

EXERCÍCIOS – REVISÃO DE ESTRUTURAS DE DADOS EM PYTHON

- 1) Tuplas são sequências imutáveis, que são utilizadas para armazenar coleções de itens. Dentre as alternativas abaixo, quais mostram formas de se criar uma tupla?

- a) `tupla([1, 2, 3])`
- b) `(1, 2, 3)`
- c) `tuple([1, 2, 3])`
- d) `1, 2, 3`

- 2) Considere o código abaixo e os seguintes códigos de seleção:

```
carros = (  
    (  
        'Jetta Variant',  
        'Motor 4.0 Turbo',  
        2003,  
        False,  
        ('Rodas de liga', 'Travas elétricas', 'Piloto automático')  
    ),  
    (  
        'Passat',  
        'Motor Diesel',  
        1991,  
        True,  
        ('Central multimídia', 'Teto panorâmico', 'Freios ABS')  
    )  
)
```

Quais chamadas produzem os seguintes resultados?

a) 'Freios ABS'

b) ('Rodas de liga', 'Travas elétricas')

- 3) Considerando a mesma estrutura da questão anterior, O que precisamos fazer para imprimir todos os acessórios que aparecem?. O resultado desejado é o seguinte:

Rodas de liga

Travas elétricas

Piloto automático

Central multimídia

Teto panorâmico

Freios ABS

- 4) A função `zip()` permite gerar um iterador de tuplas, como no exemplo abaixo:

```
nomes = ['Passat', 'Crossfox']
valores = [100000.0, 75000.0]
list(zip(nomes, valores))
saída: [('Passat', 100000.0), ('Crossfox', 75000.0)]
```

Considerando as duas listas abaixo:

```
nomes = ['Passat', 'Crossfox', 'DS5', 'C4', 'Jetta']
kms = [15000, 12000, 32000, 8000, 50000]
```

Escreva um trecho de código que possibilite a impressão dos nomes dos veículos com quilometragem abaixo de 20.000 km.

- 5) Tendo como base as duas listas abaixo:

```
nomes = ['Passat', 'Crossfox', 'DS5', 'C4', 'Jetta']
kms = [15000, 12000, 32000, 8000, 50000]
```

Escreva um trecho de código para criar um dicionário a partir destas listas.

- 6) Utilize o dicionário: dados abaixo para responder à questão:

```
dados = {
    'Passat': {
        'ano': 2012,
        'km': 50000,
        'valor': 75000,
        'acessorios': ['Airbag', 'ABS']
    },
    'Crossfox': {
        'ano': 2015,
        'km': 35000,
        'valor': 25000
    }
}
```

Escreva trechos de código que produza as seguintes respostas:

- a) Testar se a chave `acessorios` existe no dicionário de informações do carro ((Resposta: False))
- b) Testar se a chave `acessorios` existe no dicionário de informações do carro `Passat` (Resposta: True)
- c) Obter o valor do carro `Crossfox` (Resposta: 25000)
- d) Acessar o último acessório do carro `Passat` (Resposta: 'ABS')

- 7) Considere o dicionário abaixo:

```
dados = {'Jetta': 88000, 'Crossfox': 72000, 'DS5': 124000}
```

Escreva o comando para atualizar o dicionário de forma que o mesmo fique da seguinte forma:

```
{ 'Jetta': 88000, 'Crossfox': 72000, 'DS5': 124000, 'Passat': 85000, 'Fusca': 150000 }
```

- 8) Considere o dicionário abaixo:

```
dados = {
    'Crossfox': {'valor': 72000, 'ano': 2005},
```

```
'DS5': {'valor': 125000, 'ano': 2015},  
'Fusca': {'valor': 150000, 'ano': 1976},  
'Jetta': {'valor': 88000, 'ano': 2010},  
'Passat': {'valor': 106000, 'ano': 1998}  
}
```

Crie um trecho de código que imprime somente os nomes dos veículos com ano de fabricação maior ou igual a 2000.

9) Com relação as built-in functions, assinale a alternativa incorreta:

- a) type() - Retorna o tipo de um objeto
- b) dict() - Cria um dicionário
- c) tupla() - Cria uma tupla
- d) zip() - Retorna um iterador de tuplas

10) Considere o conjunto de informações abaixo:

```
dados = {  
    'Crossfox': {'km': 35000, 'ano': 2005},  
    'DS5': {'km': 17000, 'ano': 2015},  
    'Fusca': {'km': 130000, 'ano': 1979},  
    'Jetta': {'km': 56000, 'ano': 2011},  
    'Passat': {'km': 62000, 'ano': 1999}  
}
```

$Km_media = km_total / (ano_atual - ano_fabricação)$

Complete o corpo da função:

```
def km_media(dataset, ano_atual):  
    ...
```

a função acima obtém e imprime a quilometragem média anual de cada veículo em um dicionário com a estrutura do dicionário “dados” acima:

Ex. de chamada da função:

km_media(dados, 2019)

Saída:

```
2500.0  
4250.0  
3250.0  
7000.0  
3100.0
```