



Aula 16 - Variáveis compostas (tuplas)

Dificuldade	☆☆
Tags	count del fatiamento for imutável index len max min sorted tuplas índice

Toda **variável simples** é um **espaço na memória**, que vai ser preenchido com alguma coisa.

Se eu tiver uma **variável de nome lanche**, e coloco um x-burguer lá, o espaço que estava livre agora foi reservado para o x-burguer.

Mas se eu quiser beber um **suco** também, a variável lanche recebe suco, o espaço que antes pertencia ao lanche vai ser **trocado** pelo suco.

Para resolver esse problema, teríamos que criar uma **variável que suporte mais valores**, que tenha mais espaços para serem preenchidos.

Até o momento, só usamos variáveis simples. A **variável composta** pode ter o **número de espaços que você quiser**, basta definir.

Há três tipos de variáveis compostas, as tuplas, as listas e o dicionário, hoje, iremos usar as tuplas.

Índice

Cada elemento de uma variável composta é representado por um **índice**, que **começa no 0**.



Uma **String** é uma **variável composta**, por isso que cada caractere é formada por um índice também.

Fatiamento

Se quisermos mostrar apenas a pizza, basta **fatiar a variável**.

```
print(lanche[2])
```

Ou se quisermos **fatiar do hambúrguer até a pizza**, basta fazer isso:

```
print(lanche[0:3])
```



O último elemento de um range no fatiamento é ignorado, portanto, o pudim, nesse caso, não será mostrado.

Para fatiar até o final, **sem saber** qual é o final.

```
print(lanche[1:])
```

Será mostrado o suco até o pudim.

Função len

Podemos saber quantos elementos há na variável usando o len.

```
print(len(lanche))
```

Será retornado o número 4.

Repetição for

O for pode ser utilizado em uma variável composta, podemos substituir o range pelo lanche.

```
for c in lanche:  
    print(c)
```

Nesse caso, c é uma variável simples, ela começará com o primeiro elemento da variável lanche, que no caso é o hambúrguer, a cada repetição, a variável c armazenará o próximo elemento da variável lanche, até os elementos acabarem, aí a estrutura de repetição for irá acabar também.



As tuplas são imutáveis, você não consegue mudar uma tupla, uma vez que você a define, ela ficará assim até o final do programa.

Prática

Para criar uma tupla, é preciso colocar parênteses entre os elementos.

```
lanche = ('Hamburger', 'Suco', 'Pizza', 'Pudim')
```

Se quisermos fatiar a variável.

```
lanche = ('Hamburger', 'Suco', 'Pizza', 'Pudim')  
print(lanche[0])
```

Nesse caso, apenas o elemento Hamburger irá aparecer.

Prova de que tuplas são imutáveis

Para provar que tuplas são imutáveis

```
lanche = ('Hambúrguer', 'Suco', 'Pizza', 'Pudim')  
lanche[1] = 'Pastel'  
print(lanche[1])
```

Esse programa irá dar erro, pois tuplas não podem ser mudadas depois de definidas.

For

Usando um for e substituindo o range pela variável lanche

```
lanche = ('Hamburguer', 'Suco', 'Pizza', 'Pudim')
for c in lanche:
    print(f'Hoje eu vou comer {c}!')
print('Comi pra caramba!')
```

Método sorted

Se eu desejar mostrar a variável lanche em ordem alfabética, usamos o método sorted.

```
lanche = ('Hamburguer', 'Suco', 'Pizza', 'Pudim')
print(sorted(lanche))
```

Somando tuplas

Se fizermos a soma de duas tuplas, elas irão se concatenar.

```
a = (1, 2, 3)
b = (5, 3, 7, 10)
c = a + b
print(c)
```

O resultado será 1, 2, 3, 5, 3, 7, 10.

Método count

Se quisermos contar quantas vezes determinado elemento aparece, usamos o método interno count

```
a = (1, 2, 3)
b = (5, 3, 7, 10)
c = a + b
print(c.count(3))
```

No caso o print irá resultar em 2, pois o 3 aparece duas vezes.

Método index

Para acharmos a posição no índice de determinado elemento, usamos o método interno index.

```
a = (1, 2, 3)
b = (5, 3, 7, 10)
c = a + b
print(c)
print(c.index(3))
```

Teremos como resultado o número 2, que é onde o número 3 fica.

Diferentes tipos de elementos

No Python, as mesmas tuplas podem armazenar diferentes tipos de elementos, como números inteiros, números reais e letras.

```
info = ('Ian', 17, 'M', 50.90)
print(info)
```

Função min() e max()

Para descobrirmos qual o menor e o maior valor de uma tupla, usamos a função min() e max().

```
n = (9, 3, 1, 10, 11)
print(max(n))
print(min(n))
```

Deletando tupla

Podemos até apagar a variável da memória.

```
info = ('Ian', 17, 'M', 50.90)
del(info)
```

Desafio 72

Crie um programa que tenha uma tupla totalmente preenchida com uma contagem por extenso, de zero até vinte.

