

Aula 14

Operador Ternário

O operador ternário é usado para associar um valor ou outro baseado em uma condição. Ele segue a seguinte sintaxe :

```
<valor> if <condição> else <outro valor>
```

Abaixo está um exemplo de uso do operador ternário :

```
1 idade = int(input('Digite sua idade: '))
2
3 pode_votar = True if idade >= 18 else False
4
5 if pode_votar:
6     print(f'Você já pode votar, pois já tem {idade} anos.')
7 else:
8     print(f'Você ainda não pode votar, pois ainda tem {idade} anos.')
```

A estrutura pode ser encadeada com diversos "if...else" em linha, mas isso pode ficar muito confuso.

```
1 nota = 95
2
3 resultado = 'Nota = A+' if nota > 90 else 'Nota = A'
4             if nota > 80 else 'Nota = B' if nota > 70 else 'Nota = C'
5             if nota > 60 else 'Nota = D' if nota > 40 else
6             'Reprovado'
7
8 print(resultado)
```

Abaixo temos a mesma estrutura usando parênteses para vermos melhor os grupos de "if...else"

```
1 nota = 95
2
3 resultado = ('Nota = A+' if nota > 90 else ('Nota = A'
4             if nota > 80 else ('Nota = B' if nota > 70 else ('Nota = C'
5             if nota > 60 else ('Nota = D' if nota > 40 else
6             'Reprovado')))))
7
8 print(resultado)
```

Agora com os parêntesis alinhados para melhor visualização

```
1 nota = 95
2
3 resultado = (
4     'Nota = A+' if nota > 90 else (
5         'Nota = A' if nota > 80 else (
6             'Nota = B' if nota > 70 else (
7                 'Nota = C' if nota > 60 else (
8                     'Nota = D' if nota > 40 else 'Reprovado'
9                 )
10            )
11        )
12    )
13 )
14
15 print(resultado)
```

Mais alguns exemplos da forma tradicional e da forma ternária:

```
1 idade = 18
2
3 if idade >= 18:
4     print("Você pode votar!")
5 else:
6     print("Você ainda não pode votar.")
```

```
1 idade = 18
2 mensagem = "Você pode votar." if idade >= 18 else "Você ainda não pode votar."
3 print(mensagem)
```

```
1 nota = 7.5
2 if nota >= 7:
3     print("Você foi aprovado!")
4 else:
5     print("Você foi reprovado.")
```

```
1 nota = 7.5
2
3 resultado = "aprovado" if nota >= 7 else "reprovado"
4
5 print(f"Você foi {resultado}.")
```

```
1 numero = 10
2
3 if numero % 2 == 0:
4     print("O número é par.")
5 else:
6     print("O número é ímpar.")
```

```
1 numero = 10
2
3 resultado = "par" if numero % 2 == 0 else "ímpar"
4
5 print(f"O número é {resultado}.")
```

Compreensão de Listas (List Comprehensions)

Compreensão de listas é uma maneira concisa de criar listas. Muitas aplicações criam listas onde cada elemento resulta de alguma operação aplicada a cada item de outra sequência ou iterável, ou criar uma sublista de elementos que satisfazem determinada condição.

```
1 # forma 'tradicional'
2 quadrados = []
3
4 for x in range(10):
5     quadrados.append(x**2)
6
7 print(quadrados)
```

Note que desse modo cria (ou sobrescreve) uma variável chamada de 'x' que ainda existe depois da repetição terminar. Nós podemos calcular a lista de quadrados sem efeito colateral usando compreensão de listas, que é mais concisa e mais fácil de ler.

```
1 # forma 'concisa'
2 quadrados = [x**2 for x in range(10)]
3
4 print(quadrados)
```

A compreensão de listas consiste em colchetes '[]' contendo uma expressão seguido de uma cláusula 'for' e então zero ou mais cláusulas 'for' ou 'if'. O resultado será uma nova lista resultante da verificação da expressão no contexto das cláusulas 'for' e 'if' que seguem.

