📘 Lecture 6: Loops and Control Statements in Python

🌀 1. What Are Loops?

Definition: Loops are used to **repeat a block of code** multiple times until a certain condition is met or a sequence is exhausted.

🔁 Types of Loops in Python:

Loop Type	Description	Example
`for` loop	Iterates over a **sequence** (list, tuple, dictionary, etc.)	`for item in list:`
`while` loop	Runs **as long as a condition is True**	`while condition:`

🔷 For Loop Example:

```
python
CopyEdit
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num)
```

Output:

```
CopyEdit
1
2
3
4
5
```

🚦 2. Control Statements

Control statements manage **loop flow** and execution.

🔷 `continue`: Skips current iteration

```
python
CopyEdit
for i in range(10):
    if i == 3:
        print("Skipping 3")
        continue
    print(i)
```

Output:

```
CopyEdit
0
1
2
Skipping 3
4
5
6
7
8
9
```

🔷 `break`: Stops the loop completely

```
python
CopyEdit
for i in range(10):
    if i == 8:
        print("Breaking at 8")
        break
    print(i)
```

Output:

```
CopyEdit
0
1
2
3
4
5
6
7
Breaking at 8
```

🔷 `else` with loops

Executes **only if the loop is not broken**.

```
python
CopyEdit
for i in range(5):
    print(i)
```

```
else:
    print("Loop ended naturally")
```

Output:

```
vbnet
CopyEdit
0
1
2
3
4
Loop ended naturally
```

□ 3. Working with Lists & Indexing

Index: Position of an element in a list (starts from 0)

✓ `enumerate()`: Returns both index and value

```
python
CopyEdit
fruits = ["apple", "banana", "cherry"]
for index, fruit in enumerate(fruits):
    print(f"Index {index}: {fruit}")
```

Output:

```
yaml
CopyEdit
Index 0: apple
Index 1: banana
Index 2: cherry
```

□ Manual Indexing (Alternative to `enumerate()`)

```
python
CopyEdit
index = 0
for fruit in fruits:
    print(f"Index {index}: {fruit}")
    index += 1
```

□ 4. Iterating Over Dictionaries

Dictionary = key-value pairs

```
python
CopyEdit
person = {"name": "John", "age": 30, "city": "New York"}
```

🔹 Loop through keys:

```
python
CopyEdit
for key in person:
    print(key)
```

Output:

```
nginx
CopyEdit
name
age
city
```

🔹 Loop through key-value pairs:

```
python
CopyEdit
for key, value in person.items():
    print(f"{key}: {value}")
```

Output:

```
vbnet
CopyEdit
name: John
age: 30
city: New York
```

🔄 5. Combined Example: Loops + Control Statements

```
python
CopyEdit
for i in range(10):
    if i == 3:
        print("Skipping 3")
        continue
    if i == 8:
        print("Breaking at 8")
        break
    print(i)
else:
    print("Loop ended naturally")
```

Output:

```
CopyEdit
0
1
2
Skipping 3
4
5
6
7
Breaking at 8
```

🧠 Best Practices for Using Loops

✅ Avoid **infinite loops** (always use conditions that eventually become false)
✅ Prefer `enumerate()` for indexed loops
✅ Use `break` & `continue` to manage flow cleanly
✅ Test with different scenarios to build logic

💬 Difficult Words Explained

Term	Meaning
Loop	Repeating a set of instructions
Iteration	One cycle in a loop
Index	Position of an item in a sequence
enumerate()	Built-in function to get index + value
Control Statement	A statement that affects loop execution (`break`, `continue`, etc.)

❓ Review Questions

1. What is the difference between `break` and `continue`?
2. How does the `else` clause work in a `for` loop?
3. Rewrite a `for` loop that prints elements of a list along with their index using `enumerate()`.
4. Write a `for` loop that skips number 5 and stops at 8.

5. Why is `enumerate()` preferred over manual indexing?
