Loops

🧠 What are Loops?

A **loop** is a way to **repeat a set of instructions** multiple times — without writing the same code again and again.

Think of it like this:

> Imagine your mom tells you: “Clean 5 plates.”
>
> You wouldn’t write:
>
> - Clean plate 1
> - Clean plate 2
> - Clean plate 3 ...
>
> You’d just think: “Repeat cleaning until all 5 plates are done.”

That’s exactly what **loops** do — they *repeat tasks automatically*.

🔁 Types of Loops in Python

There are **two main types** of loops:

1. **for loop** → When you know *how many times* you want to repeat
2. **while loop** → When you want to repeat *until something happens*

---

1. for Loop

The **for loop** is used when you want to do something a fixed number of times or go through a sequence (like a list, string, or range of numbers).

🔷 Basic structure:

```
for variable in sequence:
    # repeat this block
```

Let's see it in action:

```
for i in range(5):
    print("Hello!")
```

Output:

```
Hello!
Hello!
Hello!
Hello!
Hello!
```

It printed 5 times!

Because `range(5)` means → numbers from `0` to `4`.

You can think of it like:

```
i = 0 → print("Hello!")
i = 1 → print("Hello!")
i = 2 → print("Hello!")
i = 3 → print("Hello!")
i = 4 → print("Hello!")
```

🔷 Printing numbers

```
for i in range(1, 6):
    print(i)
```

Output:

```
1
2
3
```

```
4
5
```

So here, `range(1, 6)` gives numbers **1 to 5** (6 is excluded).

🔷 Looping through a string or list

```
for letter in "Water":
    print(letter)
```

Output:

```
W
a
t
e
r
```

or

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

---

2. while Loop

A **while loop** runs as long as a condition is *True*.

You don't always know how many times it'll run — it depends on the condition.

🔷 Example:

```
count = 1
while count <= 5:
```

```
    print("Count is:", count)
    count = count + 1
```

Output:

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
```

Let's understand what happens:

1. `count` starts at 1
2. The loop runs **while** `count <= 5`
3. Each time it prints the value and increases it by 1
4. When `count` becomes 6, the condition fails → loop stops

⚠️ Important:

Always make sure your **while loop** changes something inside it — otherwise it will run **forever** (infinite loop).

Example of mistake:

```
x = 1
while x <= 5:
    print(x)
# forgot to increase x → infinite loop!
```

Extra Tools in Loops

🔶 `break`

Stops the loop immediately.

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

Output:

`0 1 2 3 4`

It stops when `i` becomes 5.

🔶 continue

Skips the current iteration and moves to the next one.

```
for i in range(5):
    if i == 2:
        continue
    print(i)
```

Output:

`0 1 3 4`

(Skipped 2)

Mental Model

1. **for loop** → repeat something *a fixed number of times*
2. **while loop** → repeat *until* a condition becomes false
3. **break** → stop early
4. **continue** → skip one time but keep going

Mini Exercises

1. Print numbers from 1 to 10 using a **for loop**.
2. Print only even numbers between 1 and 20 using a **while loop**.

3. Ask the user for a number and print “Hello” that many times.