Por exemplo : essa comparação de listas combina elementos de duas listas se eles não forem iguais.

```
1 # compreensão de listas
2 resultado = [(x, y) for x in [1,2,3] for y in [3,1,4] if x != y]
3 print(resultado)
4
5 # equivalente
6 resultado = []
7 for x in [1,2,3]:
8     for y in [3,1,4]:
9         if x != y:
10             resultado.append((x, y))
11 print(resultado)
```

Repare que a ordem das cláusulas 'for's e 'if' são as mesmas em ambos os códigos.

```
1 # cria a lista de inteiros
2 lista = [-4, -2, 0, 2, 4]
3
4 # cria uma nova lista com os valores duplicados
5 print(f'Lista duplicada: {[x*2 for x in lista]}')
6 # filtra a lista para excluir os valores negativos
7 print(f'Lista positivos: {[x for x in lista if x >= 0]}')
8 # aplica uma função a todos os elementos
9 print(f'Lista absolutos: {[abs(x) for x in lista]}')
10 # chama um método em cada elemento da lista
11 frutas = ['banana', 'laranja', 'bergamota']
12 print(f'Frutas sem espaços: {[fruta.strip() for fruta in frutas]}')
13 # cria uma lista de tuplas com dois valores como (numero, quadrado)
14 print(f'Tuplas: {[x, x**2] for x in range(10)}')
15 # a tupla precisa estar entre parênteses, senão ocorre o erro:
16 # SyntaxError: f-string: did you forget parentheses around the comprehension target?
17 # junta uma lista de listas em outra lista com dois 'for'
18 listas = [[1,2,3], [4,5,6], [7,8,9]]
19 print(f'Uma lista: {[num for lista in listas for num in lista]}')
```

Compreensão de listas pode conter expressões complexas e funções aninhadas.

```
1 from math import pi
2
3 print(f'Decimais do PI: {[str(round(pi, i)) for i in range(1, 6)]}')
```

Compreensão de Listas Aninhadas

A expressão inicial de uma compreensão de lista pode ser qualquer expressão, incluindo outra compreensão de lista.

Abaixo vamos transpor as linhas e colunas de uma matriz.

```
1 matriz = [
2     [1, 2, 3, 4],
3     [5, 6, 7, 8],
4     [9, 10, 11, 12],
5 ]
6
7 transposta = [[linha[coluna] for linha in matriz] for coluna in range(len(matriz[0]))]
8
9 print(f'Matriz transposta: {transposta}')
```

Como vimos, a compreensão de lista mais interna é resolvida no contexto do 'for' mais externo. Vamos desmontar para ver como fica:

```
1 matriz = [
2     [1, 2, 3, 4],
3     [5, 6, 7, 8],
4     [9, 10, 11, 12],
5 ]
6
7 transposta = []
8 for coluna in range(len(matriz[0])):
9     transposta.append([linha[coluna] for linha in matriz])
10
11 print(f'Matriz transposta: {transposta}')
```

```
1 matriz = [
2     [1, 2, 3, 4],
3     [5, 6, 7, 8],
4     [9, 10, 11, 12],
5 ]
6
7 transposta = []
8 for coluna in range(len(matriz[0])):
9     transposta_linha = []
10    for linha in matriz:
11        transposta_linha.append(linha[coluna])
12    transposta.append(transposta_linha)
13
14 print(f'Matriz transposta: {transposta}')
```

Dicionários

Outra estrutura de dados muito útil embutida em Python é o dicionário, cujo tipo é "dict". Dicionários são também chamados de “memória associativa” ou “vetor associativo” em outras linguagens. Diferente de sequências (tuplas e listas) que são indexadas por inteiros, dicionários são indexados por chaves (keys), que podem ser de qualquer tipo imutável (como strings e inteiros). Tuplas também podem ser chaves se contiverem apenas strings, inteiros ou outras tuplas. Se a tupla contiver, direta ou indiretamente, qualquer valor mutável, não poderá ser chave. Listas não podem ser usadas como chaves porque podem ser modificadas internamente pela atribuição em índices ou fatias, e por métodos como `append()` e `extend()`.

```
1 contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
```

