Exploring Python List Methods

Python List Methods

# append()

Description: It adds a single element to the end of the list.

Syntax: example\_list.append(element)

Return Type: None

Example:  
example\_list :int= [1, 2, 3]  
example\_list.append(4)

Output: [1, 2, 3, 4]

# 2. extend()

Description: Extends the list by appending elements from another list.

Syntax: list1.extend(list2)

Return Type: None

Example:  
example\_list :int= [1, 2, 3]  
example\_list.extend([4, 5])  
Output: [1, 2, 3, 4, 5]

# 3. insert()

Description: Inserts an element at a specified position in the list.

Syntax: list.insert(index, element)

Return Type: None

Example:

example\_list :int= [1, 2, 3]  
example\_list.insert(1, 10)  
# Output: [1, 10, 2, 3]

# 4. remove()

Description: Removes the first found of the specified element from the list.

Syntax: list.remove(element)

Return Type: None

Example:

example\_list :int= [1, 2, 3, 2]  
example\_list.remove(2)

# Output: [1, 3, 2]

# 5. pop()

Description: Removes and returns the element at the specified position

Syntax: list.pop([index])

Return Type: Removed Element

Example:

example\_list :int= [1, 2, 3]  
popped\_element = example\_list.pop()  
Output: [1, 2]

example\_list.pop(1)

Output: [1, 3]

# 6. clear()

Description: Removes all elements from the list, leaving it empty.

Syntax: list.clear()

Return Type: None

Example:

example\_list :str= [‘A’, ‘B’, ‘C’]  
example\_list.clear()  
# Output: []

# 7. index()

Description: Returns the index of the first occurrence of the specified element.

Syntax: list.index(element, start, end)

Return Type: int

Example:  
example\_list :int= [1, 2, 3, 2]  
idx = example\_list.index(2)  
# Output: 1

# 8. count()

Description: Returns the number of times a specified element appears in the list.

Syntax: list.count(element)

Return Type: int

Example:  
example\_list :int= [1, 2, 3, 2]  
count\_of\_twos = example\_list.count(2)  
# Output: 2

# 9. sort()

Description: Sorts the elements of the list in ascending order by default. Can be customized with parameters.

Syntax: list.sort(key=None, reverse=False)

Return Type: None

Example:  
example\_list :int= [3, 1, 2]  
example\_list.sort()  
# Output: [1, 2, 3]

Words :str = ['apple', 'banana', 'kiwi', 'cherry']

words.sort(key=len)

# Output: ['kiwi', 'apple', 'banana', 'cherry']

# 10. reverse()

Description: Reverses the elements of the list in place.

Syntax: list.reverse()

Return Type: None

Example:

example\_list :int= [1, 2, 3]  
example\_list.reverse()  
# Output: [3, 2, 1]

# 11. copy()

Description: Returns a shallow copy of the list.

Syntax: list.copy()

Return Type: A new list containing the same elements as the original list.

Example:  
example\_list :int= [1, 2, 3]  
copied\_list = example\_list.copy()  
# Output: [1, 2, 3]