



30 September 2019

Introdução ao Python

Funções, Listas, Dicionários e Tuplas
SIDIA - Setembro/2019

Jailson P. Januário

jpj.ads@uea.edu.br

Github: @jailsonpj



Funções

- Para criar uma função:
 - **def** nome_da_funcao(**parametros**):
- Para retornar algum valor, usamos o **return**
- Uma função não precisa retornar algo
- Não existe sobrecarga
- Valor padrão de um parâmetro



Funções - Exemplos

```
1  ...
2  Exemplos - Curso SIDIA Setembro/2019
3
4  ...
5
6  def funcao(a,b=5):
7      print('a = ', a)
8      print('b = ', b)
9
10 funcao(2,6)
11
```

```
12 def spam():
13     global eggs
14     eggs = 'spam'
15
16 eggs = 'global'
17 spam()
18 print(eggs)
19
```

```
21
22 def divisao(divid_by):
23     try:
24         return 42/divid_by
25     except ZeroDivisionError:
26         print('Error: Invalid argument.')
27
28 print(divisao(0))
29
30
31
```



Exercícios

- Faça uma função para calcular o fatorial de um número (função não recursiva)
- Faça uma função para verificar se um número é primo ou não



Listas

Listas

Uma *lista* é um valor que contém diversos valores em uma sequência ordenada.

- Não existe vetores em Python, nós usamos **listas**
- Uma lista guarda uma sequência de itens
- São identificadas por colchetes '[']':
 - `lista = [1,2,3,4,5]`
- Os itens podem ser acessados como um vetor:
 - `lista[0]`



Listas - Inserindo elementos

- **append(obj)**

- Insere o objeto no fim da lista

```
>>> lista = ['cat', 'dog', 'bat']  
>>> lista.append(1)  
>>> lista  
['cat', 'dog', 'bat', 1]
```

- **insert(posição,obj)**

- Insere o objeto na posição especificada

```
>>> lista = [1,2,3]  
>>> lista.insert(1,'objeto')  
>>> lista  
[1,'objeto',2,3]
```



Listas - Removendo elementos

- **remove(obj)**
 - Remove o objeto da lista

- **pop(posição)**
 - Remove e retorna o objeto da posição especificada. O `pop()` sem parâmetros retira o último elemento

```
>>> lista = ['a','b',3]
>>> lista.remove('b')
>>> lista
['a',3]
```

```
>>> lista = ['a',2,'b',4,'c']
>>> lista.pop(0)
'a'
>>> lista
[2,'b',4,'c']
>>> lista.pop()
'c'
>>> lista
[2,'b',4]
```



Listas - Outros métodos

- **count(obj)**
 - Retorna a quantidade de elemento 'obj' da lista
- **index(obj)**
 - Retorna a posição do 'obj' na lista
- **reverse(obj)**
 - Inverte a lista colocando os elementos ao contrário
- **sort()**
 - Ordena a lista



Tuplas

Tipo de dado tupla

O tipo de dado tupla é quase idêntico ao tipo de dado lista, exceto em relação a dois aspectos:

- São definidas por parênteses '()' ou objetos divididos por ','
- Assim como as strings, as tuplas são imutáveis.

```
>>> tupla = (1,2,3)
>>> tupla
(1,2,3)
```

```
>>> tupla = 1,2,3
>>> tupla
(1,2,3)
```



Dicionários

Dicionário

É uma coleção de diversos valores. Porém, de modo diferente das listas, os índices dos dicionários podem utilizar vários tipos de dados diferentes, e não apenas inteiros. Os índices nos dicionários são chamados de *chaves* (keys), e uma chave juntamente com seu valor associado é chamado de *par chave-valor* (key-value pair).



Dicionários

- O programador pode definir a chave do item
- Representados por chaves '{ }'

- `my_cat = {'size': 'fat', 'color': 'gray', 'disposition': 'loud'}`



Dicionários - métodos

- Inserir item
 - `dicionario[chave] = obj`
- Remover item
 - `dicionario.pop(chave)`
 - Se não encontrar a chave retorna erro
 - `dicionario.pop(chave, retorno)`
 - Se não encontrar a chave retorna o valor de retorno
- Remover todos itens
 - `dicionario.clear()`



Dicionários - métodos

- Acessar item
 - `dicionario[chave]`
 - `dicionario.get(chave)`
 - Se a chave não existir retorna `None`
 - `dicionario.get(chave, retorno)`
 - Se a chave não existir retorna o valor de retorno
- Verificar se existe item
 - `dicionario.has_key(chave)`



Exercícios

- Função para calcular série Fibonacci até n
- Imprimir quantos números são maiores que 10 na lista:
 - [1,4,15,26,3,6,10,13,5,3]
- Leia o nome e a nota de 5 alunos e guarde em um dicionário
- Função que receba uma string e imprima em escada, ex:
 - T
 - TE
 - TES
 - TEST
 - TESTE



Material Adicional

- Github: <https://github.com/jailsonpj/miniCurso/>



Contatos

- Jailson P. Januário
 - Email: jpj.ads@uea.edu.br ou jailsoncolares@hotmail.com
 - Twitter: @jailson_jpj
 - Github: @jailsonpj



Bibliografia

- LIVRO: Novatec - Automatize Tarefas Maçantes com Python
- LIVRO: Person - Object-Oriented Programming in Python
- <https://www.python.org/>
- <http://www.python.org.br/>
- Mais exercícios
 - <http://www.python.org.br/wiki/ListaDeExercicios>