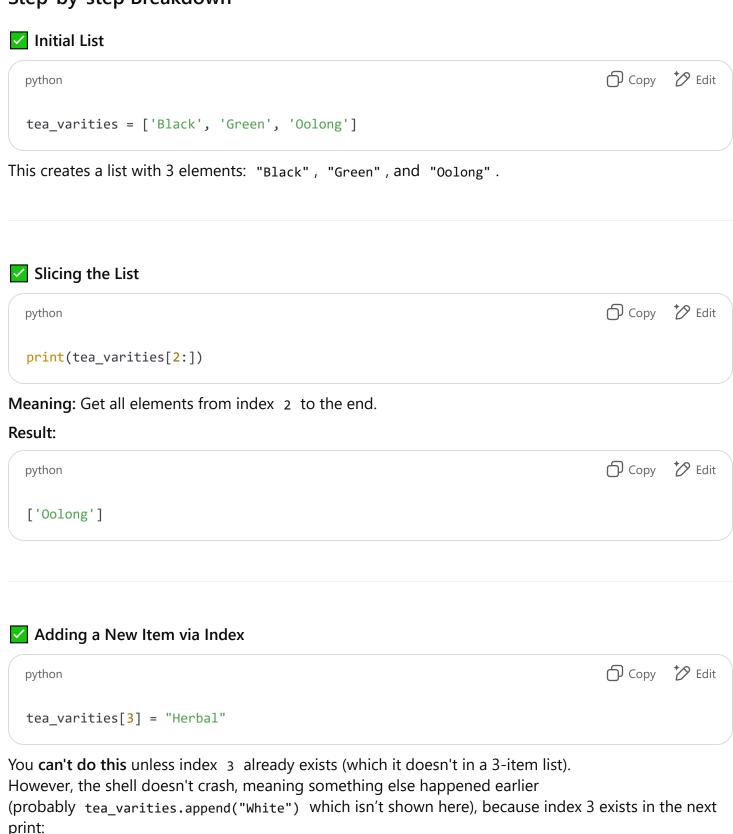
Step-by-step Breakdown



print(tea_varities) # ['Black', 'Green', 'Oolong', 'Herbal']

python

Copy Bedit

Now "Herbal" was assigned at position 3.

Mistake: Replacing Slice with a String

```
Copy Dedit
python
tea_varities[1:2] = "Lemon"
```

What's Happening:

This is replacing a slice (index 1 only: 'Green') with "Lemon", but "Lemon" is a string — and strings are iterable, so Python replaces the item at index 1 with each letter from "Lemon":

Resulting list:

```
Copy Bedit
python
['Black', 'L', 'e', 'm', 'o', 'n', 'Oolong', 'Herbal']
```

Resetting the List

```
Copy Bedit
python
tea_varities = ["Black", "Green", "Oolong", "White"]
```

You're just reassigning the list back to a clean version.

Correct Slice Replacement

```
Copy Bedit
python
tea_varities[1:2] = ["Lemon"]
```

What this does:

It replaces just the item at index 1 ('Green') with "Lemon" (as a list now).

Result:

python

['Black', 'Lemon', 'Oolong', 'White']

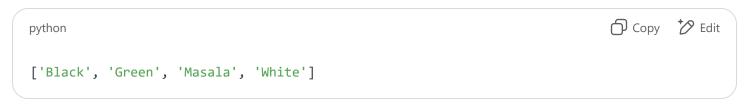
Replacing Multiple Items



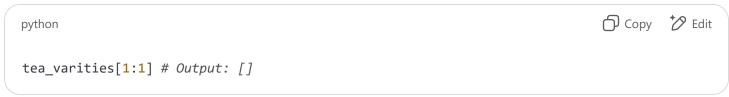
Meaning:

Replace items at index 1 and 2 ('Lemon', 'Oolong') with "Green" and "Masala".

Result:



Empty Slice Output



Why?

This slice means "start at 1, go up to but not include 1" — that range is empty.

Q Summary of Concepts Learned

Concept	Example	Result
List slicing	tea_varities[2:]	Gets a sublist from index 2 onward

Concept	Example	Result
Assigning to slice with string	<pre>tea_varities[1:2] = "Lemon"</pre>	Inserts each character as separate items
Assigning to slice with list	<pre>tea_varities[1:2] = ["Lemon"]</pre>	Correctly replaces item at index
Replacing multiple items	tea_varities[1:3] = ["G", "M"]	Replaces two items with new values
Empty slice	tea_varities[1:1]	Always results in an empty list