

# class4

December 11, 2021

## 1 Introdução a Python

### 1.1 Aula 4

## 2 Sumário

- Funções de ordem superior
- Módulo os
- Input/Output
- Exercícios

## 3 Funções de ordem superior

`map()`: Utiliza-se para aplicar uma função a um ou mais iteráveis, sequencialmente

```
[1]: def difference(x, y):  
      return x - y  
  
list1 = [8,5,3]  
list2 = [3,4,1]  
  
list(map(difference, list1, list2))
```

```
[1]: [5, 1, 2]
```

`filter()`: Utiliza-se para filtrar os elementos de um iterável, com recurso a uma função

```
[2]: list3 = [1,2,3,4,5]  
  
list(filter(lambda x: x%2==0, list3))
```

```
[2]: [2, 4]
```

`reduce()`: Utiliza-se para aplicar uma função cumulativa a um iterável

```
[2]: from functools import reduce

def sum(x,y):
    return x+y

list4 = [6,3,45,74]

reduce(sum, list4, 2) #o terceiro argumento é opcional
```

[2]: 130

## 4 Módulo os

```
[4]: import os
```

os.getcwd(): Utiliza-se para saber qual é a diretoria atual

```
[5]: os.getcwd()
```

[5]: '/home/beatriz/python-introduction/notebooks'

os.listdir(): Utiliza-se para listar uma diretoria

```
[6]: os.listdir()
```

```
[6]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'class2.ipynb',
      'solutions_class4.py',
      'temp.txt',
      'solutions_class3.py',
      'solutions_class2.py',
      'class4.ipynb']
```

os.chdir(): Utiliza-se para mudar de diretoria

```
[7]: os.chdir("/home/beatriz/python-introduction")

os.getcwd()
```

[7]: '/home/beatriz/python-introduction'

os.path.join(): Utiliza-se para unir vários caminhos

```
[8]: new_dir = os.path.join("/home/beatriz/python-introduction", "notebooks")

os.chdir(new_dir)

os.getcwd()
```

```
[8]: '/home/beatriz/python-introduction/notebooks'
```

`os.path.exists()`: Utiliza-se para verificar se um ficheiro existe

```
[9]: os.path.exists("class4.ipynb")
```

```
[9]: True
```

`os.mkdir()`: Utiliza-se para criar uma nova diretoria

```
[10]: os.mkdir("my_new_dir")

os.listdir()
```

```
[10]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'my_new_dir',
      'class2.ipynb',
      'solutions_class4.py',
      'temp.txt',
      'solutions_class3.py',
      'solutions_class2.py',
      'class4.ipynb']
```

`os.rename()`: Utiliza-se para trocar o nome de um ficheiro ou de uma diretoria

```
[11]: os.rename("my_new_dir", "my_old_dir")

os.listdir()
```

```
[11]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'class2.ipynb',
      'solutions_class4.py',
      'temp.txt',
      'solutions_class3.py',
      'my_old_dir',
      'solutions_class2.py',
```

```
'class4.ipynb']
```

`os.rmdir()`: Utiliza-se para remover uma diretoria

```
[12]: os.rmdir("my_old_dir")

os.listdir()
```

```
[12]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'class2.ipynb',
      'solutions_class4.py',
      'temp.txt',
      'solutions_class3.py',
      'solutions_class2.py',
      'class4.ipynb']
```

`os.remove()`: Utiliza-se para remover um ficheiro

```
[13]: os.remove("temp.txt")

os.listdir()
```

```
[13]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'class2.ipynb',
      'solutions_class4.py',
      'solutions_class3.py',
      'solutions_class2.py',
      'class4.ipynb']
```

## 5 Input/Output

### 5.1 Utilizador

`input()`: Permite interagir com o utilizador (a partir da versão 3.6)

```
[4]: """age = input("Enter your age: ")
      print(f"User is {age} years old")"""
```

```
Enter your age: 21
User is 21 years old
```