Um bom modelo mental é imaginar um dicionário como um conjunto não-ordenado de pares "*chave:valor*", onde as chaves são únicas em uma dada instância do dicionário. Dicionários são delimitados por chaves: {}, e contém uma lista de pares "*chave:valor*" separada por vírgulas. Dessa forma também será exibido o conteúdo de um dicionário no console do Python. O dicionário vazio é {}.

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>>
>>> print(contato)
{'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
```

As principais operações em um dicionário são armazenar e recuperar valores a partir de chaves. Também é possível remover um par "*chave:valor*" com o comando `del`. Se você armazenar um valor utilizando uma chave já presente, o antigo valor será substituído pelo novo. Se tentar recuperar um valor usando uma chave inexistente, será gerado um erro.

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>>
>>> print(contato['nome'])
João
>>> print(contato)
{'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> del contato['telefone']
>>> print(contato)
{'nome': 'João', 'idade': 25}
>>> contato['nome'] = 'Carlos'
>>> print(contato)
{'nome': 'Carlos', 'idade': 25}
>>> print(contato['telefone'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'telefone'
>>>
```

Podemos adicionar uma nova *chave:valor* ao dicionário simplesmente adicionando um valor a uma chave inexistente.

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> print(contato)
{'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> contato['altura'] = 1.8
>>> print(contato)
{'nome': 'João', 'telefone': '1234-5678', 'idade': 25, 'altura': 1.8}
>>>
```

Executar `list(dicionário)` em um dicionário devolve a lista de todas as chaves presentes no dicionário, na ordem de inserção (se desejar ordená-las basta usar a função `sorted()`).

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> print(contato)
{'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> print(list(contato))
['nome', 'telefone', 'idade']
>>> print(sorted(list(contato)))
['idade', 'nome', 'telefone']
>>>
```

Para verificar a existência de uma chave, use o operador `in`.

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>> print('nome' in contato)
True
>>> print('peso' in contato)
False
>>>
```

Podemos usar uma lista para armazenar diversos dicionários.

```
1 agenda = [
2     {'nome': 'João', 'telefone': '1234-5678', 'idade': 25},
3     {'nome': 'Maria', 'telefone': '1234-5678', 'idade': 27},
4     {'nome': 'José', 'telefone': '1234-5678', 'idade': 30},
5 ]
```

Iterando sobre um Dicionário

O Python possui 3 métodos muito úteis para trabalhar com dicionários.

O método **`keys()`** retorna uma lista com as chaves do dicionário. Esse método pode ser utilizado para verificar se uma chave específica está presente no dicionário, por exemplo:

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>>
>>> for chave in contato.keys():
...     print(f'a chave é {chave}')
...
a chave é nome
a chave é telefone
a chave é idade
>>>
```

O método **`values()`** retorna uma lista com os valores do dicionário. Esse método pode ser utilizado para verificar se um valor específico está presente no dicionário, por exemplo:

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>>
>>> for valor in contato.values():
...     print(f'o valor é {valor}')
...
o valor é João
o valor é 1234-5678
o valor é 25
>>>
```

O método **items()** retorna uma lista contendo pares chave-valor do dicionário, em forma de tuplas, ou seja, cada item retornado é uma tupla com dois elementos, o primeiro sendo a chave e o segundo o valor correspondente. Esse método é útil para percorrer o dicionário em um loop for, por exemplo:

