🧠 What is reference counting?

Python keeps track of **how many variables or parts of the program are using (or referencing) an object**.
This count is called the **reference count**.

🔄 How it works:

Let's go step by step.

✅ 1. You create a variable:

python Copy Edit

```
x = [1, 2, 3]
```

- Python creates the list `[1, 2, 3]`.
- `x` points to that list.
- So the reference count is now **1**.

✅ 2. You assign another variable:

python Copy Edit

```
y = x
```

- Now both `x` and `y` point to the **same list**.
- The reference count becomes **2**.

✅ 3. You delete one reference:

python Copy Edit

```
del x
```

- Now only `y` points to the list.
- Reference count goes down to **1**.

✅ 4. No references left:

python Copy Edit

```
del y
```

- Now nothing points to the list.
- Reference count is **0** → Python **automatically deletes (frees)** the memory.

🧪 You can check reference count like this:

python Copy Edit

```
import sys a = [1, 2, 3] print(sys.getrefcount(a)) # Output will be 2 (a + temporary reference)
```

📌 Important Notes:

- `getrefcount()` always shows **1 extra** because it adds a temporary reference when you pass the object to the function.
- Python's memory management also includes **garbage collection** to clean up cycles (like `a -> b -> a`).

🔚 Summary:

Action	Ref Count
`x = obj`	1
`y = x`	2
`del x`	1
`del y`	0 → deleted