

Programação em Python

Tópicos

- Estruturas de dados simples: sequências - listas, strings e tuplos

Exercícios

1. Tente adivinhar o resultado de cada uma das instruções abaixo. Note que algumas não têm resultado e outras dão erros. Use o Python em modo interativo (linha de comandos) para confirmar.

```
lst = [25, 33, 98, 57]
len(lst)
lst[2]
lst[4]
lst[-4]
lst[1:3]
lst[:-1]
lst[2:2]
lst[2:2] = [99] ; lst
lst.append(33) ; lst
```

2. Siga os seguintes passos, testando cada um:

- a) Crie uma função `inputIntList()` que leia uma sequência de números inteiros introduzidos pelo utilizador e os devolva numa nova lista. O utilizador deve introduzir um número por linha e indicar o fim da lista com uma linha vazia.

```
#Exercicio 2a
def inputIntList():
    print("Introduza uma sequência de números inteiros: ")
    lst=[]
    while True:
        num=input()
        if not num: break
        lst.append(int(num))
    return lst
print(inputIntList())
```

- b) Crie uma função `countLower(lst, v)` que conte (e devolva) quantos elementos da lista `lst` são inferiores ao valor `v`.

```
#Exercicio 2b
def countLower(lst, v):
    count=0
    for num in lst:
        if num<v: count+=1
    return count
print("Elementos inferiores a 5 em [1,2,3,4,5,6]: ", countLower([1,2,3,4,5,6], 5))
```

- c) Crie uma função `minmax(lst)` que devolva o mínimo e o máximo de uma lista de valores. Consegue resolver sem usar as funções `min` e `max`?

```
#Exercicio 2c
def minmax(lst):
    min=99999999          #valor muito baixo
    max=-99999999        #valor muito elevado
    for num in lst:
        if num<min: min=num
        if num>max: max=num
    return min, max
#print(minmax([1,2,33,4,5,6, 999, -8]))
```

- d) Recorra às funções anteriores para fazer um programa que leia uma lista de números, determine o valor médio entre o mínimo e o máximo e conte quantos números são inferiores a esse valor.

```
#Exercicio 2d
lst=inputIntList()
min, max =minmax(lst)
print("Valor minimo: ", min)
print("Valor maximo: ", max)
media = (min+max)/2
print("Valor medio: ", media)
print("Elementos inferiores ao valor medio: ", countLower(lst, media))
```

3. Crie uma função que conte quantos dígitos há numa dada string. Por exemplo: `countDigits("23 mil 456")` deve devolver 5. *Sugestão: o método `isdigit` verifica se uma string só tem dígitos, e.g., `"2".isdigit()` -> `True`.*

```
#Exercicio 3
def countDigits(str):
    count=0
    for c in str:
        if c.isdigit(): count+=1
    return count
print(countDigits("23 mil 456"))
```

4. Crie uma função que, dado um nome, crie uma versão abreviada, formada apenas pelas letras maiúsculas. Por exemplo:

```
shorten("Universidade do Porto") -> "UP",
shorten("United Nations Organization") -> "UNO".
```

Sugestão: o método `isupper` verifica se uma string só tem maiúsculas, e.g., `"A".isupper()` -> `True`.

```
#Exercicio 4
def shorten(str):
    newStr=""
    for c in str:
        if c.isupper(): newStr+=c
    return newStr
print(shorten("Universidade do Porto"))
print(shorten("United Nations Organization"))
```