

MODALIDADE:	Aprendizagem +	Escolha um item.	
CURSO:	Técnico/a Especialista em Aplicações Informáticas de Gestão		
UC:	Desenvolver projeto de tecnologias e aplicações informáticas de gestão	CÓDIGO UC:	00891
FORMADOR/A:	Bruno Silva	DATA:	

OBJETIVOS

- Exercícios de listas e subprogramas com a linguagem Python

Exercícios Listas

- Construa um programa para **criar uma variável do tipo lista com o nome “numeros”**. De seguida, faça os seguintes exercícios:
 - Adicione** 4 novos elementos numéricos inteiros com a função **append()**;
 - Adicione** 1 novo elemento **entre o índice 1 e 2** com a função **insert()**;
 - Altere** o valor do **índice 2** para **número a sua escolha**;
 - Remova o índice 1** com a função **pop()**;
 - Mostre quantos elementos existem** na lista com a função **len()**;
 - Percorra a lista de valores com uma estrutura de repetição FOR e mostre os dados do índice e valor que percorre no momento;
- Construa um programa para **criar uma variável do tipo lista com o nome “cidades”**. De seguida, faça os seguintes exercícios:
 - Adicione** 4 novos elementos textuais com a função **append()**;
 - Adicione** 1 novo elemento **entre o índice 2 e 3** com a função **insert()**;
 - Altere** o valor do **índice 1** para **um valor textual a sua escolha**;
 - Remova o índice 3** com a função **pop()**;
 - Mostre quantos elementos existem** na lista com a função **len()**;
 - Percorra a lista de valores com uma estrutura de repetição FOR e mostre os dados do índice e valor que percorre no momento;

3. Neste exercício vão ser fornecidas duas listas de Python com valores inteiros. Cada lista representa os gastos de 5 meses da Ana e do João:

gastos_ana = [310, 250, 210, 800, 550]

gastos_joao = [275, 410, 525, 321, 250]

O objetivo é criar duas variáveis para guardar a soma dos gastos de cada um ao longo dos 5 meses. Utilize a função **sum()** para somar os valores de cada lista e a seguir, deve utilizar uma estrutura de decisão dupla encadeada para verificar qual dos dois teve mais gastos ou se tiveram os mesmos gastos.

4. Neste exercício vamos ter um conjunto de temperaturas para analisar.

temperaturas = [18.5, 20.0, 19.3, 21.7, 22.1, 17.8, 16.9, 23.4, 24.0, 19.8]

O objetivo será elaborar um programa para:

- Utilizar a função **max()** e descobrir o **valor numérico mais alto**;
- Utilizar a função **min()** e descobrir o **valor numérico mais baixo**;
- Utilizar a função **sum()** e **somar todos os valores**;
- Utilizar a função **len()** e **contar quantos elementos existem numa lista**;
- Utilizar a função **mean()** e mostrar a **média** (mas precisa de importar o módulo **statistics**)

5. Elabora um programa para que seja capaz de encontrar e exibir a maior e a menor palavra da lista (em número de caracteres).

lista_palavras = ["Braga", "Bragança", "Lisboa", "Viseu", "Leiria", "Porto"]

Deve utilizar as funções **min()** e **max()** para encontrar e exibir a maior e a menor palavra da lista.

6. Elabora um programa para encontrar o segundo maior valor da lista mais abaixo:

numeros = [32, 10, 58, 30, 55, 12, 28, 51]

Dica: primeiro faça a ordenação da lista com a função **sort()**. Como os valores foram ordenados, o segundo maior valor corresponderá ao penúltimo elemento da lista, e podemos utilizar o valor da indexação para obter o valor de uma posição, como por exemplo, **numeros[-2]**.

7. Construa um programa para **criar uma lista com o nome “notas”**. De seguida, vamos implementar os seguintes passos:
- Deve utilizar uma estrutura de repetição para pedir os números ao utilizador e adicionar os mesmos na lista. A estrutura de repetição só deve parar quando o utilizador introduzir o valor 0.
 - O programa só pode aceitar números decimais entre 1 e 20.
 - No final, deverá mostrar:
 - i. **valor numérico mais alto;**
 - ii. **valor numérico mais baixo;**
 - iii. **média dos números;**
 - iv. **quantos números existem na lista;**

Exercícios Subprogramas

8. Construa um programa para **criar uma função com o nome “Mostra_Nome” e receba 2 parâmetros de entrada: nome e idade. No subprograma, deve mostrar os valores que recebeu no parâmetro de entrada.**
9. Construa um subprograma para **criar uma função com o nome “Soma_Numeros” e receba 2 parâmetros de entrada. No subprograma, deve somar os valores que recebeu nos parâmetros de entrada e mostrar o resultado da soma.**
10. **Refaça o exercício anterior**, mas desta vez, deve **retornar o resultado da soma para o programa principal.**
11. Construa um subprograma para **criar uma função com o nome “Calcula_Area_Triangulo” e receba dois parâmetros de entrada: base e altura. No subprograma, deve calcular a área dos valores que recebeu nos parâmetros de entrada e mostrar o resultado.**
12. Construa um subprograma para **criar uma função com o nome “Calcula_Media_Notas” e receba 3 parâmetros de entrada: nota1, nota2, nota3. No subprograma, deve calcular a média das notas que recebeu nos parâmetros de entrada e mostrar o resultado.**

13. Construa um subprograma para **criar uma função com o nome “Calcula_Saldo” e receba 2 parâmetros de entrada: saldo e levantar. No subprograma,** deve calcular a diferença entre os dois valores, mostrar o resultado.

Se o cálculo do valor for negativo deve exibir a mensagem “Não pode retirar valores abaixo de 0”, **senão,** deve mostrar o saldo final da operação.