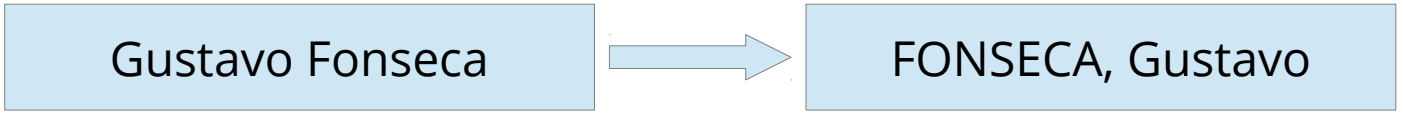
Introdução à programação para não programadores
Aula 2

Pensar em processos

Como normalizar o nome dos autores?



1. Separar as partes
2. Alterar a caixa da parte que contém o sobrenome
3. Juntar as partes, na nova sequência, utilizando a vírgula como separador

Pensar em processos

```
 1 autor = 'Gustavo Fonseca'
 2
 3 # separar as partes
 4 partes = autor.split()
 5
 6 # alterar a caixa do ultimo elemento da lista
 7 partes[-1] = partes[-1].upper()
 8
 9 # juntar as partes utilizando a virgula como separador
10 autor_normalizado = ', '.join(partes[::-1])
```

Retorna uma lista das palavras, separadas por um dado delimitador. O delimitador padrão é o espaço em branco

Atribui ao último elemento da lista

Listas são mutáveis!

É o inverso do método **str.split**. O método **str.join** monta uma string com elementos de uma lista, utilizando um determinado delimitador

Abstração

def é uma instrução que serve para definir funções. Ela é seguida pelo **nome da função** e os **parâmetros** esperados.

```
 1 def normaliza_autor(autor):
 2     # separar as partes
 3     partes = autor.split()
 4
 5     # alterar a caixa do ultimo elemento da lista
 6     partes[-1] = partes[-1].upper()
 7
 8     # juntar as partes utilizando a virgula como separador
 9     autor_normalizado = ', '.join(partes[::-1])
10
11     return autor_normalizado
12
13 autor = 'Gustavo Fonseca'
14 print normaliza_autor(autor)
```

Funções sempre retornam valores por meio do comando **return**

Sempre que desejarmos utilizar a funcionalidade, basta invocá-la passando os argumentos

SciELO

Testes automáticos

Docstrings tem a função de documentar o código e podem abrigar testes automáticos chamados de **Doctests**

```
 1 def normaliza_autor(autor):
 2     """
 3     Normaliza o nome do autor.
 4
 5     >>> normaliza_autor('Gustavo Fonseca')
 6     'FONSECA, Gustavo'
 7     """
 8     # separar as partes
 9     partes = autor.split()
10
11     # alterar a caixa do ultimo elemento da lista
12     partes[-1] = partes[-1].upper()
13
14     # juntar as partes utilizando a virgula como separador
15     autor_normalizado = ', '.join(partes[::-1])
16
17     return autor_normalizado
```

Instruções de **doctests** são similares às do console interativo

Executamos a biblioteca **doctest** junto com o nosso script

```
PAT113-SCIELO:src gustavofonseca (master) python -m doctest autores_doctest.py
PAT113-SCIELO:src gustavofonseca (master)
```

Somente os erros são exibidos. Então deu certo! =]

