



# Aula 17 - Variáveis compostas (Listas)

▼ Dificuldade	☆☆☆
☰ Tags	append del insert len pop remove sort

A maior diferença entre uma **tupla** e uma **lista** é que **tuplas são imutáveis**, já lista **não são**.

Para escrever em uma lista, basta colocar os elementos entre **colchetes [ ]**.

Se eu quiser **mudar o elemento três** de uma lista, basta fazer isso:

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']  
lanche[3] = 'picolé'
```

Na tupla, isso não seria possível, agora o **elemento** pudim vai ser **trocado** pelo elemento **picolé**.

## Método append

Como listas são **mutáveis**, podemos **adicionar elementos** depois delas definidas.

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']  
lanche.append('cookie')
```



## Método insert

Se quisermos adicionar um cachorro quente **antes** do lanche, usamos o **método insert()**.

Para usá-lo, primeiro **coloque a posição que você quer colocar o elemento** e depois **o próprio elemento**.

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']  
lanhce.insert(0, 'cachorro quente')
```

O próprio Python ira **refatorar** as **posições dos elementos**.



## Apagar elementos

É possível **apagar um elemento específico** da lista.

### comando Del

Basta colocar o **comando del** e em seguida o **elemento em índice fatiado**.

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']  
del lanche[3]
```

**Pudim** será **apagado da lista**.

## Método pop

Para usá-lo, é só colocar entre os parênteses a **posição que será apagada**.

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']  
lanche.pop(3)
```

Se ele não tiver **nada** entre parênteses, automaticamente o último elemento da lista será eliminado.

## Método remove

Coloque **entre parênteses** o próprio elemento.

```
lanche = ['lanche', 'suco', 'pizza', 'pudim']
lanche.remove('pudim')
```

Para **removermos um elemento** apenas que **esteja na lista**, podemos usar a estrutura **if** e o **in**.

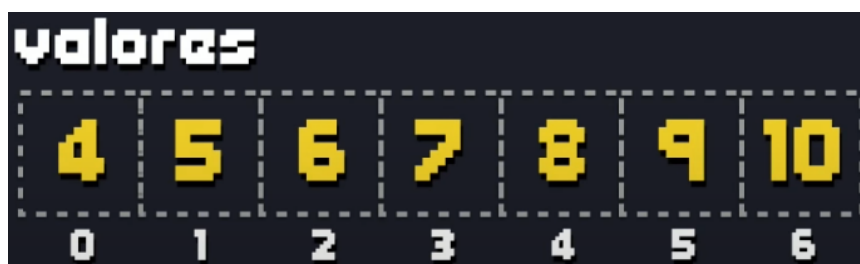
```
if 'pizza' in lanche:
    lanche.remove('pizza')
```

## Range

Usando **range** para criar listas.

```
valores = list(range(4,11))
```

Perceba que usamos o **método list**, que **criará a lista**, e dentro dela um **range**, que vai do **número 4 até o 10** (lembre-se que em ranges, o último número é desconsiderado).



## Método sort

Organizamos os valores usando o **método sort**.

```
valores = [8, 2, 5, 4, 9, 3, 0]
print(valores.sort())
```

Teremos como resultado 0, 2, 3, 4, 5, 8, 9.

Podemos organizar na **ordem inversa** também, para isso usamos `revert = True`.

```
valores = [8, 2, 5, 4, 9, 3, 0]
valores.sort(revert=True)
```

Como resultado, teremos 9, 8, 5, 4, 3, 2, 0.

## Função len

Podemos contar quantos elementos há na lista.

```
valores = [8, 2, 5, 4, 9, 3, 0]
len(valores)
```

Teremos como resultado **o número 7**.

## Recebendo valores do usuário

Podemos colocar uma variável que receba os valores e automaticamente os coloque na lista.

```
lista = []
lista.append(int(input('Digite um valor: 4')))
```

Podemos encontrar a **posição do índice** e o valor digitado com um **for** junto de um **enumerate**.

```
for i, c in enumerate(lista):
    print(f'Na posição {i} encontrei o valor {c}')
```

## Relação entre listas

Quando colocamos uma lista igual a outra, ela não estará apenas fazendo uma cópia, o Python **criará um relacionamento entre elas**.

```
a = [8, 4, 9, 10]
b = a
b.append(9)
print(a)
print(b)
```

Nesse caso, **a adição** virá para o b e para o a

Para criarmos apenas **uma cópia** de a em b, temos que usar o fatimanto [:]

```
a = [8, 4, 9, 10]
b = a[:]
b.append(9)
print(a)
print(b)
```

Agora sim, **a adição** do elemento 9 **ocorrerá apenas em b**, pois **não existe uma relação** entre eles.

## Desafio 78

Faça um programa que leia 5 valores numéricos e guarde-os em uma lista.

No final, mostre qual foi o maior e o menor valor digitado e as suas respectivas posições na lista.

