

Python para Ciência de Dados



Luiz Alberto

Ciência da Computação

May 7, 2019



Tipos

- String
 - "Data Science", "Programming", "Python"
- Inteiros
 - -11, 7, 500, 700, 0, -80
- Ponto flutuante
 - 1.5, 0.5679, 2.909, -3.4560



Strings (1)

- Um **string** é uma cadeia de caracteres envolvida por aspas simples ou duplas
 - "Data Science", 'Programming', "Python"
- Mudando para letras maiúsculas e minúsculas

```
1     nome = "ada Lovelace"  
2     print(nome.title())  
3     Ada Lovelace  
4     print(nome.upper())  
5     ADA LOVELACE  
6     print(nome.lower())  
7     ada lovelace  
8
```



Strings (2)

■ Combinando ou concatenando strings

```
1     primeiro_nome = "ada"
2     segundo_nome = "lovelace"
3     nome_completo = primeiro_nome + " " + segundo_nome
4     mensagem = "Ola  , " + nome_completo.title() + "!"
5     print(mensagem)
6     Boa tarde, Ada Lovelace!
7
```

■ Acrescentando espaços com tabulações ou quebras de linhas

```
1     print("\tAda Lovelace")
2         Ada Lovelace
3     print("Ada\nLovelace")
4         Ada
5         Lovelace
6
```



Strings (3)

■ Removendo espaços em branco

```
1     nome = " Ada Lovelace "  
2     print("["+nome.rstrip()+"]")  
3     [ Ada Lovelace]  
4     print("["+nome.rstrip()+"]")  
5     [Ada Lovelace ]  
6     print("["+nome.strip()+"]")  
7     [Ada Lovelace]  
8
```



Inteiros

- Python trata números de várias maneiras diferentes

```
1      >>> 2 + 3
2      5
3      >>> 2 * 3
4      6
5      >>> 3 / 2
6      1.5
7      >>> 3 ** 2
8      9
9      >>> 10 ** 6
10     1000000
11     >>> 2 + 3 * 4
12     14
13     >>> (2 + 3) * 4
14     20
15
```



Pontos flutuantes

- Python chama qualquer número com um ponto decimal de *número de ponto flutuante* (float)

```
1      >>> 0.1 + 0.1
2      0.2
3      >>> 0.2 + 0.2
4      0.4
5      >>> 2 * 0.1
6      0.2
7      >>> 2 * 0.2
8      0.4
9
```



Convertendo números para string

```
1 >>> mensagem = "Dia " + str(11) + "de maio."  
2 >>> print(mensagem)  
3 Dia 11 de maio.  
4
```




Comentários

- Um comentário permite escrever notas em seus programas em linguagem natural.

```
1      # o que a pessoas acham sobre o Python.  
2      print('Python is cool!')
```

3



Zen do Python

1. Bonito é melhor do que feio
2. Simples é melhor que complexo
3. Complexo é melhor que complicado
4. Legibilidade conta
5. Deve haver -e, de preferência, apenas uma maneira óbvia de fazer algo
6. Agora é melhor do que nunca

```
1 >>> import this
2 The Zen of Python, by Tim Peters
3
4 Beautiful is better than ugly
5
```



Listas (1)

- Coleção de itens em uma ordem particular
- Itens podem ser letras, dígitos e etc.
 - não precisam estar relacionados de nenhum modo em particular
- Em Python, colchetes([]) indicam uma lista, e elementos individuais da lista são separados por vírgula
- A posição dos índices de uma lista começa em 0 e não em 1



Acessando elementos da lista (1)

- Escreva o nome da lista seguido do índice do item entre colchetes

```
1      >>> mestres = ['yoda', 'qui-gon', 'kenobi'  
2                  , 'luke']  
3      >>> print(mestres[0])  
4      yoda  
5      >>> print(mestres[0].title())  
6      Yoda  
7
```

- Python tem uma sintaxe especial para acessar o último elemento de uma lista



Acessando elementos da lista (2)

```
1 >>> print(mestres[-1].title())  
2 Luke  
3
```

Acrescentando elementos no final da lista (1)



```
1      >>> mestres = ['yoda', 'qui-gon', 'kenobi',  
2                    , 'luke']  
3      >>> mestres.append('windu')  
4      >>> print(mestres)  
5      ['yoda', 'qui-gon', 'kenobi', 'luke', 'windu']  
6
```

- O método **append()** acrescenta o elemento 'windu' no final da lista

Acrescentando elementos no final da lista (2)



- O método **append()** facilita a criação de listas dinamicamente

```
1      casas = []
2      casas.append('starks')
3      casas.append('greyjoy')
4      casas.append('tyrell')
5      casas.append('lannister')
6      print(casas)
7      ['starks', 'greyjoy', 'tyrell', 'lannister']
8
```



Inserindo elementos em uma lista (1)

- O método **insert** permite adicionar um elemento em qualquer posição da lista

```
1     mestres = ['yoda', 'qui-gon', 'kenobi', 'luke']
2     mestres.insert(0, 'vader')
3     print(mestres)
4     ['vader', 'yoda', 'qui-gon', 'kenobi', 'luke']
5
```