamento.
- `fim` se refere ao índice final do fatiamento. A lista final não vai conter esse elemento.
- `passo` é um parâmetro opcional e é utilizado para se pular elementos da lista original

Vamos entender melhor em seguida!

Se quisermos criar uma fatia de uma lista do índice 2 ao 4, podemos fazer da seguinte forma:

```
lista = [10, 20, 30, 40, 50, 60]

print(lista[2:5])
```

O *slicing* conta a partir do índice 2 até o índice 5 (mas não o utiliza), pegando os índices 2, 3, 4.

Sua saída será:

```
[30, 40, 50]
```

Percorrendo listas

A forma mais comum de percorrer os elementos em uma lista é com um loop `for elemento in lista`, assim:

```
lista = [10, 20, 30, 40, 50, 60]

for num in lista:
    print(num)
```

Saída:

```
10
20
30
40
50
60
```

Com a função `enumerate()` podemos percorrer também o índice referente a cada valor da lista:

```
lista = [10, 20, 30, 40, 50, 60]

for indice, valor in enumerate(lista):
    print(f'índice={indice}, valor={valor}')
```

Sua saída será:

```
índice=0, valor=10
índice=1, valor=20
índice=2, valor=30
índice=3, valor=40
índice=4, valor=50
índice=5, valor=60
```

Que tal poupar algumas linhas e obter o mesmo resultado com *List Comprehension*?

```
[print(num) for num in lista]

# Com enumerate:
[print(f'índice={indice}, valor={valor}') for indice, valor in enumera-
        te(lista)]
```

A saída será a mesma! 😉

Métodos para manipulação de Listas

O Python tem vários métodos disponíveis em listas que nos permite manipulá-las.

Separamos esse conteúdo em outro post que você pode acessar agora clicando aqui!

Conclusão

Nesse post vimos todo o poder e flexibilidade das **Listas** do Python.

Dominá-las é muito importante e ajuda muito no dia a dia do desenvolvedor Python!

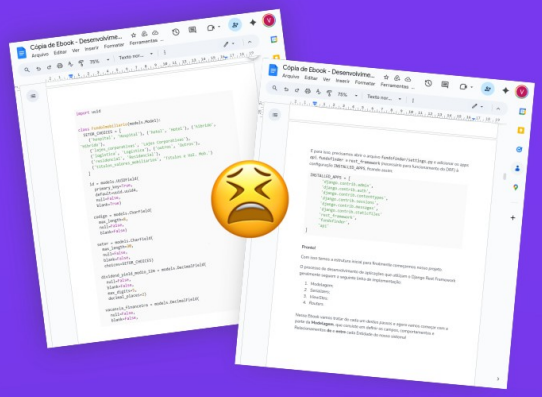
Nos vemos no próximo post, dev! 😉

Não se esqueça de conferir!

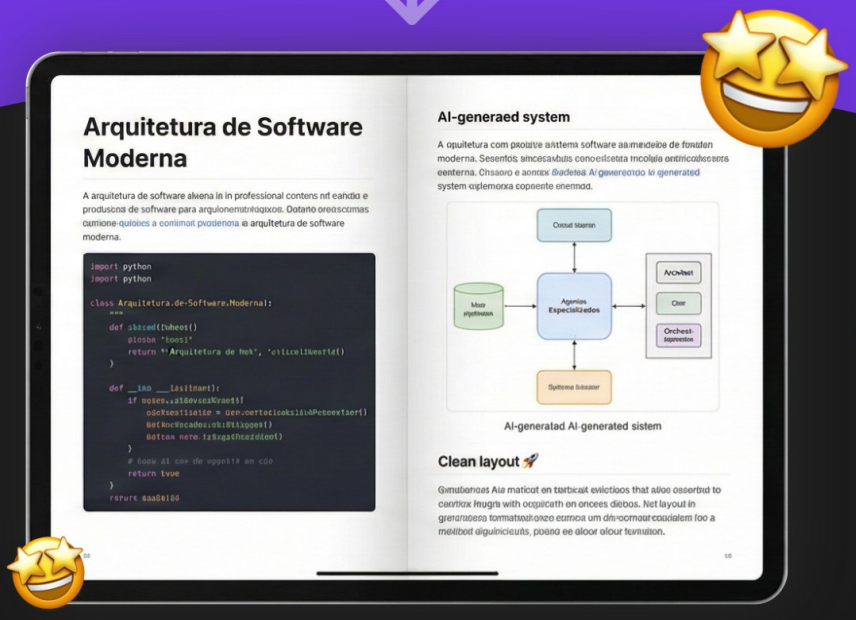
DevBook

Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código**!



Chega de formatar código no Google Docs



Syntax Highlight

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

TESTE AGORA

PRIMEIRO CAPÍTULO 100% GRÁTIS