Pró-reitora de Pós-Graduação, Pesquisa e Inovação

Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1 / Aula 2: Revisão de Programação Estruturada com Python

Agenda

- Overview de Programação com a Linguagem Python

✓ Tipos numéricos
✓ Strings
✓ Prints
- Listas
- Dicionários
- Booleanos
- Tuplas e Sets
- Operadores Lógicos

- if, else, elif
- for, while
- range()
- Funções
- Expressões lambda
- Mapas e Filtros

Lógica booleana

- Operadores
  - ==, !=, >, <, >=, <=
  - not, and, or
- Instrução de controle
  - if, elif, else
- **Atenção à identação >>>**

Sobre tabela verdade e lógica:
https://www.significados.com.br/tabela-verdade/

```
dicProfs = {'UPE':['Bruno','Byron','Carmelo'],'UFPE':['Aluizio','Tereza'],
            'UNICAP':['Anthony','Madeiro']}
```

```
dicProfs['UPE'][0] == 'Carmelo'
```

```
False
```

```
dicProfs['UPE'][2] == 'Carmelo'
```

```
True
```

```
dicProfs['UPE'][0] != 'Carmelo'
```

```
True
```

```
dicProfs['UPE'][0] == 'Bruno'
```

```
True
```

```
qtd1 = len(dicProfs['UPE'])
qtd2 = len(dicProfs['UFPE'])
qtd3 = len(dicProfs['UNICAP'])
```

```
qtd1 == qtd2
```

```
False
```

```
qtd2 == qtd3
```

```
True
```

```
if qtd1 == qtd2 and qtd1 == qtd3:
    print('Mesmo número de docentes')
elif qtd1 > qtd2 and qtd1 > qtd3:
    print('UPE possui mais docentes')
elif qt2 > qt1 and qtd2 > qtd3:
    print('UFPE possui mais docentes')
else:
    print('UNICAP possui mais docentes')
```

```
UPE possui mais docentes
```

Loops

- for

```
In [128]: dicProfs = {'UPE':['Bruno','Byron','Carmelo'],'UFPE':['Aluizio','Tereza'],
                      'UNICAP':['Anthony','Madeiro']}
```

```
In [129]: for item in dicProfs:
              print(item)
```

```
UPE
UFPE
UNICAP
```

```
In [130]: for item in dicProfs:
              print(dicProfs[item])
```

```
['Bruno', 'Byron', 'Carmelo']
['Aluizio', 'Tereza']
['Anthony', 'Madeiro']
```

Como imprimir os professores agrupados por Universidade ?

Loops

- range

```
for item in dicProfs:
    print('\nProfessores da', item)
    for prof in range(0, len(dicProfs[item])):
        print('   ', prof+1, '-', dicProfs[item][prof])
```

```
Professores da UPE
   1 - Bruno
   2 - Byron
   3 - Carmelo

Professores da UFPE
   1 - Aluizio
   2 - Tereza

Professores da UNICAP
   1 - Anthony
   2 - Madeiro
```

Loops

- while

```
i = 0
while i < 10:
    print(i)
    i = i +1
```

```
0
1
2
3
4
5
6
7
8
9
```

```
i = 0
while i < 10:
  print(i)
  if i == 7:
    break
  i += 1
```

```
0
1
2
3
4
5
6
7
```

```
i = 0
while i < 10:
  i += 1
  if i == 7:
    continue
  print(i)
```

```
1
2
3
4
5
6
8
9
10
```

Exercícios

- Crie um programa que solicita ao usuário que pense em um número qualquer no intervalo de 1 a 100.
- Em seguida o programa deve tentar adivinhar o número pensado pelo usuário usando a abordagem dividir para conquistar:
  - O programa efetua a primeira tentativa no meio do intervalo de 1 a 100, ou seja, 50.
  - O usuário então deve responder se o valor "chutado" pelo computador é o valor correto. Caso a resposta do usuário seja negativa, o computador deve perguntar se o número é menor ou maior que o valor chutado pelo computador.
  - Se o usuário responder que o valor pensado por ele é menor, então você já sabe que o número em questão está no intervalor de 1 a 49 e não mais de 1 a 100.
- Ao final, quando o computador encontrar a resposta, o programa deve informar quantos chutes ele deu.

