

Codificação da informação e Estruturas de dados (Registros, Matrizes e Listas)

Oficina de Programação para o Ensino médio

Safira Soares & Igor Novaes

Porta Temperos = Lista ? Sim !

- **Porta temperos** são utilizados para **guardar diferentes tipos de temperos juntos** e de forma organizada
- Para identificar os temperos , cada recipiente possui um nome , na programação esses nomes são os **índices**



Como são as listas em Python ?

Em Python, **list** é uma estrutura de dados utilizada para armazenar elementos de forma **sequencial e ordenada**. As listas são acessadas por meio de um **índice** que inicia com o valor 0 para o primeiro elemento e é incrementada com 1 a cada novo item. Além disso, o último item da lista é representado pelo índice “-1”

Valores	Sal	Curry	Coentro	Orégano
Índices	0	1	2	3

Como criar e acessar os elementos de uma lista em Python ?

```
lista_temperos = ["Curry", "Sal", "Orégano", "Coentro"]  
  
print(lista_temperos)  
#resultado: ["Curry", "Sal", "Orégano", "Coentro"]  
  
print(lista_temperos[0])  
#resultado: Curry
```

Como percorrer uma lista ?



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

```
lista_temperos = ["Curry", "Sal", "Orégano", "Coentro"]  
  
for temp in lista_temperos:  
    print(temp)
```

```
#Resultado  
Curry  
Sal  
Orégano  
Coentro
```



Em python, podemos manipular listas através de diversos métodos

Veremos agora quais são os métodos e como utilizá-los!

Manipulando Listas

Método index()

Utilizado encontrar a posição de um valor especificado:

```
lista = ['Sal', 'Curry', 'Orégano', 'Coentro']  
  
pos = lista.index('Curry')  
  
print(f'0 item desejado está na posição: {pos}')
```

#Saída

```
0 item desejado está na posição: 1
```

Método count ()

Retorna o número de vezes que o valor especificado aparece na lista.

```
lista = ['Sal', 'Curry', 'Orégano', 'Curry']
```

```
pos = lista.count('Curry')
```

```
print(f'0 item desejado aparece: {pos}')
```

```
#Saída
```

```
0 item desejado aparece: 2
```


Método append()

Usado para adicionar um elemento ao final da lista:

```
lista = ['Sal', 'Curry', 'Orégano', 'Curry']
```

```
pos = lista.append('Cúrcuma')
```

```
print(lista)
```

```
#Saída
```

```
['Sal', 'Curry', 'Orégano', 'Curry', 'Cúrcuma']
```

Método insert()

Usado para adicionar um item em um índice especificado

```
lista = ['Sal', 'Curry', 'Orégano', 'Curry']
```

```
pos = lista.insert(0, 'Páprica')
```

```
print(lista)
```

#Saída

```
["Páprica", 'Sal', 'Curry', 'Orégano', 'Curry', "Cúrcuma"]
```

Método remove()

Usado para remover um item com valor especificado:

```
lista = ['Sal', 'Curry', 'Orégano', 'Curry']
```

```
pos = lista.remove('Orégano')
```

```
print(lista)
```

```
#Saída
```

```
['Sal', 'Curry', 'Curry']
```

Método reversed()

A função reversed() reverte a ordem da lista de entrada

```
lista = [1, 2, 3, 4, 7]

for item in reversed(lista):
    print(item)
```

#Saída

7

4

3

2

1

Método sorted()

É utilizada para ordenar a lista de entrada:

```
lista = [2, 4, 8, 1]

lista_ord = sorted(lista)
print(lista_ord)

#Saída
[1, 2, 4, 8]
```



Hora de praticar !

1. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.
2. Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.
3. Faça um Programa que leia 4 notas, mostre as notas e a média na tela.

Referências Bibliográficas

- ❑ [Vetores · Programando na Cozinha \(gitbooks.io\)](https://gitbooks.io)
- ❑ [Python list: armazenando itens em uma única variável! | Insights para te ajudar na carreira em tecnologia | Blog da Trybe \(betrybe.com\)](https://betrybe.com)
- ❑ [Manipulação de Listas no Python \(pythonacademy.com.br\)](https://pythonacademy.com.br)