Laboratorio III Mariela Montenegro Redondo.

De que forma podemos agregar, eliminar, modificar y recorrer una estructura tipo list?

La clase list en Python es una de las más utilizadas por su naturaleza, dinamismo, fácil manejo y potencia.
Las listas en Python son un tipo contenedor, compuesto, que se usan para almacenar conjuntos de elementos relacionados del mismo tipo o de tipos distintos.

Añadir elementos a una lista en Python

Para añadir un nuevo elemento a una lista se utiliza el método append() y para añadir varios elementos, el método extend():

```
>>> vocales = ['a']
>>> vocales.append('e') # Añade un elemento
>>> vocales
['a', 'e']
>>> vocales.extend(['i', 'o', 'u']) # Añade un grupo de elementos
>>> vocales
['a', 'e', 'i', 'o', 'u']
```

También es posible utilizar el operador de concatenación + para unir dos listas en una sola. El resultado es una nueva lista con los elementos de ambas:

```
>>> lista_1 = [1, 2, 3]
>>> lista_2 = [4, 5, 6]
>>> nueva_lista = lista_1 + lista_2
>>> nueva_lista
[1, 2, 3, 4, 5, 6]
```

También es posible añadir un elemento en una posición concreta de una lista con el método insert(índice, elemento). Los elementos cuyo índice sea mayor a índice se desplazan una posición a la derecha:

```
>>> vocales = ['a', 'e', 'u']
>>> vocales.insert(2, 'i')
>>> vocales
['a', 'e', 'i', 'u']
```

Modificar elementos de una lista

Es posible modificar un elemento de una lista en Python con el operador de asignación =. Para ello, lo único que necesitas conocer es el índice del elemento que quieres modificar o el rango de índices:

```
>>> vocales = ['o', 'o', 'o', 'o', 'u']
# Actualiza el elemento del índice 0
>>> vocales[0] = 'a'
>>> vocales
['a', 'o', 'o', 'o', 'u']
# Actualiza los elementos entre las posiciones 1 y 2
>>> vocales[1:3] = ['e', 'i']
>>> vocales
['a', 'e', 'i', 'o', 'u']
```

Eliminar un elemento de una lista en Python

En Python se puede eliminar un elemento de una lista de varias formas.
Con la sentencia del se puede eliminar un elemento a partir de su índice:

```
# Elimina el elemento del índice 1
>>> vocales = ['a', 'e', 'i', 'o', 'u']
>>> del vocales[1]
>>> vocales
['a', 'i', 'o', 'u']
# Elimina los elementos con índices 2 y 3
>>> vocales = ['a', 'e', 'i', 'o', 'u']
>>> del vocales[2:4]
>>> vocales
['a', 'e', 'u']
# Elimina todos los elementos
>>> del vocales[:]
>>> vocales
[]
```

Además de la sentencia del, podemos usar los métodos remove() y pop([i]). remove() elimina la primera ocurrencia que se encuentre del elemento en una lista. Por su parte, pop([i]) obtiene el elemento cuyo índice sea igual a i y lo elimina de la lista. Si no se especifica ningún índice, recupera y elimina el último elemento.

```
>>> letras = ['a', 'b', 'k', 'a', 'v']
# Elimina la primera ocurrencia del carácter a
>>> letras.remove('a')
>>> letras
['b', 'k', 'a', 'v']
# Obtiene y elimina el último elemento
>>> letras.pop()
'v'
>>> letras
['b', 'k', 'a']
```

Finalmente, es posible eliminar todos los elementos de una lista a través del método clear():

```
>>> letras = ['a', 'b', 'c']
>>> letras.clear()
>>> letras
[]
```

Recorrer una lista en Python

El caso general, sería hacerlo por medio de un ciclo for que vaya accediendo uno por uno a los elementos o incluso aumentando de uno en uno el índice que queremos acceder de la lista.

```
edades = [20, 41, 6, 18, 23]

# Recorriendo los elementos
for edad in edades:
    print(edad)

# Recorriendo los índices
for i in range(len(edaded)):
    print(edaded[i])

# Con while y los índices
indice = 0

while indice < len(edaded):
    print(edades[indice])
    indice += 1
```