

 Neste guião irás continuar a aprender a utilizar **listas** em **PYTHON**. As **listas** são **variáveis multidimensionais** que podem **guardar diversas informações** na **mesma variável**. Cada **conteúdo** é posteriormente **acedido** por **posições numéricas** em que podem ser **aplicados ciclos** para as percorrer.

 Cria um **novo programa** de nome **6listaimprime.py**

 Cria uma **lista** com o nome **compras** com **6 posições** no **total preenchidas** da seguinte forma:


```
compras=["Cebolas", "Leite", "Pão", "Carne", "Batatas", "Peixe"]
```

 Aplica um ciclo **FOR** que te **permita imprimir** toda a lista apenas com **uma instrução**.
Relembro a sintaxe do FOR:

```
for contador in range (início, fim, salto):
```

Ajuda: lembra-te que podes aceder ao conteúdo de cada posição através do seu índice numérico, exemplo:

```
print(lista[2])
```

 Guarda o teu **programa**.


 Cria um **novo programa** de nome **6listajogos.py**


 Cria uma **lista** com o nome **jogos** com **5 posições** no **total preenchidas** da seguinte forma:


```
jogos = ["Tomb Raider", "CoD", "CS", "UFO", "Serious Sam"]
```

 **Imprime** a lista no ecrã através de `print(nome_lista)`

 Acrescenta **código** que te **permita imprimir** o jogo **CS** e **Serious Sam** no ecrã.

 **Sabendo** que podes **acrescentar** jogos ao **final** da lista através de `jogos.append("jogo a acrescentar")`, acrescenta **dois jogos** ao teu gosto desta forma à lista.


 **Sabendo** que podes **acrescentar** um **jogo** numa **posição específica** através de `jogos.insert(pos, "jogo a acrescentar")` insere um **jogo** entre os jogos **CS** e **UFO**.

 **Substitui** o valor **"CS"** por **"CS: GO"** utilizando a **posição numérica**. Exemplo:


```
lista[1]="novo valor"
```

 **Imprime, por linha**, todos os **conteúdos** da lista final utilizando os **índices numéricos**, mas **sem ciclos**.

 Aplica um ciclo **FOR** que te **permita imprimir** toda a lista apenas com **uma instrução**.


 Guarda o teu **programa**.

 Cria um **novo programa** de nome **6listafilmes.py**

 Cria uma **lista** de nome **filmes** vazia.

```
filmes = []
```



 Faz com que as **posições** da **lista** sejam **preenchidas através** de **INPUTS** do **utilizador**. Deves utilizar um **ciclo** para **solicitar 3 vezes** o **INPUT** do **utilizador** e conforme ele **insere** um **valor** vai **adicionando** à **lista**.

PRO: solicita os filmes desta forma:

Nome filme 1:


Nome filme 2:

Nome filme 3:


Onde o valor 1, 2 e 3 são inseridos automaticamente utilizando o contador do FOR.


 No **final** deves **imprimir** a informação da **lista** no **ecrã**:

Os seus filmes favoritos são: x, y e z

 **Guarda** o teu **programa**.

 **Cria** um novo **programa** de nome **6listacopia.py**

 **Cria duas listas** com os **nomes original** e **copia** em que o **original** possui **5 nomes** de **jogos** ao teu **gosto** e a **copia** está, para já, **vazia**.

 Neste programa pretende-se que **cries** uma **cópia** da tua **lista original** para **dentro** da lista **copia**. Utiliza um **ciclo** que faça isso **automaticamente**.

NOOB: faz o ciclo 5 vezes.


Acrescenta um **novo jogo** à **lista original** e **altera** o **ciclo** para que **copie também** este **novo jogo** para a **lista copia**.

PRO: Como **limite** de **vezes** para o **loop** deves utilizar **len(nome_lista)** para saber o **tamanho** da **lista**.


Adiciona um **novo jogo** à **lista original** e **verifica** que **automaticamente** é **copiada** para a **outra lista** sem que **necessites** de **alterar** o **ciclo**.

 **Imprime**, no final a **lista copia** e **verifica** se foi **tudo copiado**.


*Desafio: cria uma **nova lista** de nome **invertido** que contenha o mesmo que a lista original, mas **ordenados do final** para o **início**.*

 **Guarda** o teu **programa**

 **Cria** um novo programa de nome **6maxmin.py**

 Já **realizamos** um **exercício semelhante** a este (**4.3minmax.py**). A grande **diferença** é que na altura **não sabias** o que eram **listas** e sempre que querias **guardar** um **valor introduzido** pelo **utilizador** utilizavas **mais uma variável**. Neste caso vais utilizar **listas** para **guardar** tudo o que o **utilizador insere**, poupando **imensas linhas** de **código**. Podes, em vez de criar um ficheiro novo **aproveitar** o **código** do **4.3minmax.py**

 **Preenche** as **posições** da lista através de **inputs** do **utilizador**.

 **Cria** um **algoritmo** que **analise** cada **posição** da lista e **guarde** o **valor maior** e menor **introduzido** pelo **utilizador**. Está atento à **imagem seguinte**, para entenderes o **pedido**:



```
===== RESTART: F:/Escola18ano/12anoAPIb/Python/6.maxmin.py =====
>>>
Insira valor maior do que zero: 78
Insira valor maior do que zero: 45
Insira valor maior do que zero: 4
Insira valor maior do que zero: 245
Insira valor maior do que zero: 23
Máximo: 245
Mínimo: 4
>>>
```

Ajuda: compara cada valor introduzido pelo utilizador com o anterior e guarda apenas o maior/menor.



Guarda o teu programa.



Chama o teu professor para avaliar.



Chama o teu professor para avaliar. Mostra também os outros programas que realizaste.



Envia todos os programas para a tua pasta Programas na cloud.