```
>>> contato = {'nome': 'João', 'telefone': '1234-5678', 'idade': 25}
>>>
>>> for chave, valor in contato.items():
...     print(f'{chave} : {valor}')
...
nome : João
telefone : 1234-5678
idade : 25
>>>
```

Sugestões VS Code:

Configuração:

- Format On Save: auto formata o arquivo Python;
- Format On Type: formata a linha automaticamente após digitação;

Exercícios para Praticar

Exercícios Operador Ternário

1. Crie uma variável "idade" e use o operador ternário para atribuir a string "maior de idade" se a idade for maior ou igual a 18, ou "menor de idade" caso contrário.
2. Crie uma variável "temperatura" e use o operador ternário para atribuir a string "quente" se a temperatura for maior do que 30 graus Celsius, ou "frio" caso contrário.
3. Crie uma variável "nota" e use o operador ternário para atribuir a string "aprovado" se a nota for maior ou igual a 7, ou "reprovado" caso contrário.
4. Crie uma variável "numero" e use o operador ternário para atribuir a string "positivo" se o número for maior do que zero, ou "negativo ou zero" caso contrário.
5. Crie uma variável "time" e use o operador ternário para atribuir a string "grêmio" se a string for igual a "Grêmio", ou "Inter" caso contrário.
6. Crie uma variável "mes" e use o operador ternário para atribuir a string "fevereiro" se a variável "numero_mes" for igual a 2, ou "outro mês" caso contrário.
7. Crie uma variável "idade" e use o operador ternário para atribuir a string "idoso" se a idade for maior ou igual a 60 anos, ou "não idoso" caso contrário.
8. Crie uma variável "salario" e use o operador ternário para atribuir a string "alto" se o salário for maior do que R\$ 10.000, ou "baixo ou médio" caso contrário.
9. Crie uma variável "cor" e use o operador ternário para atribuir a string "verde" se a variável tipo for igual a "amigo", ou "vermelho" caso contrário.
10. Crie uma variável "numero" e use o operador ternário para atribuir a string "par" se o número for divisível por 2, ou "ímpar" caso contrário.
11. Crie uma variável "idade" e use o operador ternário para atribuir a string "jovem" se a idade for menor do que 30 anos, ou "não jovem" caso contrário.
12. Crie uma variável "altura" e use o operador ternário para atribuir a string "alta" se a altura for maior do que 1,80 metros, ou "baixa ou média" caso contrário.
13. Crie uma variável "pessoa" e use o operador ternário para atribuir a string "adulto" se a variável idade for maior ou igual a 18 anos e menor do que 60 anos, ou "jovem" ou "idoso" caso contrário.
14. Crie uma variável "sabor" e use o operador ternário para atribuir a string "chocolate" se a variável tipo for igual a "bolo", ou "outro sabor" caso contrário.
15. Crie uma variável "numero" e use o operador ternário para atribuir a string "maior do que 10" se o número for maior do que 10, ou "menor ou igual a 10" caso contrário.
16. Crie uma variável "genero" e use o operador ternário para atribuir a string "plural" se a variável quantidade for maior do que 1, ou "singular" caso contrário.
17. Crie uma variável "idade" e use o operador ternário para atribuir a string "criança" se a idade for menor do que 12 anos, "adolescente" se a idade estiver entre 12 e 17 anos, ou "adulto" caso contrário.
18. Crie uma variável "nota1" e "nota2" e use o operador ternário para atribuir a string "aprovado" se a média das notas for maior ou igual a 7, ou "reprovado" caso contrário.
19. Crie uma variável "idade" e use o operador ternário para atribuir a string "maior de 65 anos" se a idade for maior ou igual a 65 anos, ou "menor do que 65 anos" caso contrário.
20. Crie uma variável "valor" e use o operador ternário para atribuir a string "positivo" se o valor for maior do que zero, "negativo" se o valor for menor do que zero, ou "zero" caso contrário.

Exercícios Compreensão de Listas

1. Crie uma lista com os números de 1 a 10 e imprima-a na tela.
2. Crie uma lista com os nomes de três pessoas e imprima a primeira pessoa da lista.
