

Loops in Python

for loop

- Used to iterate over a sequence.
- Sequence can be list, string, tuple, range, etc.
- It automatically goes element by element.

Example idea:

```
for i in sequence:  
    statements
```

while loop

- Used when a condition is true.
- Loop runs until the condition becomes false.
- Condition is checked every time before running.

Important:

- Updating (increment or decrement) inside while loop is compulsory.
- If we don't update, it becomes an infinite loop.

Basic structure:

```
while condition:  
    statements  
    update statement
```

range() function

- Used to generate numbers.
- Index in Python starts from 0.
- range() does not include the upper limit value.

Example:

```
range(10)
```

It gives numbers from 0 to 9.

Using range with len()

Used when we need index and value.

```
for i in range(len(list_name)):  
    print(i, list_name[i])
```

Printing characters using while loop

- We can access characters using index.
- We must increase index every time.

```
i = 0  
while i < len(s):  
    print(s[i])  
    i += 1
```

Reversing a string

- Take an empty string.
- Add characters one by one in reverse order.

Logic:

```
x = s[i] + x
```

Sum of natural numbers using while

- Initialize sum = 0
- Initialize counter i = 1
- Run loop till $i \leq n$
- Add i to sum
- Increase i by 1

Updating counter is necessary.

break statement

- Stops the loop immediately.
- Control goes to the next line after loop.
- In nested loops, it stops only the inner loop.

continue statement

- Skips current iteration.
- Does not stop the loop.
- Moves to next iteration.

Infinite loop example

```
while True:  
    take input  
    if condition satisfied:  
        break
```

- while True creates infinite loop.
- break is used to stop it when condition matches.

Important points

- for loop → used for sequences
- while loop → used for conditions
- range() → upper limit not included
- Index starts from 0
- while loop must have update statement
- break → stops loop
- continue → skips iteration