

**Aluna: Laura Damasceno de Campos**

**Matrícula: 2020020021**

**Elaborar 2 questões objetivas alinhadas com os 2 objetivos de aprendizagem:**

1. Definir tipos de dados compostos como Tupla, Lista, Dicionário;
2. Usar as estruturas básicas de programação com Listas, Tuplas e Dicionários.

Pergunta relacionada ao objetivo 1: Os tipos de dados compostos permitem armazenar múltiplos itens dentro de uma única unidade. String, tuplas, listas, dicionários e sets são alguns exemplos de tipos compostos. A seguir, todas as opções estão corretas, exceto:

- a) Tanto a lista como a tupla são tipos de sequências de dados, mas uma é mutável e a outra imutável.
- b) Imutável refere-se a um objeto com valor fixo. Objetos imutáveis incluem strings, números e tuplas.
- c) Listas e tuplas possuem muitas propriedades comuns, mas a diferença básica é que a lista é imutável e a tupla é mutável.
- d) Dicionários são similares a outros tipos compostos, exceto por eles poderem usar qualquer tipo imutável de dados como índice.

Pergunta relacionada ao objetivo 2: Suponhamos que você deseja programar um sistema de agenda de contatos telefônicos em Python. Para isso, é preciso armazenar os números dos contatos. A princípio, podemos pensar em usar uma lista: telefones = ['1234-5678', '1212-1212', '3434-3434']. Mas, qual o sentido de termos uma lista de números soltos? O ideal seria conectar os telefones aos seus respectivos contatos. A tupla pode nos ajudar com isso: contato = ('Laura', '1234-5678'). Para não precisarmos de uma variável para cada contato, podemos colocá-los direto em uma lista de contatos: contatos\_lista = [('Laura', '1234-5678'), ('Maria', '1212-1212'), ('Amelia', '3434-3434')]. Até agora temos uma lista de contatos em que, ao menos, cada contato tem seu nome e telefone conectados. Entretanto, por enquanto, só conseguimos acessar um contato individualmente pela sua posição na lista, e não pelo seu próprio nome. O ideal seria mapear o nome de cada contato com seu telefone, evitando outros problemas. Para isso, podemos utilizar a estrutura dicionário. Considerando esse fato, assinale abaixo a única opção incorreta:

- a) Para não ter que redigitar os dados de contatos que já colocamos em nossa lista de contatos, uma opção seria criar um dicionário usando sua função construtora `dict()` e passando, como parâmetro, uma lista de tuplas, como em nosso caso: `contatos = dict(contatos_lista) // print(contatos)`, que retornará: `{'Laura': '1234-5678', 'Maria': '1212-1212', 'Amelia': '3434-3434'}`.
- b) Podemos acessar os valores do dicionário de forma similar a como acessamos os valores de uma lista, por exemplo. Um modo seria: `print(contatos['Laura'])`, que retornaria `1234-5678`.
- c) Considerando o exemplo do enunciado, podemos verificar se um contato está em nosso dicionário através da palavra chave `in`. Se eu digitar `print('Amelia' in contatos)`, a resposta será `True`.
- d) Suponhamos que você gostaria de adicionar um novo número ao seu dicionário de contatos. Para isso, basta utilizar o método `append()`, como nas listas.