3. Crie uma lista com os números de 1 a 5 e multiplique cada número por 2, imprimindo o resultado.
4. Crie uma lista com os números de 1 a 10 e imprima apenas os números pares.
5. Crie uma lista com os números de 1 a 5 e some todos os elementos da lista.
6. Crie uma lista com os números de 1 a 10 e imprima apenas os números ímpares.
7. Crie uma lista com os nomes de cinco países e ordene-os em ordem alfabética.
8. Crie uma lista com os números de 1 a 10 e encontre o maior número da lista.
9. Crie uma lista com os números de 1 a 10 e encontre o menor número da lista.

10. Crie uma lista com os números de 1 a 10 e some apenas os números ímpares.
11. Crie uma lista com os números de 1 a 10 e some apenas os números pares.
12. Crie uma lista com os números de 1 a 5 e ordene-os em ordem decrescente.
13. Crie uma lista com as letras 'a', 'b', 'c', 'd' e imprima a lista em ordem reversa.
14. Crie uma lista com os números de 1 a 10 e imprima apenas os números divisíveis por 3.
15. Crie uma lista com os números de 1 a 5 e multiplique cada número por 3, imprimindo o resultado.
16. Crie uma lista com os números de 1 a 10 e encontre a média dos valores da lista.
17. Crie uma lista com as palavras 'casa', 'carro', 'computador', 'celular' e ordene-as em ordem alfabética inversa.
18. Crie uma lista com os números de 1 a 5 e eleve cada número ao quadrado, imprimindo o resultado.
19. Crie uma lista com os números de 1 a 10 e imprima apenas os números que são divisíveis por 2 e 3 simultaneamente.
20. Crie uma lista com os números de 1 a 5 e imprima apenas os números maiores do que 2.
21. Crie uma lista com os nomes de quatro frutas e imprima todas as frutas que começam com a letra "m".
22. Crie uma lista com os números de 1 a 10 e encontre o segundo maior número da lista.
23. Crie uma lista com os números de 1 a 5 e calcule a soma cumulativa dos elementos da lista.
24. Crie uma lista com os nomes de três animais e imprima-os em ordem inversa.
25. Crie uma lista com os números de 1 a 10 e imprima apenas os números que são múltiplos de 2 e 5 simultaneamente.
26. Crie uma lista com os números de 1 a 10 e calcule o produto de todos os elementos da lista.
27. Crie uma lista com os números de 1 a 5 e calcule a média dos valores da lista.
28. Crie uma lista com os números de 1 a 10 e imprima apenas os números que são divisíveis por 4.
29. Crie uma lista com os números de 1 a 5 e imprima a lista em ordem aleatória.
30. Crie uma lista com os nomes de cinco cidades e imprima a cidade que possui o maior nome.
31. Crie uma lista com os números de 1 a 10 e encontre o terceiro menor número da lista.
32. Crie uma lista com os números de 1 a 5 e calcule a soma acumulada dos elementos da lista, mas começando do final da lista.
33. Crie uma lista com os números de 1 a 10 e calcule a soma dos valores da lista, excluindo o primeiro e o último elemento.
34. Crie uma lista com os números de 1 a 10 e encontre a mediana dos valores da lista.
35. Crie uma lista com os números de 1 a 5 e eleve cada número a ele mesmo (potência), imprimindo o resultado.
36. Crie uma lista com os nomes de três cores e imprima todas as cores em ordem crescente de tamanho.
37. Crie uma lista com os números de 1 a 10 e imprima apenas os números que são primos.
38. Crie uma lista com os números de 1 a 5 e calcule a soma cumulativa dos elementos da lista, mas pulando um elemento sim e outro não.
39. Crie uma lista com os números de 1 a 10 e encontre o número mais frequente na lista.
40. Crie uma lista com os números de 1 a 5 e imprima apenas os números maiores do que a média da lista.

Exercícios Dicionário

1. Crie um dicionário com o nome e a idade de três pessoas e imprima o nome e a idade da segunda pessoa.
2. Adicione um novo par chave-valor ao dicionário criado no exercício 1 com o nome e a idade de uma quarta pessoa.
3. Crie um dicionário com o nome e o salário de quatro funcionários e calcule a média salarial.
