



# Aula 19 - Variáveis compostas (Dicionários)

▼ Dificuldade	★★
☰ Tags	<code>.copy</code> <code>.items</code> <code>.values</code> <code>del</code> <code>for</code> <code>itemgetter</code> <code>keys</code>

Se eu quiser mostrar determinado índice, basta dar print no número do índice.

Com o **dicionário** é possível identificar esse índice com um nome.

Para se criar um dicionário, coloque os dados entre colchetes {}.

Vamos criar uma lista com dois elementos, um nome e uma idade.

```
dados = []
dados.append('Pedro')
dados.append(25)
print(dados[0])
print(dados[1])
```



Perceba que o nome é representado pelo 0 e a idade pelo 1.

Mas é possível mudar o número por nomes.

```
dados = {'nome': 'Pedro', 'idade': 25}
print(dados['nome'])
print(dados['idade'])
```

Agora invés de números, podemos usar nomes que representam melhor os dados.

## Adicionar elementos

Basta atribuímos o valor no dicionário.

```
dados['sexo'] = 'M'
```

## Remover elementos

Basta usarmos o comando **del** junto com o índice.

```
dados['sexo'] = 'M'
del dados['idade']
```

Iremos criar um dicionário com informações de um filme.

```
filme = {'titulo': 'Star Wars', 'ano': 1977, 'diretor': 'George Lucas'}
```

## Método values

Para darmos print em todos os **valores**, usamos o método interno `values`

```
filme = {'titulo': 'Star Wars', 'ano': 1977, 'diretor': 'George Lucas'}  
print(filme.values())
```

## Método keys

Para darmos print em **todas as chaves**, usamos o método interno `keys`.

```
filme = {'titulo': 'Star Wars', 'ano': 1977, 'diretor': 'George Lucas'}  
print(filme.keys())
```

## Método items

Para pegarmos **tanto os itens como as chaves**, usamos o método interno `items`.

```
filme = {'titulo': 'Star Wars', 'ano': 1977, 'diretor': 'George Lucas'}  
print(filme.items())
```

## Estrutura de repetição for

Podemos usar uma **estrutura de repetição** para **pegar as chaves e os itens** e darmos um print. Para isso, usamos o método `items`.

```
filme = {'titulo': 'Star Wars', 'ano': 1977, 'diretor': 'George Lucas'}  
for k, v in filme.items():  
    print(f'O {k} é {v}.')
```

A variável **k** é a **chave** e o **v** é o **valor**.

```
O titulo é Star Wars  
O ano é 1977  
O diretor é George Lucas
```

## Integrando uma lista dentro de um dicionário

Podemos dar um **append** de um dicionário para **dentro de uma lista**, assim, cada elemento da lista é um dicionário.

## Método copy

Mas para fazer uma **cópia** de um dicionário para a lista, temos que usar o método interno `copy`, invés do **fatiamento** `[:]`.

Por exemplo, podemos **criar um dicionário** `locadora` e **colocá-lo dentro de uma lista** com as informações de filmes.

```
filme = {}
locadora = []
for c in range(0, 3):
    filme['uf'] = input('Digite o nome do filme: ')
    filme['sigla'] = input('Digite a sigla desse estado: ')
    locadora.append(filme.copy())
print(locadora)
```



Índice	titulo	ano	diretor
0	'Star Wars'	1977	'George Lucas'
1	'Avengers'	2012	'Joss Whedon'
2	'Matrix'	1999	'Wachowski'

Perceba que os **elementos da lista** são identificados por **números** e os **elementos do dicionário** são identificados por **texto**.

Para darmos **print em elementos específicos**, podemos **referenciar o número do índice da lista** e depois a **chave**.

```
print(locadora[0]['ano'])
```

O resultado mostrará **1977**, que está **dentro do índice 0** dentro da **chave 'ano'**.

```
print(locadora[2]['titulo'])
```

O resultado mostrará **'Matrix'**, que está **dentro do índice 2**, dentro da **chave 'titulo'**.