Testes automáticos

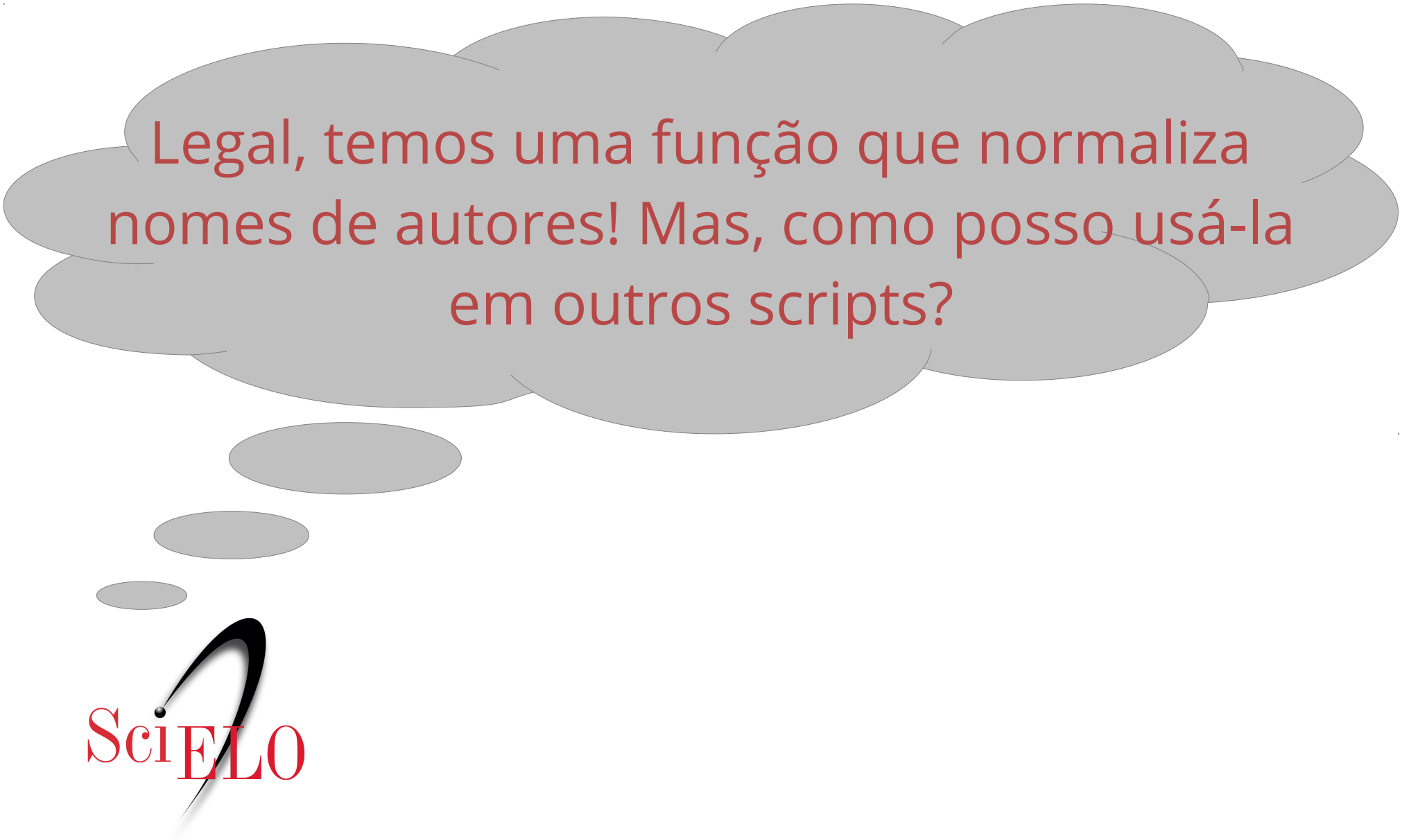
```
 1 def normaliza_autor(autor):
 2     """
 3     Normaliza o nome do autor.
 4
 5     >>> normaliza_autor('Gustavo Fonseca')
 6     'FONSECA, Gustavo'
 7     """
 8     # separar as partes
 9     partes = autor.split()
10
11     # alterar a caixa do ultimo elemento da lista
12     partes[-1] = partes[-1].upper()
13
14     # juntar as partes utilizando a virgula como separador
15     autor_normalizado = ', '.join(partes)
16
17     return autor_normalizado
```

Introduzimos um **bug** aqui

```
PAT113-SCIELO:src gustavofonseca (master) python -m doctest autores_doctest_err.py
**********************************************************************
File "autores_doctest_err.py", line 5, in autores_doctest_err.normaliza_autor
Failed example:
    normaliza_autor('Gustavo Fonseca')
Expected:
    'FONSECA, Gustavo'
Got:
    'Gustavo, FONSECA'
**********************************************************************
1 items had failures:
   1 of   1 in autores_doctest_err.normaliza_autor
***Test Failed*** 1 failures.
PAT113-SCIELO:src gustavofonseca (master)
```

SciELO

Reuso de código



Importação de módulos

Adicionamos o nome **normaliza_autor** na tabela local de símbolos

```
>>> from autores_doctest import normaliza_autor
>>> normaliza_autor('Jimmi Hendrix')
'HENDRIX, Jimmi'
>>>
```

De maneira semelhante, pode-se utilizar a forma **import autores_doctest** e utilizar na forma **autores_doctest.normaliza_autor('Pato Donald')** ou ainda **from autores_doctest import *** e utilizar da mesma forma que a figura

Quando importamos um **nome**, o interpretador percorre os locais listados em **sys.path** em busca do pacote/módulo a ser importado e o adiciona na tabela local de símbolos

```
>>> import sys
>>> sys.path
['', '/Library/Python/2.7/site-packages/pip-1.0.2-py2.7.egg', '/Library/Python/2.7/site-packages/NoseGrowl-0.4-py2.7.egg', '/Library/Python/2.7/site-packages/py_Growl-0.0.7-py2.7-macosx-10.7-intel.egg', '/Library/Python/2.7/site-packages/readline-6.2.2-py2.7-macosx-10.7-intel.egg', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python27.zip', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-darwin', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac/lib-scriptpackages', '/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-tk', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-old', '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-dynload', '/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC', '/Library/Python/2.7/site-packages', '/opt/gtk/lib/python2.7/site-packages', '/opt/gtk/lib/python2.7/site-packages/gtk-2.0']
>>>
```

SciELO

Resumo

- Geral
  - Os processos que nos dizem **como** resolver o problema são chamados de **Algoritmos**
  - Uma sequência lógica de processos pode ser nomeada
- Tipos de dados
  - **str.split** – Separa uma string em uma lista de palavras
  - **str.join** – Junta uma lista de palavras em uma string
- Estruturas de dados
  - Listas são **mutáveis**

Resumo

- Comandos compostos
  - **def** – Define uma nova função
  - **return** – Retorna valores de uma função
- Testes automáticos
  - Validam os códigos e garantem que não hajam regressões
  - Podem ser utilizados como documentação
- Importação de módulos
  - Promove o **reuso** dos códigos
  - Possibilita a **organização** dos códigos