

Definições por compreensão e expressões geradoras**Pesquisa, ordenação e expressões lambda****Exercícios**

1. Usando o Python em modo interativo, execute as instruções abaixo e interprete os resultados. Tente prever os resultados de cada expressão

<pre> Lx = [1, 3, 5, 7, 9] [10+x for x in Lx] Ly = [2, 4, 6] [x+y for x in Lx for y in Ly] {x+y for x in Lx for y in Ly} [(x,y) for x in Lx for y in Ly] [(x,y) for y in Ly for x in Lx] [x*c for c in "abc" for x in Lx] [x%3==0 for x in Lx] [(x,x//3) for x in Lx if x%3==0] {x:x//3 for x in Lx if x%3==0} [(x,y) for x in Lx for y in Ly if x<y] { x:[y for y in Ly if x<y] for x in Lx } any(x%2==0 for x in Lx) L = ["Mario", "Carla", "anabela", "Maria", "nuno"] sorted(L) sorted(L, reverse=True) sorted(L, key=len) </pre>	<pre> L[0].casefold() str.casefold(L[0]) #equivalent sorted(L, key=str.casefold) def lenFold(s): return (len(s), s.casefold()) lenFold(L[0]) sorted(L, key=lenFold) D = [('Republic', 1910, 10, 5), ('Christmas', 1, 12, 25), ('Liberty', 1974, 4, 25), ('Restoration', 1640, 12, 1)] sorted(D) sorted(D, key=lambda t:(t[2],t[3])) N = [3,4,4,4,6,7,7,8] import bisect bisect.bisect_left(N,6) bisect.bisect_left(N,10) bisect.bisect_left(N,4) bisect.bisect_right(N,4) </pre>
---	---

2. O programa `imctable2.py` define uma lista com informação dos nomes, pesos e alturas de diversas pessoas e usa uma *list comprehension* para obter uma lista com os nomes apenas. Substitua as reticências por outras *list comprehensions* que produzam:
 - a) Uma lista com os valores de IMC de todas as pessoas.
 - b) Uma lista de tuplos das pessoas com altura superior a 1.7m.
 - c) Uma lista com os nomes das pessoas com IMC entre 18 e 25.
3. Na aula 08 fez um programa que conta ocorrências de letras num ficheiro de texto. Faça uma nova versão desse programa que liste o resultado por ordem decrescente do número de ocorrências. Use o método `sorted` com os argumentos `key=` e `reverse=` para ordenar a sequência de pares chave-valor (`items`) do dicionário.

```

$ python3 countLetters2.py pg3333.txt
e 33406
a 32088
o 28598

```

4. O programa `tabelaFutebol.py` tem uma lista com a tabela classificativa de um campeonato de futebol. Cada elemento da lista é um tuplo com o nome da equipa e números de vitórias, de empates, de derrotas, de golos marcados e sofridos. O programa

já tem uma função para mostrar a tabela devidamente formatada e uma função, definida por uma expressão lambda, para determinar o número de jogos realizados por uma equipa. Complete o programa nos locais indicados para resolver cada uma das alíneas.

- a) Complete a expressão lambda para definir a função `pontos` que, dado um registo de uma equipa, deve devolver o número de pontos da equipa. (Cada vitória vale 3 pontos, cada empate vale 1 ponto.)
 - b) Acrescente os argumentos adequados à função `sorted` para obter uma tabela ordenada por ordem decrescente de pontos.
 - c) Acrescente os argumentos adequados à função `sorted` para obter uma tabela ordenada por ordem decrescente da diferença de golos marcados e sofridos.
 - d) Acrescente os argumentos adequados à função `sorted` para ordenar a tabela por ordem decrescente de pontos e, se iguais, pela diferença de golos.
5. Faça uma função que calcule a mediana de uma lista de valores. A mediana é um valor que é maior que metade dos valores da lista e menor que a outra metade. Se a lista tiver um número ímpar de valores, a mediana é o valor a meio da lista ordenada. Se a lista tiver um número par de valores, a mediana é a média dos dois valores a meio da lista ordenada.
6. O programa `interests.py` tem uma tabela (dicionário) com os interesses de um conjunto de pessoas. Substitua as reticências por expressões adequadas para:
- a) Criar um dicionário com os interesses comuns a cada par de pessoas. Ou seja, a cada par de pessoas, deve associar o conjunto dos interesses comuns a ambos. Note que se incluir o par (X, Y) não deve incluir (Y, X).
 - b) Achar o maior número de interesses em comum. *Sugestão: use a função `max` e uma expressão geradora que percorra o dicionário criado na alínea anterior.*
 - c) Criar uma lista dos pares de pessoas que têm o número máximo de interesses comuns.
 - d) Criar uma lista de pares de pessoas com menos de 25% de similaridade de interesses. Para medir a similaridade, use o índice de Jaccard entre dois conjuntos, que é dado pela razão entre o tamanho da interseção e o tamanho da união entre os conjuntos. O resultado esperado é o seguinte.
7. O ficheiro `wordlist.txt` contém uma lista de palavras de língua inglesa, por ordem. Leia essas palavras para uma lista e, usando uma função de pesquisa binária (do módulo `bisect`), descubra quantas palavras começam por “ea”, sem ter de percorrer tudo. *Sugestão: procure a primeira palavra com “ea” e a primeira com “eb” e subtraia os índices.* E quantas palavras começam por “tb”? Nenhuma? Então qual é primeira letra, maior que ‘b’, que ocorre após um “t”, nas palavras inglesas?
8. Usando o mesmo princípio, faça uma função que indique todas as letras que podem suceder a um certo prefixo. Pode usar esta função num sistema de escrita inteligente que vai apresentando as letras possíveis para completar um certo prefixo já introduzido. Quando o utilizador introduz mais uma letra, é acrescentada ao prefixo anterior e apresenta-se nova lista de possibilidades e assim sucessivamente.