

## Folha 6 - Módulos, ficheiros

Python v 3.\*

### Exercícios

1) Considere que tem um módulo myStuff.py com o seguinte código:

```
#Definido em myStuff.py
def lime():
    print("I am a lime!")
banana = "I am a banana"
def grapes(x):
    return "I am a grape cluster with " + str(x) + " grapes!"
```

a) Identifique os erros do seguinte excerto de código:

```
import myStuff
grapes()
banana = "I was a banana"
print(banana, myStuff.banana())
myStuff.lime()
```

b) Qual seria o resultado no ecrã de executar o programa completamente corrigido?

c) Qual a diferença entre as seguintes instruções:

```
import myStuff
from myStuff import *
from myStuff import grapes
```

2) Considere o seguinte excerto de código:

```
inFile = open("inFile.txt", 'r')
inData = in_file.read()
outFile = open("outFile.txt", 'w')
outFile.write(inData)
outFile.close()
inFile.close()
```

a) Descreva o seu funcionamento.

b) Seria possível reduzir o número de instruções mantendo o mesmo comportamento?

**3)** Suponha que executava os seguintes programas. Qual seria o resultado?

```
1. outFile = open("outFile.txt", 'w')
   for i in range(100):
       outFile.write(str(i))

   outFile.close()

2. inFile = open("inFile.txt", 'r')
   inData = inFile.read()
   inFile.close()
   inData = inFile.read()
   inFile.close()

3. inFile = open("inFile.txt", 'r')
   print inFile.readline()
   inFile = open("inFile.txt" , 'r')
   print inFile.readline()
   inFile.close()
```

**4)** Identifique o resultado de cada um dos seguintes excertos de código assumindo que o conteúdo do ficheiro `inFile.txt` é:

```
kiwi
pera
maçã
ananás
```

```
1. inFile = open("inFile.txt", 'r')
   inData = inFile.read()
   print(inData)
   inFile.close()

2. inFile = open("inFile.txt", 'r')
   inData = inFile.readline()
   print(inData)
   inFile.close()

3. inFile = open("inFile.txt", 'r')
   inData = inFile.readlines()
   print(inData)
   inFile.close()

4. inFile = open("inFile.txt", 'r')
   inData = list(inFile)
   print(inData)
   inFile.close()
```

```
5. inFile = open("inFile.txt", 'r')
   for i in range(5):
       print inFile.readline()
   inFile.close()

6. inFile = open("inFile.txt", 'r')
   for i in inFile:
       print(i)
   inFile.close()
```

**5)** Admitindo que o conteúdo do ficheiro `inFile.txt` é:

```
10
30
40
21
```

Considere o seguinte programa que tem como objectivo calcular a soma dos inteiros contidos no ficheiro `inFile.txt`:

```
inFile = open("inFile.txt", 'r')
sum = 2
for i in range(8)
    sum += inFile.readline()
outFile = open("outFile.txt", 'r')
outFile.write(i)
```

- a) Identifique os erros no programa.
- b) Apresente uma versão corrigida.

- 6)** Escreva um programa que leia um ficheiro de texto linha a linha e escreva o seu conteúdo no ecrã.
- 7)** Modifique o programa da alínea anterior de forma a aparecer também o número de cada linha.
- 8)** Construa um programa que peça ao utilizador várias linhas e as escreva num ficheiro. O programa termina quando o utilizador introduzir uma linha vazia.

## Problemas

### 1) Considere o seguinte programa:

```
number = 7
guess = -1
print("Adivinha o número!")
while guess != number:
    guess = int(input("Será... "))
    if guess == number:
        print("Viva! Adivinhaste!")
    elif guess < number:
        print("É maior...")
    elif guess > number:
        print("Não é tão grande.")
```

Modifique-o de modo a que o número a adivinhar seja gerado aleatoriamente. Recorra a um módulo das bibliotecas Python.

