

Mod 4.3 - Listas

① Created	@November 2, 2022 4:22 PM
	Algoritmia e estruturas de dados
Materials	https://moodle.esmad.ipp.pt/pluginfile.php/7451/mod_resource/content/1/AED-Modulo%2004.3%20-%20Listas.pdf
☑Reviewed	
Linkexplicativo	

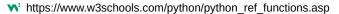
Uma lista é uma sequência de dados e se define por []

→ sequencia de posicoes em que a primeira posicao é 0

funcoes builtin

Python Built-in Functions

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL, Java, and many, many more.





lista métodos built-in

max(lista / min(lista):

vai dar o maior número e quando for texto vai dar a palavra que tem a letra mais próxima do Z (ou seja, seria como se o A valesse 0 e o Z 24, então o z é maior que o A) qualquer número é menor que uma letra, então numa lista mista os números apareceriam antes das letras

• Count:

Serve para contar quantas vezes um elemento aparece dentro de uma lista

Exemplo: if newList.count(lnumbers[i]) ==0 newList.append(lnumbers[i])

É uma condição, se o numero da lista lnumbers não exisitr dentro da nova lista (newList), o código adiciona esse numero a nova lista.

• Enumerate:

Pega objetos iteráveis (que podem se repetir) e transforma numa tuple, podendo pegar tanto o index quanto o valor dos elementos dessa lista

Sintaxe: enumerate(lista , start)

se nada for colocado em start, ele comeca a contar o index do 0



Note: An **iterable** is any Python object capable of returning its members one at a time, permitting it to be iterated over in a for-loop.



→ Para pegar tanto o index quanto o valor de um (ou de cada) elemento em uma lista: usar enumerate(lista). Exemplo:

```
def positiveList(nomes, notas):
    notasPositivas = []
    nomesPositivos = []
   for i, nota in enumerate(notas):
        if nota >= 10:
            notasPositivas.append(nota)
            nomesPositivos.append(nomes[i])
    return notasPositivas, nomesPositivos
listaNotas = []
listaNomes = []
for i in range(10):
    nome = input("Nome {0}: ".format(i + 1))
    listaNomes.append(nome)
    nota = int(input("Pontuacao {0}: ".format(i + 1)))
    listaNotas.append(nota)
resultado = positiveList(listaNomes, listaNotas)
print(resultado)
```

No "for i, nota": i é o índice/index e nota é o valor desse índice

A função "enumerate" é usada para iterar sobre uma lista, mantendo um contador do índice do elemento atual. Por exemplo, suponha que você tenha uma lista "lines" com várias linhas de texto e queira escrever cada linha em um arquivo de texto, mas também queira manter um rastreamento do número da linha atual para fins de depuração. Em vez de escrever:

```
for i in range(len(lines)):
    line = lines[i]
    print("Linha", i, ":", line)
```

Você pode usar "enumerate" para escrever:

```
for i, line in enumerate(lines):
   print("Linha", i, ":", line)
```

A função "enumerate" retorna um objeto iterável que produz pares (índice, elemento) a cada iteração, onde "índice" é o índice do elemento na lista "lines" e "elemento" é o elemento em si.

→ Para achar a posição de um elemento dentro de uma lista: lista.index(valor). Mas só acha a primeira posicao do número. Exemplo:

```
def maiorFaturacao(faturacoes):
    maiorFatuc = max(faturacoes)
    posicao = faturacoes.index(maiorFatuc)
    mes = meses[posicao]
    return mes
```

Nesse caso, "faturações" é uma lista e queremos achar a posicao da maior faturação encontrada e colocamos essa posicao dentro de uma variavel

→ Para usar o index de uma lista, fazendo um range e a cada ciclo pega a info de um lugar da lista: .format(lista[i])

```
meses = ["janeiro", "fevereiro", "março", "abril", "maio", "junho", "julho", "ago
faturacoes = []
for i in range(len(meses)): # ou for i in range(12)
    fat = int(input("Faturação do mês {0}:-".format(meses[i])))
    faturacoes.append(fat)
```

→ Para saber se uma lista está vazia: if not lista

```
def searchNumber(lista, pesquisa):
    posicoes = []
    for pos, i in enumerate(lista):
        if i == pesquisa:
            posicoes.append(pos)
        if ·not · posicoes: · · # · checa · se · a · lista · está · vazia
            return -1
        return posicoes
```

Para inverter a ordem dos elementos de uma lista

lista.reverse()

Exemplo:

```
def reverseStr(str):
    palavras = []
    div = str.split(" ")
    for i in div:
        palavras.append(i)
    palavras.reverse()
    for palavra in palavras:
        print (palavra, end=" ")

str= "0i meu nome é bia"
reverseStr(str)
```

Resultado:

```
PS D:\ESMAD\AED\AED-exercicios\AED-refaz-exs> & C:/Use SMAD/AED/AED-exercicios/AED-refaz-exs/testes anteriore bia é nome meu Oi
PS D:\ESMAD\AED\AED-exercicios\AED-refaz-exs> []
```

Para contar quantas vezes uma palavra é repetida em uma lista:

```
def contagem (str):
 palavras = []
    div = str.split(" ")
    for i in div:
        palavras.append(i)
    palavras.reverse()
    for palavra in palavras:
        print (palavra, end=" ")
# aqui em baixo!
mais_repetida = None
    mais\_repetida\_contagem = 0
    for palavra in palavras:
        contagem = palavras.count(palavra)
        if contagem > mais_repetida_contagem:
            mais_repetida = palavra
            mais_repetida_contagem = contagem
    print("\nPalavra mais repetida:", mais_repetida)
str = "oi oi oi sou bia bia"
contagem(str)
```