

## Módulo 1 - Laboratório prático – Exercícios Versão 2

1. Um tuplo (*tuple*) é uma sequência imutável de elementos ou uma sequência de elementos imutáveis? Ilustre a sua resposta com exemplos em código Python.

2. Criar funções para executar as seguintes operações.

Documentação sobre o tipo de dados *tuple* do Python (sugestão):  
[https://www.tutorialspoint.com/python/python\\_tuples.htm](https://www.tutorialspoint.com/python/python_tuples.htm)

- a) Calcular a distância Euclidiana entre dois pontos ponto1 = (x1, y1) e ponto 2= (x2, y2).  
Recorde que a distância Euclidiana entre dois pontos é dada pela fórmula:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- b) Converter coordenadas cartesianas para coordenadas polares. A função recebe o tuplo (x,y) e devolve o tuplo (r,θ).

3. Criar funções para executar as seguintes operações.

Documentação sobre o tipo de dados *list* do Python (sugestão):  
[https://www.tutorialspoint.com/python/python\\_lists.htm](https://www.tutorialspoint.com/python/python_lists.htm)

- a) Dada uma lista de números inteiros e dois números inteiros i e s, que definem um intervalo [i, s], contar quantos números da lista pertencem ao intervalo.

- i. sem usar *comprehension*
- ii. usando *comprehension*.

- b) Devolver uma lista de resultados de divisões de pares (x, y), sendo os pares dados através de uma lista de tuplos. Deve evitar a divisão por zero e só deve dividir caso a divisão seja maior ou igual a 1.

Exemplo:

list\_res\_div((82, 40), (23, 4), (12, 3), (15, 3), (1, 2), (20, 0)) → [2.05, 5.75, 4.0, 5.0, 0.5, None]

- i. sem usar *comprehension*
- ii. usando *comprehension*.

- c) Dadas duas listas, devolver uma lista de pares, juntando cada elemento da 1.<sup>a</sup> lista com o elemento da 2.<sup>a</sup> na mesma posição, ordenada pelo segundo elemento. Exemplo:

`lst_triplos([1, 5, 2, 3], [20, 56, 10, 22]) → [(2, 10), (1, 20), (3, 22), (5, 56)]`

4. Criar funções para executar as seguintes operações.

Documentação do tipo de dados *set* do Python (sugestão):  
[https://www.w3schools.com/python/python\\_sets.asp](https://www.w3schools.com/python/python_sets.asp)

- a) Devolver os elementos comuns entre dois tuplos dados. Exemplo: `tuple_intersect((-5, -1, 4), (-5, -2, 4, 5)) → {-5, 4}`

- i. sem usar *comprehension*
- ii. usando *comprehension*.

- b) Remover duplicados de uma dada lista. Exemplo: para `l1 = [4, 5]` e `l2 = [3, 4]`, `remove_duplicates(l1, l2) → [3, 5]`

5. Criar funções para executar as seguintes operações.

Documentação do tipo de dados *dict* do Python (sugestão):  
[https://www.tutorialspoint.com/python/python\\_dictionary.htm](https://www.tutorialspoint.com/python/python_dictionary.htm)

- a) Dada uma string `s`, por exemplo um texto relativo a uma notícia, contar o número de ocorrências de cada palavra num dicionário. A chave do dicionário deve ser a palavra em causa, e o valor, a sua respetiva frequência no texto.

- b) Dado o seguinte dicionário:

```
people_dict = {
    "Alice": {"age": 35, "sex": "female", "occupation": "teacher"},
    "Bob": {"age": 42, "sex": "male", "occupation": "engineer"},
    "Charlie": {"age": 28, "sex": "male", "occupation": "scientist"},
    "Diana": {"age": 31, "sex": "female", "occupation": "doctor"},
    "Eva": {"age": 26, "sex": "female", "occupation": "artist"},
    "Frank": {"age": 38, "sex": "male", "occupation": "writer"},
}
```

```
"Grace": {"age": 40, "sex": "female", "occupation": "programmer"},  
"Henry": {"age": 33, "sex": "male", "occupation": "musician"},  
"Isabelle": {"age": 29, "sex": "female", "occupation": "designer"},  
"John": {"age": 45, "sex": "male", "occupation": "businessman"}  
}
```

- i. Devolver a idade do Bob.
- ii. Devolver o conjunto das diferentes ocupações das pessoas.
- iii. Remover do dicionário todas as pessoas com idades inferiores a 30 anos.
- iv. Devolver a média de todas as idades, após a remoção das pessoas com idades inferiores a 30.

6. Criar funções para executar as seguintes operações.

- a) Devolver uma lista de resultados de divisões de pares (x, y), sendo os pares dados através de uma lista de tuplos. Deve evitar a divisão por zero e só deve dividir caso a divisão seja maior ou igual a 1. Exemplo: `list_res_div((82, 40), (23, 4), (12, 3), (15, 3), (1, 2))` → `[2.05, 5.75, 4.0, 5.0, 0.5]`

- i. sem usar *comprehension*
- ii. usando *comprehension*.

7. Usando uma única linha de código *Python*, crie funções para:

- a) Devolver a matriz transposta de uma matriz dada.

Documentação do tipo de dados *numpy.array* do Python (sugestão):  
[https://numpy.org/doc/stable/user/absolute\\_beginners.html#](https://numpy.org/doc/stable/user/absolute_beginners.html#)

- b) Devolver a matriz inversa de uma da matriz dada.

Documentação do tipo de dados *numpy.array* do Python, em particular das operações de álgebra linear (sugestão): <https://numpy.org/doc/stable/reference/routines.linalg.html#>