### 2) Construa o seguinte:

- Um módulo que disponibiliza operações com círculos, nomeadamente, cálculo do perímetro e da área dado o comprimento do raio.
- Um módulo que disponibiliza operações com quadrados, nomeadamente, cálculo do perímetro e da área dado o comprimento do lado.
- Um programa que recorrendo aos módulos desenvolvidos nas alíneas anteriores, peça ao utilizador um valor e escreva no ecrã o perímetro e a área do círculo e do quadrado em que o valor dado corresponde à medida do raio e do lado.
- Um programa que recorrendo aos módulos desenvolvidos nas alíneas anteriores, peça ao utilizador um valor e escreva no ecrã o perímetro e a área do maior quadrado contido no círculo com o raio igual ao valor dado.

- 3)** Construa um módulo que contenha operações definidas sobre polígonos regulares genéricos. As operações devem calcular o perímetro e a área de um polígono regular dados o número de lados e o comprimento de um lado. Lembre-se que o perímetro é a soma dos comprimentos de todos os lados e a área é o produto do apótema pelo perímetro sobre 2.

$$raio = \frac{1}{2} * lado * \sin^{-1}\left(\frac{\pi}{n}\right)$$

$$apótema = \frac{1}{2} * lado * \cot\left(\frac{\pi}{n}\right)$$

- a) Use o módulo desenvolvido para calcular o polígono regular com menor número de lados que pode ser inscrito num círculo de raio unitário de forma a que a diferença dos seus perímetros seja inferior a 0.001.

O comprimento de um lado de um polígono regular com  $n$  lados inscrito num círculo de raio  $r$  é dado por:

$$lado = 2 * r * \text{seno}\left(\frac{\pi}{n}\right)$$

- b) Resolva a alínea anterior usando a área em vez do perímetro.

- 4)** Escreva o enunciado de um problema para cuja resolução seja necessário escrever um programa que invoque pelo menos uma vez cada uma das funções do módulo definido no problema anterior. Construa o programa que resolve esse problema.

- 5)** Construa um módulo para manipulação de ficheiros de texto que contenha as seguintes funções:

- a) Uma função que conte o número de espaços em branco de um ficheiro dado.
- b) Uma função que conte o número de palavras de um ficheiro dado.
- c) Defina uma função que escreva num ficheiro toda a informação relevante de um outro ficheiro, nomeadamente, nome, número de linhas, número de palavras e número de caracteres.

**6)** Escreva o enunciado de um problema para cuja resolução seja necessário escrever um programa que invoque pelo menos uma vez cada uma das funções do módulo definido no problema anterior. Construa o programa que resolve esse problema.

**7)** Construa um programa que permita duplicar um ficheiro dado. O nome do ficheiro a copiar deve ser fornecido como parâmetro. O código seguinte ilustra como obter os argumentos passados para o programa.

```
from sys import argv
script_name, from_file, to_file = argv
print("Copying from {:s} to {:s}".format(from_file, to_file))
```

**8)** Construa um módulo que ofereça a função `cesar` que permite codificar letras. A função `cesar` recebe uma letra **c** e um deslocamento **d** e devolve a letra do alfabeto que se encontra à distância **d** da letra original. Por exemplo `cesar('b',5)` devolve a letra 'g'. Considere que o alfabeto é circular.

**9)** Recorrendo ao módulo construído na alínea anterior, desenvolva um programa que permita ler um texto, codificá-lo e escrevê-lo novamente num ficheiro. Consegue escrever também um descodificador?

**10)** Desenvolva uma função que, dado um ficheiro de texto contendo um programa em Python, faça a sua apresentação no ecrã, removendo todos os comentários.

- a) Considere que os comentários ocorrem apenas no início de uma linha, e são assinalados por um carácter #.
- b) Considere que os comentários são assinalados pelo carácter # e podem ocorrer em qualquer lugar de uma linha.
- c) Remova igualmente as *docstrings*.

- 11)** A capacidade de manipular ficheiros permite construir ferramentas mais complexas. Por exemplo, um pequeno programa Python para obter uma página da web pode ter a seguinte forma:

```
import urllib
the_page = urllib.urlopen("http://www.fc.ul.pt")
the_text = the_page.read()
print(the_text)
```

Construa uma função que dado um endereço web, escreva num ficheiro todos os links existentes nessa página (os links são identificados pelo tag

"<a href="<http://...>" )