Se quisermos **referenciar itens** em uma frase, basta usar o **fatiamento**.

```
peessoas = {'nome': 'Ian', 'idade': 17, 'sexo': 'M', }
print(f'0 {peessoas["nome"]} tem {peessoas["idade"]} anos.')
```

## Função itemgetter

Para mostrar um dicionário **ordenado pelo valor**, teremos que usar uma função chamada `itemgetter`, temos que **importá-la de uma biblioteca**.

Criaremos uma lista para essa lista ordenada, dentro dela **colocaremos um sorted**, com os **itens do dicionário** e como **chave do sorted**, o **itemgetter com a posição do valor a ser ordenado**.

Para colocar do **maior para o menor**, basta colocar um `reverse=True`.

```
from operator import itemgetter
jogadores = {'jogador1': randint(1, 6),
             'jogador2': randint(1, 6),
             'jogador3': randint(1, 6),
             'jogador4': randint(1, 6)}
lista = sorted(jogadores.items(), key=itemgetter(1), reverse=True)
for i, c in enumerate(lista):
    print(f'{i+1}º lugar: {c[0]} tirou {c[1]}')
```

## Desafio 90

Faça um programa que leia nome e média de um aluno, guardando também a situação em um dicionário. No final, mostre o conteúdo da estrutura na tela.

```
dados = {}
dados['Nome'] = input('Digite o nome do aluno: ')
dados['Média'] = float(input('Digite a média do aluno: '))
if dados['Média'] >= 7:
    dados['Situação'] = 'APROVADO'
else:
    dados['Situação'] = 'REPROVADO'
for k, v in dados.items():
    print(f'{k} é igual a {v}.')
```

<https://youtu.be/HipQYUk4koA>

## Desafio 91

Crie um programa onde 4 jogadores joguem um dado e tenham resultados aleatórios. Guarde esses resultados em um dicionário. No final, coloque esse dicionário em ordem, sabendo que o vencedor tirou o maior número no dado.

```
from random import randint
from operator import itemgetter
cont = 1
jogadores = {'jogador1': randint(1, 6), 'jogador2': randint(1, 6), 'jogador3': randint(1, 6), 'jogador4': randint(1, 6)}
for k, v in jogadores.items():
    print(f'O {k} tirou {v}.')
print('O ranking dos jogadores é: ')
valores = sorted(jogadores.values(), reverse=True)
lista = sorted(jogadores.items(), key=itemgetter(1), reverse=True)
for i, c in enumerate(lista):
    print(f'{i+1}º lugar: {c[0]} tirou {c[1]}')
```

<https://youtu.be/cwrglZtaAwk>

## Desafio 92

Crie um programa que leia nome, ano de nascimento e carteira de trabalho e cadastre-os (com idade) em um dicionário.

Se por acaso a CTPS for diferente de zero, o dicionário receberá também o ano de contratação e o salário. Calcule e acrescente, além da idade, com quantos anos a pessoa vai se aposentar.

```
from datetime import date
dados = {}
dados['Nome'] = input('Digite seu nome: ')
anonasc = int(input('Digite o seu ano de nascimento: '))
dados['idade'] = date.today().year - anonasc
dados['CTPS'] = int(input('Digite o número da sua carteira de trabalho [0 se não tiver]: '))
if dados['CTPS'] != 0:
    dados['contratacao'] = 0
    error = ''
    while dados['contratacao'] < anonasc+14:
        dados['contratacao'] = int(input(f'{error}Digite o ano de sua contratação: '))
        error = 'Você não pode trabalhar com menos de 14 anos, tente novamente...\n'
```

```

dados['aposentadoria'] = dados['contratacao'] + 35 - anonasc
dados['salário'] = float(input('Digite o seu salário mensal: R$'))
for k, v in dados.items():
    print(f'{k} tem o valor {v}.')

```

<https://youtu.be/Vsqemzdrj78>

## Desafio 93

Crie um programa que gerencie o aproveitamento de um jogador de futebol. O programa vai ler o nome do jogador e quantas partidas ele jogou. Depois vai ler a quantidade de gols feitos em cada partida. No final, tudo isso será guardado em um dicionário, incluindo o total de gols feitos durante o campeonato.

```

dados = {}
gols = []
v = 1
dados['nome'] = input('Nome do jogador: ')
n = int(input(f'Quantas partidas {dados["nome"]} jogou? '))
for c in range(0, n):
    g = (int(input(f'    Quantas gols na partida {c+1}? ')))
    gols.append(g)
dados['total_gols'] = sum(gols)
dados['gols'] = gols[:]
for cont in gols:
    print(f'Na partida {v}, {dados["nome"]} fez {cont} gols.')
    v += 1
print(f'Foi um total de {dados["total_gols"]} gols em {n} partidas.')

