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

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`.*
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`.