4. Crie um dicionário com as notas de um aluno em três provas e calcule a média.
5. Crie um dicionário com o nome e o número de telefone de três amigos e remova o amigo com o número de telefone mais baixo.
6. Crie um dicionário com o nome e o número de telefone de cinco pessoas e verifique se uma pessoa específica está na lista.
7. Crie um dicionário com os nomes de três países e suas respectivas capitais e imprima a capital do segundo país.
8. Crie um dicionário com os nomes de três frutas e seus preços e imprima o preço da segunda fruta.
9. Crie um dicionário com o nome e a idade de cinco pessoas e ordene as pessoas por idade, da mais nova para a mais velha.

10. Crie um dicionário com o nome e a altura de cinco pessoas e imprima o nome da pessoa mais alta.
11. Crie um dicionário com o nome e o endereço de três restaurantes e adicione o horário de funcionamento de cada um.
12. Crie um dicionário com o nome e o preço de três produtos e calcule o total da compra de uma pessoa que comprou um de cada.
13. Crie um dicionário com o nome e a nota de três alunos e verifique quem obteve a maior nota.
14. Crie um dicionário com os nomes de três países e seus idiomas oficiais e verifique se um determinado idioma é falado em um dos países.
15. Crie um dicionário com o nome e a profissão de quatro pessoas e remova a pessoa que tem a profissão de "estudante".
16. Crie um dicionário com o nome e o peso de cinco pessoas e ordene as pessoas por peso, do mais leve para o mais pesado.
17. Crie um dicionário com o nome e o sobrenome de três pessoas e verifique se um determinado sobrenome está na lista.
18. Crie um dicionário com o nome e a cidade de três times de futebol e adicione a quantidade de títulos que cada time tem.
19. Crie um dicionário com o nome e o gênero de três personagens de filmes e imprima o gênero do segundo personagem.
20. Crie um dicionário com o nome e a idade de três animais de estimação e calcule a idade total em anos.
21. Crie um dicionário com o nome, a idade e o endereço de três pessoas e verifique quantas pessoas têm a mesma idade.
22. Crie um dicionário com o nome e a nota de dez alunos e calcule a média de cada um, armazenando os resultados em outro dicionário.
23. Crie um dicionário com o nome e o preço de cinco produtos e ordene os produtos por preço, do mais barato para o mais caro.
24. Crie um dicionário com o nome e a idade de cinco pessoas e remova todas as pessoas com idade ímpar.
25. Crie um dicionário com o nome e a altura de cinco pessoas e calcule a média de altura das pessoas do sexo masculino e feminino separadamente.
26. Crie um dicionário com o nome e o ano de lançamento de cinco filmes e ordene os filmes por ordem cronológica.
27. Crie um dicionário com o nome e a quantidade de visualizações de três vídeos do YouTube e verifique qual vídeo tem o maior número de visualizações.
28. Crie um dicionário com o nome e a nacionalidade de cinco artistas e remova todos os artistas que não são brasileiros.
29. Crie um dicionário com o nome e a quantidade de seguidores de cinco perfis no Instagram e calcule a média de seguidores.
30. Crie um dicionário com o nome e a idade de cinco pessoas e ordene as pessoas por idade, do mais velho para o mais novo.
31. Crie um dicionário com o nome e a altura de cinco pessoas e verifique se há duas pessoas com a mesma altura.
32. Crie um dicionário com o nome e a data de nascimento de cinco pessoas e calcule a idade de cada uma.
33. Crie um dicionário com o nome e a pontuação de cinco jogadores de um jogo e verifique qual jogador tem a maior pontuação.
34. Crie um dicionário com o nome e a cidade de três times de futebol e adicione o número de jogadores que cada time tem.
35. Crie um dicionário com o nome e o preço de cinco produtos e verifique quantos produtos custam mais de R\$50,00.
36. Crie um dicionário com o nome e a idade de cinco pessoas e calcule a média de idade.
