

## Exercícios da Semana 2: ANDREA SAMPAIO ELIAS

**(PARTE 1)** Vamos entender as estruturas dos dados em Python:

1) Qual a diferença entre uma variável tipo int e uma tipo float?

O int representa uma variável de valor inteiro (positivo ou negativo), já o float representa uma variável de valor real. Ou seja, a diferença entre eles é a precisão.

2) O que é uma variável do tipo string?

Strings são cadeias de caracteres, armazenam dados textuais, ou seja, letras, palavras ou frases.

**(PARTE 2)** Condicionais, Loops, Funções:

3) Crie uma função que encontre o maior valor entre 3 números imputados pelo usuário.

```
#Questão 3
a=input('Digite o primeiro valor: ')
b=input('Digite o segundo valor: ')
c=input('Digite o terceiro valor: ')
y=0

if a > b:
    if a > c:
        x=a;
    elif c > a:
        x=c;
elif b < a:
    if b > c:
        x=b;
    elif c > b:
        x=c;
```

```

else:
    if b > c:
        x=b;
    elif c > b:
        x=c;
    else:
        print('\nOs três valores são iguais')
        y=1

if y ==0:
    print('\nO maior valor é', x)

```

- 4) Crie uma função que identifique se duas palavras inseridas pelo usuário são anagramas.

```

#Questão 4
word1 = input('Digite a primeira palavra: ')
word2 = input('Digite a segunda palavra: ')
tamanho = len(word2)
invertida=word2[tamanho::-1]

if word1 == invertida:
    print('\nAs palavras são anagramas');
else:
    print('\nAs palavras não são anagramas');

```

- 5) Crie uma função que retorne a contagem de consoantes e vogais de uma palavra imputada pelo usuário.

```

#Questão 5
vogal = 0
consoante = 0

```

```

word = input('Digite uma palavra: ')

for i in range(len(word)):
    letra = word[i]
    if (letra=='a' or letra=='e' or letra=='i' or
letra=='o' or letra=='u' or letra=='A' or letra=='E' or
letra=='I' or letra=='O' or letra=='U'):
        vogal = vogal + 1
    else:
        consoante = consoante + 1
if vogal > 1:
    v='vogais'
else:
    v='vogal'

if consoante > 1:
    c='consoantes'
else:
    c='consoante'

print('\nA palavra tem', vogal, v, consoante, c+'.')

```

## 6) O que são dicionários em Python?

Os dicionários em python possuem uma chave que remete a um valor. É um tipo de dado que recebe uma chave e um valor. É uma espécie de estrutura de dados do tipo coleção. Se nas listas, os elementos são acessados por meio de uma posição ou índice, nos dicionários isso ocorre de forma diferente.

7) Crie uma lista com os 100 mil primeiros números naturais na sequência (i.e., de 1 à 100000). Salve esta lista em um objeto com nome *cem\_mil\_lista*. Inicie então uma lista vazia chamada *numeros\_pares*. A partir da lista *cem\_mil\_lista*, salve todos os %%valores pares na lista *numeros\_pares*. Resolva este exercício de duas formas: a primeira utilizando um loop com o **for** e a segunda com **list comprehension**. Compare o tempo de execução das duas.

\* Para comparar o tempo de execução, insira na primeira linha da célula %%time.

O tempo de execução usando o list comprehension é menor.

```
In [137]: %%time
#Questão 7
#For

cem_mil_lista = []

for i in range(1,100001):
    cem_mil_lista.append(i)

numeros_pares = []

for valor in cem_mil_lista:
    if valor%2==0:
        numeros_pares.append(valor)

CPU times: user 31.2 ms, sys: 0 ns, total: 31.2 ms
Wall time: 26.4 ms
```

```
In [138]: %%time
#Questão 7
#List Comprehension

cem_mil_lista = []

for i in range(1,100001):
    cem_mil_lista.append(i)

numeros_pares = [valor for valor in cem_mil_lista if valor%2==0]

CPU times: user 31.2 ms, sys: 0 ns, total: 31.2 ms
Wall time: 19.9 ms
```

8) Para que serve o argumento Kwargs nas funções?

Para criar um dicionário com os parâmetros passados na função em que os kwargs foram usados. Kwargs significa keyword arguments (argumentos de palavras chave). Ele permite passar um dicionário com

inúmeras keys para a função. A convenção é usar dois asteriscos \*\* antes do nome do kwargs.

