

## ACTIVIDAD 01 de Actividades Evaluables 01

- Para clonar una lista en Python 3 la forma mas sencilla es utilizar una formula que se llama "slicing".

```
# Usando slicing para clonar una lista
original = [1, 2, 3, 4, 5]
copia = original[:]
```

Luego existen otros métodos para clonar una lista, serían copy() y deepcopy() del módulo copy. Lo que significa que habría que importar copy.

```
# Usando copy() para shallow copy
import copy
copia_shallow = copy.copy(original)

# Usando deepcopy() para deep copy
copia_deep = copy.deepcopy(original)
```

### Diferencia entre "shallow copy" y "deep copy" en Python

Shallow Copy: Crea una nueva lista, pero los elementos de la lista se referencian en la lista creada anterior (osea la original). Cambios en los elementos internos de la lista clonada afectarán a la lista original si estos elementos son mutables. Por lo tanto, si modificas los objetos anidados dentro de la copia, esos cambios se reflejarán tanto en la copia como en el original.

Deep Copy: Por otro lado, crea una nueva estructura de datos y copia recursivamente todos los elementos internos también. En otras palabras, no solo se copian las referencias a los objetos internos, sino que se crean copias completamente independientes de esos objetos. Por lo tanto, los cambios realizados en la estructura de datos original no afectan a la copia, y viceversa.

```
import copy

# Lista original con una lista anidada
original = [1, 2, [3, 4]]

# Realizar una deep copy
copia_profunda = copy.deepcopy(original)

# Modificar la lista anidada en la copia profunda
copia_profunda[2].append(5)

print("Original:", original)          # Output: Original: [1, 2, [3, 4]]
print("Copia Profunda:", copia_profunda) # Output: Copia Profunda: [1, 2, [3, 4, 5]]
```

Ejemplo de Deep Copy

-Para agregar un elemento a una lista se utiliza el método append().

```
lista = [1, 2, 3]
lista.append(4)
```

- Para quitar un elemento de una lista se utiliza el método remove() o pop().

```
lista = [1, 2, 3, 4]
lista.remove(2) # Eliminar el elemento con valor 2
elemento_eliminado = lista.pop(1) # Eliminar el elemento en la posición 1
```

- Para crear una lista nueva con los 4 últimos elementos que hay usar un metodo de Python que es la indexación negativa para obtener los últimos elementos de la lista. Esta es una forma muy buena para acceder a los elementos de una lista desde el final hacia el principio. Mientras que la indexación normal empieza desde 0 para el primer elemento, la indexación negativa cuenta desde -1 para el último elemento y sigue hacia atrás.

**ATENCIÓN:** Es importante entender que el primer elemento desde el final es -1, el segundo es -2, y así sucesivamente.

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]
ultimos_elementos = lista[-4:] # Obtener los 4 últimos elementos
```

- Para convertir palabras de una cadena a una lista hay que utilizar el método split() para dividir una cadena en palabras y convertirlas en una lista.

```
cadena = "Hola, mi nombre es Jaume"
lista_palabras = cadena.split()
```

Para los comentarios de una línea y comentarios multilínea se utiliza el símbolo # para comentarios de una línea y "" o """ para comentarios multilínea. Muy sencillo.

Comentario de una sola línea

```
lista = [1, 2, 3, 4, 5, 6, 7, 8]
ultimos_elementos = lista[-4:] # Obtener los 4 últimos elementos
```

Comentario multilíneas

```
'''
Así se escribe
un comentario
multilínea
'''

"""
También puedes utilizar
esta forma de comentar
varias líneas. Jaume Mir
"""
```