```
valores = []
for cont in range(0, 5):
    valores.append(int(input(f'Digite um valor para a posição {cont}: ')))

print(f'O menor valor digitado foi {min(valores)} nas posições ', end='')
for i, c in enumerate(valores):
    if c == min(valores):
        print(f'{i}... ', end='')
print(f'\nO maior valor digitado foi {max(valores)} nas posições ', end='')
for i, c in enumerate(valores):
    if c == max(valores):
        print(f'{i}... ', end='')
```

<https://youtu.be/q8Z1cRdJnfk>

## Desafio 79

Crie um programa onde o usuário possa digitar vários valores numéricos e cadastre-os em uma lista. Caso o número já exista lá dentro, ele não será

adicionado. No final, serão exibidos todos os valores únicos digitados, em ordem crescente.

```
num = []
resp = ''

while True:
    n = (int(input('Digite um número: ')))
    if n not in num:
        print('Valor adicionado com sucesso...')
        num.append(n)
    else:
        print('Valor duplicado! Não irei adicionar...')
    resp = input('Quer continuar? [S/N]: ') .upper().strip()[0]
    if resp in 'N':
        break

print('-=' * 40)
print(f'Você digitou os números {sorted(num)}')
```

[https://youtu.be/LkAzRnc\\_GPk](https://youtu.be/LkAzRnc_GPk)

## Desafio 80

Crie um programa onde o usuário possa digitar cinco valores numéricos e cadastre-os em uma lista, já na posição correta de inserção (sem usar o `sort()`).

No final, mostre a lista ordenada na tela.

```
lista = []
for c in range(0, 5):
    n = int(input('Digite um número: '))
    if c == 0 or n > lista[-1]:
        lista.append(n)
        print('Adicionando no final da lista!')
    else:
        pos = 0
        while pos < len(lista):
            if n <= lista[pos]:
                lista.insert(pos, n)
                print(f'Adicionando na posição {pos} da lista!')
                break
            pos += 1
print(f'Os valores digitados em ordem foram {lista}')
```

<https://youtu.be/QDpwjBYRcVE>

## Desafio 81

Crie um programa que vai ler vários números e colocar em uma lista.

Depois disso, mostre:

1. Quantos números foram digitados.
2. A lista de valores, ordenada de forma decrescente.
3. Se o valor 5 foi digitado e está ou não na lista.

```
lista = []
while True:
    lista.append(int(input('Digite um número: ')))
    r = input('Você quer continuar? [S/N]: ') .upper().strip()[0]
    if r in 'N':
        break
print(f'Foram digitados {len(lista)} números.')
if 5 in lista:
    print(f'O número cinco foi digitado e está na posição {lista.index(5)}.')
else:
    print(f'O número cinco não foi digitado!')
lista.sort(reverse=True)
print(f'Os números digitados de forma decrescente: {lista}')
```

<https://youtu.be/SXJKAVVlvGA>

## Desafio 82

Crie um programa que vai ler vários números e colocar em uma lista.

Depois disso, crie duas listas extras que vão conter apenas os valores pares e os valores ímpares digitados, respectivamente.

Ao final, mostre o conteúdo das três listas geradas.

```
num = []
imp = []
par = []
```

```

while True:
    num.append(int(input('Digite um número: ')))
    r = input('Você quer continuar? [S/N]: ') .strip().upper()[0]
    if r in 'N':
        break

print(f'Todos os números digitados: {num}')
for c in num:
    if c % 2 == 0:
        par.append(c)
    else:
        imp.append(c)
print(f'Os números pares digitados: {par}')
print(f'Os números ímpares digitados: {imp}')

```

[https://youtu.be/uk0gDFQEo\\_I](https://youtu.be/uk0gDFQEo_I)

## Desafio 83

Crie um programa onde o usuário digite uma expressão qualquer que use parênteses. Seu aplicativo deverá analisar se a expressão passada está com os parênteses abertos e fechados na ordem correta.

```

exp = input('Digite uma expressão matemática: ')
pilha = []
for c in exp:
    if c == '(':
        pilha.append(c)
    else:
        if len(pilha) > 0:
            pilha.pop()
        else:
            pilha.append(c)
            break
if len(pilha) > 0:
    print('A expressão é inválida!')
else:
    print('A expressão é válida!')

```

<https://youtu.be/dvhP41Z7TLk>