```

<https://youtu.be/5yKiud-YNaE>

## Desafio 94

Crie um programa que leia nome, sexo e idade de várias pessoas, guardando os dados de cada pessoa em um dicionário e todos os dicionários em uma lista. No final, mostre:

1. Quantas pessoas foram cadastradas.
2. A média de idade do grupo.
3. Uma lista com todas as mulheres.
4. Uma lista com todas as pessoas com idade acima da média.

```

dados = []
pessoa = {}
soma_idade = 0
mulheres = []
idade_acima = []
while True:
    pessoa['nome'] = input('Nome: ')
    while True:
        pessoa['sexo'] = input('Sexo: ').strip().upper()[0]
        if pessoa['sexo'] in 'MF':
            break
        print('ERRO, responda apenas com M ou F.')
    if pessoa['sexo'] in 'F':
        mulheres.append(pessoa['nome'])
    pessoa['idade'] = int(input('Idade: '))
    soma_idade += pessoa['idade']
    dados.append(pessoa)
    continuar = input('Quer continuar? [S/N] ').strip().upper()[0]
    if continuar in 'N':
        break
media_idade = soma_idade / len(dados)
print(f'A média de idade é {media_idade:.1f} anos.')
print(f'As mulheres cadastradas foram: {mulheres}.')
print(f'Lista das pessoas que estão acima da média:')
for p in dados:
    if p['idade'] > media_idade:
        print(p)

```

```

soma_idade += pessoa['idade']
dados.append(pessoa.copy())
pessoa.clear()
while True:
    r = input('Quer continuar? [S/N]: ') .strip().upper()[0]
    if r in 'SN':
        break
    print('ERRO, responda apenas com S ou N!')
if r in 'N':
    break
print(dados)
media_idade = soma_idade/len(dados)
print(f'{len(dados)} pessoas foram cadastradas.')
print(f'A média de idade das pessoas cadastradas é {media_idade:.1f} anos.')
print(f'Todas as mulheres cadastradas: {mulheres}.')
print(f'Lista de pessoas que estão acima da média de idade:\n ')
for p in dados:
    if p['idade'] > media_idade:
        print(p)
        print()

```

<https://youtu.be/ETnExBCFeps>

## Desafio 95

Aprimore o DESAFIO 093 para que ele funcione com vários jogadores, incluindo um sistema de visualização de detalhes do aproveitamento de cada jogador.

```

dados = []
jogador = {}
gols = []
cont = 0
while True:
    jogador['nome'] = input('Nome do jogador: ')
    n = int(input(f'Quantas partidas {jogador["nome"]} jogou? '))
    for c in range(0, n):
        gols.append(int(input(f'Quantas gols na partida {c+1}? ')))
    jogador['cod'] = cont
    cont += 1
    jogador['gols'] = gols.copy()
    dados.append(jogador.copy())
    gols.clear()
    r = input('Quer continuar? [S/N]: ') .upper()[0]
    print('-'*28)
    if r in 'N':
        break
print(f'{"cod":<4} {"nome":<10} {"gols":<10} {"total de gols":>10}')
for k, v in enumerate(dados):
    print(f'{"dados[k]":<4} {"dados[k]":<10} {"str(dados[k]":<20}{sum(dados[k]":<20}')

r = 0
print('-'*20)
while True:
    while True:
        r = int(input('Mostrar dados de qual jogador? '))
        if r > k and r != 999:
            print(f'ERRO! Não existe jogador com o código {r}! Tente novamente.')
        else:
            break
    if r == 999:
        break
    print('-'*30)
    print(f'-LEVANTAMENTO DO JOGADOR {dados[r]":<10}')
    for n, c in enumerate(dados[r]":<10}')
    print(f'Na partida {n+1} fez {c} gols')
    print('-'*30)

```

<https://youtu.be/mw1So0r317Y>

---