

# **Exercicis**

# **Python**

**Miguel Navarro Bellido**  
**Gestión Empresarial**  
**2º Dam**

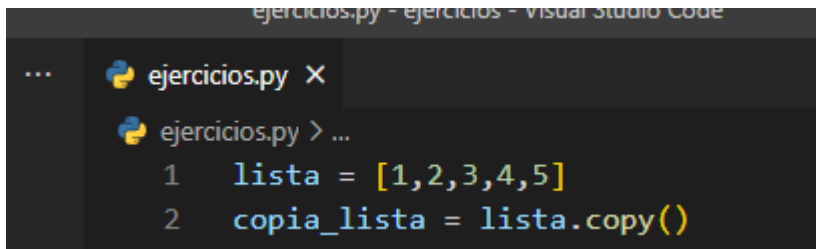
# Índex

Clonar una llista .....	Pág1
¿Quina és la diferència en Python entre “shallow copy” i “deep copy”? .....	Pág1
Afegir un element a una llista .....	Pág1
- Llevar un element a una llista .....	Pág2
- Crear una nova llista amb els 4 últims elements d’una llista .....	Pág2
- Convertir les paraules d’una cadena (separades per espai) a una llista .....	Pág2
- Comentaris amb una línia .....	Pág2
- Comentaris multilínia .....	Pág3
<b>Crea un exemple amb 3 funcions que:</b>	
- Reva 2 números i torne la suma .....	Pág3
- Reva una llista i modifiqui eixa mateixa llista (referència) doblant els valors de tots els elements. No ha de retornar res .....	Pág3
- Reva una llista i torne una còpia de la llista mateixa llista (referència) doblant els valors de tots els elements. La llista original no hi ha de modificar-se .....	Pág4
- Explica amb un exemple, com passar diversos paràmetres des de consola a un programa Python 3 .....	Pág4

## SGE-Unit\_05-Act\_01 Navarro.Miguel

### Activitat 05.01

#### - Clonar una llista.



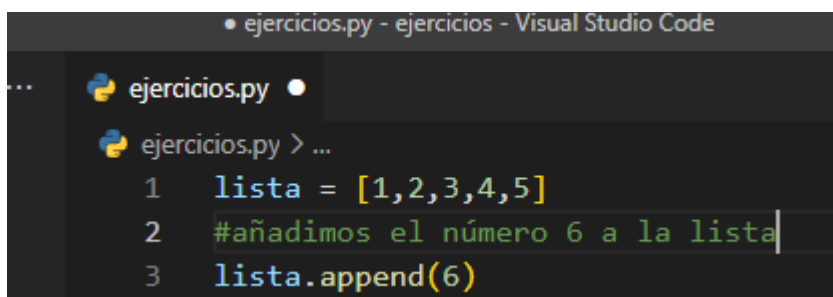
```
ejercicios.py - ejercicios - Visual Studio Code
...
ejercicios.py X
ejercicios.py > ...
1 lista = [1,2,3,4,5]
2 copia_lista = lista.copy()
```

#### - ¿Quina és la diferència en Python entre “shallow copy” i “deep copy”?

Amb “shallow copy”, s'emmagatzema una còpia de l'objecte original i finalment només es copia la direcció de referència. Amb “deep copy”, s'emmagatzema la còpia de l'objecte original i les còpies repetitives. Per a objectes relativament simples, “shallow copy” funciona bé. No obstant això, si els nostres objectes contenen altres objectes, només es copiarà la referència a aquests amb els problemes que això comporta. Si s'utilitza “deep copy”, si qualsevol objecte de la còpia es modifica no hi haurà cap repercussió sobre l'original.

#### - Afegir un element a una llista.

Afegim el 6 a la llista.



```
ejercicios.py - ejercicios - Visual Studio Code
...
ejercicios.py
ejercicios.py > ...
1 lista = [1,2,3,4,5]
2 #añadimos el número 6 a la lista
3 lista.append(6)
```

## - Llevar un element a una llista.

Li llevem el 5 a la llista

```
ejercicios.py - ejercicios - Visual Studio Code
... ejercicios.py X
ejercicios.py > ...
1 lista = [1,2,3,4,5]
2 #quitamos el número 5 a la lista
3 lista.remove(5)
```

## - Crear una nova llista amb els 4 últims elements d'una llista.

```
ejercicios.py - ejercicios - Visual Studio Code
... ejercicios.py X
ejercicios.py > ...
1 lista = [1,2,3,4,5]
2 copia_lista = [lista[-4],lista[-3],lista[-2],lista[-1]]
3
```

## - Convertir les paraules d'una cadena (separades per espai) a una llista.

```
ejercicios.py X
ejercicios.py > ...
1 frase = "Que pases unas buenas fallas"
2 lista = frase.split(" ")
```

## - Comentaris amb una línia.

```
6 #Comentario de una línea
```

## - Comentaris multilínia.

```
6  ▾ '''Comentario de
7    varias
8    líneas'''
```

## Activitat 05.02

Crea un exemple amb 3 funcions que:

- Reva 2 números i torne la suma.

```
ejercicios.py > suma
1  ▾ def suma(num1,num2):
2    return num1+num2
```

- Reva una llista i modifique eixa mateixa llista (referència) doblant els valors de tots els elements. No ha de retornar res.

```
ejercicios.py X
ejercicios.py > doblar_valores
1  def doblar_valores(lista):
2    for i in range(len(lista)):
3    lista[i] *= 2
```

- Reva una llista i torne una còpia de la llista mateixa llista (referència) doblant els valors de tots els elements. La llista original no hi ha de modificar-se.

```
ejercicios.py X
ejercicios.py > doblar_valores
1  #importamos copy para hacer la copia
2  import copy
3
4  def doblar_valores(lista):
5      copia_lista = copy.deepcopy(lista)
6      for i in range(len(copia_lista)):
7          copia_lista[i] *= 2
8      return copia_lista
```

## Activitat 05.03

Explica amb un exemple, com passar diversos paràmetres des de consola a un programa Python 3.

```
ejercicios.py X
ejercicios.py > ...
1  '''Guardamos en la variable nombre
2  lo que el usuario escribe en consola
3  y se le pasa como parametro a la
4  función saludar'''
5
6  nombre = input("Cómo se llama?")
7
8  def saludar(nombre):
9      print("Hola ", nombre)
10
11  saludar(nombre)
```

ejercicios.py X

ejercicios.py &gt; mayor\_de\_edad

```
1  '''Si queremos hacerlo con
2  valores numéricos'''
3
4  edad = int(input("Cuántos años tienes?"))
5  def mayor_de_edad(edad):
6      if(edad>=18):
7          print("Así que eres mayor de edad...")
8      else:
9          print("Aún eres joven para ir a la disco por la noche")
10
11  mayor_de_edad(edad)
```