pop(): Remove the last item

```
Copy Bdit
  python
  tea_varities.pop()
   Removes and returns the last item from the list ('Oolong').
   List after popping:
                                                                            Copy Bdit
      python
      ['Black', 'Green', 'Masala', 'White']
    remove("item"): Remove specific item
                                                                            Copy Dedit
  python
  tea_varities.remove("Masala")
   Finds and removes the first occurrence of "Masala".
List becomes:
```

```
python

['Black', 'Green', 'White']
```

insert(index, value): Insert an item at a specific position

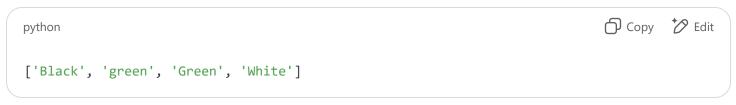
```
python

tea_varities.insert(1, "green")

Copy * Edit
```

• Inserts "green" at index 1.

Resulting list:



Note: Now you have both 'green' (lowercase) and 'Green' (capitalized). Python is case-sensitive.



Wrong way:

```
Copy Dedit
python
tea_varities_copy = tea_varities
```

- This does **not** create a real copy.
- Both variables point to the same list in memory.

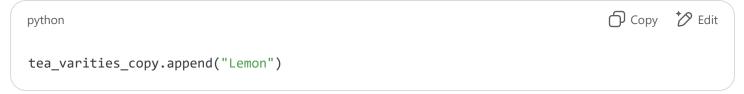
Right way:



This creates an independent copy.



append(): Add an item to the end of the list



Adds "Lemon" to the end of the copy only, not the original.

Result Check

python

tea_varities # ['Black', 'green', 'Green', 'White'] tea_varities_copy # ['Black', 'green', 'Green', 'White', 'Lemon']

Perfectly done! Now you have:

- Original list unchanged.
- Copy with extra item.

Summary of New Concepts

Operation	Code Example	What It Does
pop()	<pre>lst.pop()</pre>	Removes and returns last item
remove()	<pre>lst.remove("item")</pre>	Removes first match of "item"
insert()	<pre>lst.insert(1, "new")</pre>	Inserts at a specific index
copy()	<pre>copy_lst = lst.copy()</pre>	Creates a shallow copy of the list
append()	<pre>lst.append("item")</pre>	Adds item to the end