REBANADAS

Suponiendo que lista = [1,2,3,4,5,6,7]

- Todos los elementos en posiciones pares o impares: **lista[::2] (par) / lista[1::2] (impar)**

```
print(lista[::2])
print(lista[1::2])
```
```
[1, 3, 5, 7]
[2, 4, 6]
```

Ponemos el índice 1 en el start, el stop lo dejamos por omisión y el paso (salto) que pasa de 2 en 2 para poder considerar solo los indices impares.
Para los pares el start sera 0 por omisión, stop también por omision,
stop también por omisión y salto de 2 en 2 para considerar solo índices pares.

- Invertir todos los elementos de la lista : **lista [::-1]**

```
print(lista[::-1])
```
```
[7, 6, 5, 4, 3, 2, 1]
```

El start comienza con omisión y lo mismo para el stop, rebanamos con salto -1 desde el inicio hasta el final, esto hará que se invierta toda la lista.

- Eliminar elementos consecutivos : **lista [1:3]=[]**

```
lista[1:3]=[]
print(lista)
```
```
[1, 4, 5, 6, 7]
```

En el start pusimos en este ejemplo índice 1 y en el stop índice 3 para abarcar dos números consecutivos, ya que el último índice no se considera.
Luego colocamos un igual y corchetes vacíos (=[]), así se eliminarán de la lista de los números entre los índices establecidos con start incluido, en este caso 1 y 2.

- Reemplazar elementos consecutivos: **lista[0:2]=[4,4]**

```
lista[:2]=[4,4]
print(lista)
```
```
[4, 4, 3, 4, 5, 6, 7]
[4, 4, 3, 4, 5, 6, 7]
```

El start lo dejamos por omisión, que sería el índice 0, y luego el 2 en el stop para abarcar dos números consecutivos, ya que el útlimo no lo considera (osea el índice2 no sufrirá reemplazo) Luego, en el lado derecho del igual (=), ponemos los números que queremos insertar entre corchetes y separados entre comas ([elemento1,elemento2,...elementoN])

- Insertar elementos a una lista: **lista[len(lista)+1:]=[8,7,9]**

```
lista=[1,2,3,4,5,6,7]
lista.insert(3,0)
print(lista)

lista=[1,2,3,4,5,6,7]
lista.extend([8,7,9])
print(lista)

lista=[1,2,3,4,5,6,7]
lista[len(lista)+1:]=[8,7,9]
print(lista)
```
```
[1, 2, 3, 0, 4, 5, 6, 7]
[1, 2, 3, 4, 5, 6, 7, 8, 7, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 7, 9]
```

Insert ({posición},{numero}) “Solo se puede insertar un elemento en la posición elegida, desplazando los demás elementos hacia la derecha.”
extend([{elemento1, elemento2,...elementoN}])” puede insertar varios elementos al final de la lista. Y “[len(lista)+1:]” también inserta varios elementos en forma de rebanada al final de una lista. La otra forma de insertar elementos sin eliminar o + reemplazar, es haciendo una rebanada nula.

- Eliminar todos los elementos no consecutivos en posiciones impares: **del lista[1::2]**

```
del lista[1::2]
print(lista)
```
```
[1, 3, 5, 7]
```

Utilizar una rebanada =[], ejemplo: ‘lista[1:3]=[]’, nos deja eliminar elementos consecutivos, pero a esa rebanada no se le puede establecer un paso mayor al de 1 en 1. Para eso, debemos usar la función “del”. Esta permite establecer un s tart, stop y paso que se desee (es decir, sí permite saltos). Ejemplo: “del lista[1::2]”, ésta función eliminará todos los elementos de la lista ubicados en índices impares. Su forma es “del nombreLista[start:stop:paso]”.

- Rebanadas nulas para insertar elementos: **lista[3:3]=[1,2,3]**

```
lista[3:3]=[1,2,3]
print(lista)
```
```
[1, 2, 3, 1, 2, 3, 4, 5, 6, 7]
```

Al poner start y stop con mismo índice, realizamos una rebanada [] nula, por ende, podemos insertar elemento/s en esa posición, sin eliminar, ni reemplazar otros elementos en la lista. Es decir, los demás elementos de la lista se desplazan.

- ¿Qué sucede si superamos un índice en una rebanada?: **lista[0:230]**

```
print(lista[:230])
```
```
[1, 2, 3, 4, 5, 6, 7]
```

Como se ve en la línea de código, el tamaño de la lista es de 7 índices, y al establecer un índice con rango superior en el stop de una rebanada, ésta busca solo hasta el último índice de la lista sin dar "error por índice fuera de rango'". Lo mismo sucede cuando se utiliza un paso negativo y se establece un índice de start fuera de rango.

BIBLIOGRAFÍA

Programación ATS. (6 de diciembre de 2018). *25. Programación en Python | Colecciones | Listas* https://www.youtube.com/watch?v=xdCJa2QXmJ8

Programación ATS. (7 de diciembre de 2018). *26. Programación en Python | Colecciones | Listas (parte 2)*. https://www.youtube.com/watch?v=aF1tiL5A-iU

Javier Montero. (16 de diciembre de 2015). *Python – Troceando desde el lado izquierdo*. El Club del Autodidacta. http://elclubdelautodidacta.es/wp/2015/12/python_troceando_desde_el_lado_izquierdo/

W3Schools. *Python - Remove List Items*. https://www.w3schools.com/python/python_lists_remove.asp

W3Schools. *Python - Change List Items*. https://www.w3schools.com/python/python_lists_change.asp

W3Schools. *Python - Add List Items*. https://www.w3schools.com/python/python_lists_add.asp

W3Schools. *Python - Access List Items*. https://www.w3schools.com/python/python_lists_access.asp

W3Schools. *Python - List Methods.* https://www.w3schools.com/python/python_lists_methods.asp

veepsk (usuario). (19 de agosto de 2020). *Extraer elementos de la lista en posiciones impares*· Javaer101. https://www.javaer101.com/es/article/3480032.html

Parzibyte. (18 de diciembre de 2018). *Invertir arreglo o lista en Python.* https://parzibyte.me/blog/2018/12/18/invertir-lista-python/