**2022-05-30-SS**

**Fonte: Programadores de Python: João Rodrigues, António Neves, Artur Pereira Fundamentos de Programação, 2019**

[1 Exercícios 1](#_Toc104798616)

[1.1 Média 1](#_Toc104798617)

[1.2 Turtle 1](#_Toc104798618)

[1.3 Pautas 1](#_Toc104798619)

[1.4 Média de uma lista de valores 2](#_Toc104798620)

[1.5 Geração de jogos de futebol 2](#_Toc104798621)

[2 Python – Ficheiros 2](#_Toc104798622)

[2.1 Exemplo 3](#_Toc104798623)

[2.2 Funções: split 3](#_Toc104798624)

[2.3 Funções: strip, rstrip, lstrip 3](#_Toc104798625)

# Exercícios

Resolva os exercícios dos slides: 06-Vetores.pdf, 10-Strings.pdf e 08-Listas.pdf

## Média

Crie um programa que calcule a soma de uma lista de valores gravados num ficheiro. Considere que o ficheiro contém apenas um valor por linha, como no exemplo abaixo. O nome do ficheiro deve ser pedido ao utilizador.

**nums.txt**

12.39

1.93

7.85

...

## Turtle

O ficheiro mystery.txt contém linhas com as palavras UP, DOWN ou com um par de números. UP e DOWN são instruções para a tartaruga levantar ou pousar o pincel. Os pares são coordenadas x, y de pontos. Complete o programa turtledraw.py para ler as instruções do ficheiro e usar uma tartaruga para traçar o desenho. *Sugestão: use o* método .split() para dividir as linhas que têm coordenadas. (Adaptado do exercício 11.5 do livro “How to think like a computer scientist”.)

## Pautas

O ficheiro school.csv contém uma tabela com notas de uma disciplina. Cada linha tem o registo de um aluno e cada coluna tem um campo de informação. As colunas são separadas por caracteres TAB. A primeira linha contém um cabeçalho com os títulos dos campos. Complete o programa school.py para ler e processar esse ficheiro.

1. Complete a função loadFile(fname) para que, dado o nome de um ficheiro com este formato, leia o seu conteúdo e devolva uma lista com um tuplo por cada aluno.
2. Cada tuplo deve ter os campos (numero, nome, nota1, nota2, nota3). Use o método .split('\t') para dividir cada linha e converta as notas e os números para os tipos adequados.
3. Crie uma função notaFinal(reg) que, dado um tuplo com o registo de um aluno, devolva a nota final calculada pela média das três notas no registo.
4. Crie uma função printPauta(lst) que, dada uma lista com registos de alunos, mostra uma tabela com os números, nomes e notas finais, formatados e alinhados como no exemplo abaixo. O nome deve aparecer centrado, enquanto o número e a nota devem aparecer ajustados à direita. Use o método .format.
5. Usando estas funções, complete a função principal para ler o ficheiro, ordenar a lista com o método .sort() e mostrar a pauta.

## Média de uma lista de valores

Altere o programa anterior para gravar a pauta formatada num ficheiro de texto. Pode usar o método write ou a função print com o argumento file=. *Sugestão: se* generalizar a função printPauta para também receber um argumento file extra, pode reutilizá-la para gravar no ficheiro

## Geração de jogos de futebol

Escreva uma função que, dada uma lista de equipas de futebol, gere uma lista de todos os jogos que se podem fazer entre elas grave o resultado num ficheiro texto. Por exemplo:

**allMatches(["SCP", "SLB", "FCP"]) -> [("SCP", "SLB"), ("SCP", "FCP"), ("SLB", "SCP"), ...]**

Com 3 equipas deve obter 6 jogos, com 4 equipas deve obter 12 jogos. Confirme e teste com ainda mais equipas.

# Python – Ficheiros

<https://www.w3schools.com/python/python_file_open.asp>

“*Os módulos descritos neste capítulo possuem suporte ao armazenamento de dados do Python em um formato persistente no disco. Os módulos*[*pickle*](https://docs.python.org/pt-br/dev/library/pickle.html#module-pickle)*e*[*marshal*](https://docs.python.org/pt-br/dev/library/marshal.html#module-marshal)*podem transformar muitos tipos de dados do Python em um fluxo de bytes e então recriar os objetos a partir dos bytes. Os vários módulos relacionados ao DBM possuem suporte a uma família de formatos de arquivo baseados em hash que armazenam um mapeamento de strings para outras strings.*”

Fonte: <https://docs.python.org/pt-br/dev/library/persistence.html>

<https://docs.python.org/pt-br/dev/library/fileinput.html>

<https://docs.python.org/pt-br/dev/library/csv.html>

## Exemplo

f = open("demofile.txt", "r")  
print(f.readline())

f = open("demofile.txt", "r")  
print(f.readline())  
print(f.readline())

Loop through the file line by line:

f = open("demofile.txt", "r")  
for x in f:  
  print(x)

## Funções: split

<https://www.w3schools.com/python/ref_string_split.asp>

Split a string into a list where each word is a list item:

txt = "welcome to the jungle"  
x = txt.split()  
print(x)

Use a hash character as a separator:

txt = "apple#banana#cherry#orange"  
x = txt.split("#")  
print(x)

## Funções: strip, rstrip, lstrip

<https://www.w3schools.com/python/ref_string_rstrip.asp>

Remove any white spaces at the end of the string:

txt = "     banana     "  
x = txt.rstrip()  
print("of all fruits", x, "is my favorite")

Remove the trailing characters if they are commas, s, q, or w:

txt = "banana,,,,,ssqqqww....."  
x = txt.rstrip(",.qsw")  
print(x)

Remove spaces at the beginning and at the end of the string:

txt = "     banana     "  
  
x = txt.strip()  
  
print("of all fruits", x, "is my favorite")