Pró-reitora de Pós-Graduação, Pesquisa e Inovação

Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1 / Aula 3: Revisão de Programação Estruturada com Python

Agenda

- Overview de Programação com a Linguagem Python

✓ Tipos numéricos
✓ Strings
✓ Prints
✓ if, else, elif
✓ for, while
✓ range()
✓ Operadores Lógicos

- Listas
- Dicionários
- Booleanos
- Tuplas e Sets
- Funções
- Expressões lambda
- Mapas e Filtros

Listas

```
In [79]: lista = [1, 2, 3]

In [80]: type(lista)
Out[80]: list

In [81]: x = lista[0]

In [82]: x
Out[82]: 1

In [83]: len(lista)
Out[83]: 3

In [84]: lista[len(lista)-1]
Out[84]: 3

In [85]: lista[1 : 3]
Out[85]: [2, 3]
```

Slice

Listas Heterogêneas e Strings

```
In [45]: listaHibrida = [nome, idade, lista]

In [46]: listaHibrida
Out[46]: ['Byron', 41, [1, 2, 3]]

In [47]: 1 listaHibrida[2]
Out[47]: [1, 2, 3]

In [48]: listaHibrida[2][2]
Out[48]: 3

In [49]: len(listaHibrida[0])
Out[49]: 5

In [50]: listaHibrida[0][0:3]
Out[50]: 'Byr'

In [51]: listaHibrida[0][2:]
Out[51]: 'ron'
```

Operador *in*

- O operador ***in*** permite de forma simples e intuitiva a localização de informações em listas. Ex.:

```
lista = ['Formação', 'de', 'Recursos', 'Humanos',
'Qualificados', 'em', 'Inteligência', 'Artificial']
termo = 'Inteligência'
r = termo in lista
print(r)
True
```

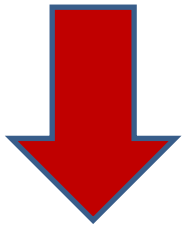
Exercício

- Realize as seguintes operações:
  1. Crie uma lista vazia.
  2. Adicione três elementos do tipo inteiro ao final da lista (*o método para adição de novos elementos em uma lista atua de forma in-place, portanto, basta invocá-lo.*).
  3. Adicione um novo elemento na segunda posição da lista (*método in-place).
  4. Exiba a soma dos quatro elementos (*python já possui uma função built-in para esssa operação*.)
  5. Re-assinale o elemento da terceiro posição com um valor do tipo float.
  6. Repita o passo 4.
  7. Inverta os elementos da lista atual. (*função para inversão é in-place*)
  8. Re-ordene os elementos da lista.
  9. Armazene uma nova lista em uma variável qualquer com 2 números inteiros.
  10. Concatena a nova lista criada com a lista resultante do passo 7.
  11. Exiba o valor máximo e mínimo da resultante do passo anterior.

Dica: *Consulte as funções disponíveis em https://www.w3schools.com/python/python_ref_list.asp*

List comprehensions

```
new_list = []
for i in old_list:
    if filter(i):
        new_list.append(expressions(i))
```



```
new_list = [expression(i) for i in old_list if filter(i)]
```

https://www.pythonforbeginners.com/basics/list-comprehensions-in-python

Exemplo

```
x = range(0,10)
```

```
x2 = []
```

```
for i in x:
    x2.append(i**2)
```

```
x2
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

List comprehensions:

```
x2 = [i**2 for i in x]
```

```
x2
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Pró-reitora de Pós-Graduação, Pesquisa e Inovação

Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 1 / Aula 4: Revisão de Programação Estruturada com Python