37. Crie um dicionário com o nome e a altura de cinco pessoas e verifique se a média de altura das pessoas é maior que 1,70m.
38. Crie um dicionário com o nome e a quantidade de seguidores de cinco perfis no Instagram e ordene os perfis por número de seguidores, do maior para o menor.

39. Crie um dicionário com o nome e a idade de cinco pessoas e verifique se há pelo menos duas pessoas com idade superior a 30 anos.
40. Crie um dicionário com o nome e a nota de cinco alunos e calcule a média das notas, removendo a nota mais baixa e a nota mais alta.

Exercícios sobre o método `items()`:

1. Crie um dicionário com o nome e idade de três pessoas e utilize o método `items()` para exibir os pares chave-valor.
2. Utilize o método `items()` para verificar se um determinado par chave-valor está presente em um dicionário.
3. Crie uma função que receba um dicionário como parâmetro e utilize o método `items()` para retornar uma lista com as chaves do dicionário em ordem alfabética.
4. Crie um programa que solicite ao usuário que informe as notas de um aluno em três disciplinas e armazene em um dicionário, utilizando o nome da disciplina como chave. Em seguida, utilize o método `items()` para exibir as notas e média do aluno.
5. Crie uma função que receba um dicionário como parâmetro e utilize o método `items()` para retornar uma lista com os valores do dicionário em ordem crescente.

Exercícios sobre o método `values()`:

1. Crie um dicionário com o nome e idade de três pessoas e utilize o método `values()` para exibir somente as idades.
2. Utilize o método `values()` para verificar se um determinado valor está presente em um dicionário.
3. Crie um programa que solicite ao usuário que informe as notas de um aluno em três disciplinas e armazene em um dicionário, utilizando o nome da disciplina como chave. Em seguida, utilize o método `values()` para calcular a média do aluno.
4. Crie uma função que receba um dicionário como parâmetro e utilize o método `values()` para retornar uma lista com os valores do dicionário em ordem decrescente.
5. Crie um dicionário com o nome e idade de três pessoas e utilize o método `values()` para calcular a média das idades.

Exercícios sobre o método `keys()`:

1. Crie um dicionário com o nome e idade de três pessoas e utilize o método `keys()` para exibir somente os nomes.
2. Utilize o método `keys()` para verificar se uma determinada chave está presente em um dicionário.
3. Crie uma função que receba um dicionário como parâmetro e utilize o método `keys()` para retornar uma lista com as chaves do dicionário em ordem decrescente.
4. Crie um programa que solicite ao usuário que informe as notas de um aluno em três disciplinas e armazene em um dicionário, utilizando o nome da disciplina como chave. Em seguida, utilize o método `keys()` para exibir as disciplinas em ordem alfabética.
5. Crie um dicionário com o nome e idade de três pessoas e utilize o método `keys()` para contar quantos elementos o dicionário possui.

Exercícios com múltiplos métodos:

1. Crie um dicionário com o nome e idade de três pessoas e utilize os métodos `items()` e `values()` para exibir os pares chave-valor e as idades.
2. Crie um programa que solicite ao usuário que informe as notas de um aluno em três disciplinas e armazene em um dicionário, utilizando o nome da disciplina como chave. Em seguida, utilize os métodos `items()` e `values()` para exibir as notas e média do aluno.
3. Crie um dicionário com o nome e idade de três pessoas e utilize os métodos `keys()` e `values()` para exibir os nomes e idades.
4. Crie um programa que solicite ao usuário que informe as notas de um aluno em três disciplinas e armazene em um dicionário, utilizando o nome da disciplina como chave. Em seguida, utilize os métodos `keys()` e `items()` para exibir as disciplinas e notas do aluno.
5. Crie um dicionário com o nome, idade e cidade de três pessoas e utilize os métodos `items()` e `values()` para exibir os pares chave-valor e os valores de idade e cidade.