## 5.2 Ficheiro

`open()`: Utiliza-se para abrir um ficheiro e recebe dois argumentos (caminho do ficheiro e modo)

Diferentes modos de abrir um ficheiro:

- `r`: Abre um ficheiro para leitura e levanta um erro caso o ficheiro não exista (modo por omissão)
- `a`: Abre um ficheiro para acrescentamento e cria o mesmo caso este não exista
- `w`: Abre um ficheiro para escrita e cria o mesmo caso este não exista
- `x`: Cria o ficheiro especificado e levanta um erro caso este já exista
- `r+`: Abre um ficheiro para leitura e escrita e levanta um erro caso o ficheiro não exista
- `a+`: Abre um ficheiro para acrescentamento e leitura e cria o mesmo caso este não exista
- `w+`: Abre um ficheiro para escrita e leitura e cria o mesmo caso este não exista

### 5.2.1 Leitura

```
[15]: f = open("example.txt", "r")
      print(f.read())
      f.close()
```

```
Hello world!
Welcome to Python introduction course.
This is class number 4.
Stay focused!
```

```
[16]: with open("example.txt", "r") as f: #com o with, não precisamos de fechar o
      ↪ ficheiro
      result = f.read()
      print(result)
```

```
Hello world!
Welcome to Python introduction course.
This is class number 4.
Stay focused!
```

```
[17]: with open("example.txt", "r") as f:
      result = f.read(5)
      print(result)
```

```
Hello
```

```
[18]: with open("example.txt", "r") as f:
      result = f.readline()
      print(result)
```

Hello world!

```
[19]: with open("example.txt", "r") as f:
      result = f.readlines()
      for line in result:
          print(line)
```

Hello world!

Welcome to Python introduction course.

This is class number 4.

Stay focused!

### 5.2.2 Escrita/Criação

```
[20]: with open("example.txt", "a") as f:
      f.write("\nMore information.")

      with open("example.txt", "r") as f:
          result = f.read()
          print(result)
```

Hello world!

Welcome to Python introduction course.

This is class number 4.

Stay focused!

More information.

```
[21]: with open("example.txt", "w") as f:
      f.write("All gone.")

      with open("example.txt", "r") as f:
          result = f.read()
          print(result)
```

All gone.

```
[22]: with open("new_file.txt", "x") as f:
      f.write("I've just created my first file!")

      os.listdir()
```

```
[22]: ['class1.ipynb',
      'class3.ipynb',
      '.ipynb_checkpoints',
      'example.txt',
      'class2.ipynb',
      'solutions_class4.py',
      'new_file.txt',
      'solutions_class3.py',
      'solutions_class2.py',
      'class4.ipynb']
```

## 6 Exercícios

- 1) Escreva uma função que seja capaz de simular uma calculadora (para simplificar, apenas suporte as operações de soma, subtração, divisão e multiplicação entre dois números).
- 2) Escreva uma função que seja capaz de escrever todo o conteúdo de um ficheiro num novo ficheiro, passando à frente a linha número 5.
- 3) Escreva uma função que recebe o caminho de um ficheiro como input e retorna o número de palavras presentes nesse ficheiro.
- 4) Escreva uma função que recebe uma lista de palavras e um inteiro e retorna a lista de palavras que são maiores que esse inteiro, com recurso a funções de ordem superior.
- 5) Escreva uma função que recebe uma lista de números e retorna a lista dos valores absolutos desses números, com recurso a funções de ordem superior.
- 6) Escreva uma função que recebe uma lista de números e retorna o menor, com recurso a funções de ordem superior.
- 7) Escreva uma função que retorna a série harmónica de 1 até  $n-1$ , com recurso a funções de ordem superior.
- 8) Escreva uma função que retorna a soma de três listas, com recurso a funções de ordem superior.
- 9) Escreva uma função que aceita uma palavra e o caminho de um ficheiro como input e retorna a frequência dessa palavra nesse ficheiro.
- 10) Escreva uma função que recebe uma lista de números e retorna outra lista com apenas os números positivos, com recurso a funções de ordem superior.