Seu programa deverá ler um número pelo teclado (entre 0 e 20) e mostrá-lo por extenso.

```
n = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)
nextenso = ('zero', 'um', 'dois', 'três', 'quatro', 'cinco', 'seis', 'sete', 'oito', 'nove', 'dez', 'onze', 'doze', 'treze',
            'quatorze', 'quinze', 'dezesesseis', 'dezesesete', 'dezoito', 'dezenove', 'vinte')
nd = 2
error = ''

while True:
    nd = int(input(f'{error}Digite um número entre 0 e 20: '))
    if nd not in n:
        error = 'Tente novamente. '
    else:
        error = ''
        break

print(f'Você digitou o número {nextenso[nd]}')
```

<https://youtu.be/ei2Kr3ccfOO>

Desafio 73

Crie uma tupla preenchida com os 20 primeiros colocados da Tabela do Campeonato Brasileiro de Futebol, na ordem de colocação. Depois mostre:

1. Apenas os 5 primeiros colocados.
2. Os últimos 4 colocados da tabela.
3. Uma lista com os times em ordem alfabética.
4. Em que posição na tabela está o time do Bahia.

```
tabela = ('Flamengo', 'Internacional', 'Atlético-MG', 'São Paulo', 'Fluminense', 'Grêmio',
          'Palmeiras', 'Santos', 'Athletico-PR', 'Bragantino', 'Ceará SC', 'Corinthians',
          'Atlético-GO', 'Bahia', 'Sport Recife', 'Fortaleza', 'Vasco da Gama', 'Goiás',
          'Coritiba', 'Botafogo')
print(f'Lista da tabela do Brasileirão: {tabela}')
print(f'Os primeiros cinco colocados da tabela: {tabela[0:5]}')
print(f'Os últimos 4 colocados: {tabela[16:]}')
print(f'Tabela em ordem alfabética: {sorted(tabela)}')
print(f'O Bahia está na {tabela.index('Bahia')+1}ª posição')
```

<https://youtu.be/RexybLcGewA>

Desafio 74

Crie um programa que vai gerar cinco números aleatórios e colocar em uma tupla.

Depois disso, mostre a listagem de números gerados e também indique o menor e o maior valor que estão na tupla.

```
from random import randint

n = (randint(0, 10), randint(0, 10), randint(0, 10), randint(0, 10), randint(0, 10))
print(f'Os números sorteados foram: ', end='')
for c in n:
    print(f'{c} ', end='')
print(f'\nO maior número sorteado é {max(n)}')
print(f'O menor número sorteado é {min(n)}')
```

<https://youtu.be/mlwt2CRQkTQ>

Desafio 75

Desenvolva um programa que leia quatro valores pelo teclado e guarde-os em uma tupla. No final, mostre:

1. Quantas vezes o valor 9 apareceu.
2. Em que posição foi digitado o primeiro valor 3.
3. Quais foram os números pares.

```
n = (int(input('Digite um valor: ')),
      int(input('Digite outro valor: ')),
      int(input('Digite mais um valor: ')),
      int(input('Digite o último valor: ')))

print(f'O número 9 apareceu {n.count(9)} vezes.')
if n.count(3) >= 1:
    print(f'Você digitou o número 3 na {n.index(3)+1}º vez.')
else:
    print('O número 3 não foi digitado nenhuma vez.')
print(f'Os números que foram digitados que são pares: ', end='')
for c in n:
    if c % 2 == 0:
        print(f'{c} ', end='')
    else:
        print(' ', end='')
print('')
```

<https://youtu.be/1u7oA8ckjAc>

Desafio 76

Crie um programa que tenha uma tupla única com nomes de produtos e seus respectivos preços, na sequência.

No final, mostre uma listagem de preços, organizando os dados em forma tabular.

```
lista = ('Leite', 3.5, 'Chocolate', 9.25, 'Macarrão', 6.5, 'Lápis', 1.25, 'Caderno', 15.98)
print('-'*40)
```

```
print('{:^40}'.format('LISTAGEM DE PREÇO'))
print('-'*40)
for c in lista:
    if c == str(c):
        print(f'{c:<30}', end='')
    else:
        print(f'R${c:6}')
```

<https://youtu.be/Qp2cXfCHK2l>

Desafio 77

Crie um programa que tenha uma tupla com várias palavras (não usar acentos). Depois disso, você deve mostrar, para cada palavra, quais são as suas vogais.

```
palavras = ('Celular', 'Monitor', 'Computador', 'Garrafa', 'Fone')
for c in palavras:
    print(f'\nNa palavra {c.upper()} temos as vogais: ', end='')
    for letras in c:
        if letras.lower() in 'aeiou':
            print(f'{letras} ', end='')
```

<https://youtu.be/8BgSqrOpKvU>