- 9) Escreva um programa em Python para encontrar os números que são divisíveis por 7 e múltiplos de 5, entre 1500 e 2700 (ambos incluídos). Em outras palavras, combine o que você aprendeu de loop e condicional para criar uma lista de valores que atendam às condições exigidas.

```
#Questão 9
div7_mult5 = []
for i in range(1500,2701):
    if i%7==0 and i%5==0:
        div7_mult5.append(i)
```

- 10) Escreva um programa Python para construir o padrão a seguir, usando um número de loop aninhado. Ir para o editor

Saída esperada:

1

22

333

4444

55555

666666

7777777

88888888

999999999

```
#Questão 10
x = input('Digite um valor: ')

for i in range(int(x)+1):
    print(i*str(i),'\n')
```

- 11) Escreva um programa em Python para verificar se um alfabeto é uma vogal ou uma consoante. Ir para o editor

Saída esperada:

Insira uma letra do alfabeto: m

m é uma consoante.

```
#Questão 11
letter = input('Insira uma letra do alfabeto: ')

if (letter=='a' or letter=='e' or letter=='i' or
letter=='o' or letter=='u' or letter=='A' or letter=='E'
or letter=='I' or letter=='O' or letter=='U'):
    print(letter,'é uma vogal.')
else:
    print(letter, 'é uma consoante.')
```

### (PARTE 3) Bibliotecas e Numpy:

- 12) Para que serve o PIP no Python?

O PIP é uma ferramenta para gerenciamento de pacotes de software escrito em Python.

Similar ao Composer (Gerenciador de pacotes do PHP), o PIP serve para instalar, remover, atualizar ou listar os pacotes instalados em um determinado projeto.

- 13) Crie um array com 10 números no formato float. Converta-os para o formato inteiro.

```
#Questão 13
import numpy as np

floats = np.array([2.5, 3.2, 3.3, 7.5, 8.9, 5.3, 1.2,
9.4, 7.3, 6.1])
print(floats)
inteiros = floats.astype(int)
print(inteiros)
```

- 14) Converta o array abaixo para o formato booleano.

```
a = np.array([[1,0,1],[1,1,1],[0,0,0]])
```

```
#Questão 14
a = np.array([[1,0,1],[1,1,1],[0,0,0]])
booleans = a.astype(bool)
print(booleans)
```

- 15) Empilhe horizontalmente os arrays abaixo.

```
a = np.array([[10,20,15],[0,0,20]])
b = np.array([[15,7,3],[5,5,8]])
```

```
#Questão 15
```

```

a = np.array([[10,20,15],[0,0,20]])
b = np.array([[15,7,3],[5,5,8]])
c = np.concatenate((a,b))

h1 = np.hstack(a)
print( h1)
h2 = np.hstack(b)
print('\n', h2)
h3 = np.hstack(c)
print('\n', h3)

```

- 16) Empilhe verticalmente os arrays do exercício 15. (Use o np.vstack)

```

#Questão 16
a = np.array([[10,20,15],[0,0,20]])
b = np.array([[15,7,3],[5,5,8]])
c = np.concatenate((a,b))

h1 = np.vstack(a)
print(h1)
h2 = np.vstack(b)
print('\n',h2)
h3 = np.vstack(c)
print('\n',h3)

```

- 17) Gere uma lista de números inteiros que vão de 0 a 100, de 2 em 2, e, na sequência, converta essa lista em um array.

```

#Questão 17
lista = np.arange(101, step = 2)
print(lista)
arr = np.asarray(lista)

```



```
print(arr)
```

18) Gere uma lista de valores que vão de 5 a 10 de 0.5 em 0.5.

```
#Questão 18  
lista = np.arange(5, 10, 0.5)  
print(lista)
```

19) Crie um array 3x3 cujos valores seguem uma distribuição normal.

```
#Questão 19  
A = np.random.normal(0, 1, (3, 3))  
print(A)
```

20) Como calcular o determinante de uma matriz no Numpy?

Usando a função `np.linalg.det()`, como no exemplo:

```
#Questão 20  
a = np.array([[2, 3, 5], [6, 7, 8], [10, 11, 12]])  
print(a)  
np.linalg.det(a)
```