Python - Day 5 (Beginner)

Topic :- while loops, loops, Range

(1) what are loops?
loops allow you code to repeat tasks automatically

Python has two main types of loops
(1). for loop → when you know how many time to repeat
(2). while lopp → when you repeat until a Condition becomes false

(2) for loops
Syntax

```
for item in collection:
    # action
```

Example

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

Key points
- fruit is a loop variable
- loop runs Once per item
- works with lists, strings, ranges etc.

(3) Looping through Strings

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

- A string behaves like a list of characters
- Loop Executes Once for character

(4) Looping Using range ()

what is range () ?

range (start, stop, step) → generates numbers from start to start - 1

Examples:

```
range (10)     # 0 → 9
range (1,5)    # 1 → 4
range (1,11,2) # 1, 3, 5, 7, 9
```

Example loop using range:

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

Q Add All Even numbers 1 to 100

```
total = 0
for number in range (1,101):
    if number % 2 == 0:
        total += number
print (total)
```

Key Concept: modulo (%) for checking Even numbers.

Q fizzBuzz

- Divisible by 3 → fizz
- Divisible by 5 → Buzz
- Divisible by both → fizzBuzz

```
for number in range (1, 101):
    if number % 3 == 0 and number % 5 == 0:
        print ("fizzbuzz")
    Elif number % 3 == 0:
        print (" fizz")
    Elif number % 5 == 0:
        print ("Buzz")
    Else:
        print (number)
```

important
check both conditions before individual ones.

(5) What is a while loop?
A while loop repeats until the condition becomes false

Syntax

```
while Condition is true:
    # repeat this code
```

(5.1) Basic Example

```
i = 1                      Output
while i <= 5:          #  1 → 5
    print (i)
    i += 1
```

(5.2) Important Difference: for vs while

for loop	while loop
Used when number of iterations is known	used when duration is unknown
works with lists, strings range	works with conditions
Automatically increments	you must manually update variables
less risk of infinite loop	Higher risk of infinite loop

(5.3) Infinite loop (Very important)

Danger of while loops → if condition never becomes false, loop will run forever

Example:

```
x = 1
while x > 0:
    print ("Hello")
```

this never stops because x > 0 is always true

(5.4) Breaking Out of a while loop

Use break to Exit the loop forcefully

Example:

```
while True:
    name = input ("Enter name: ")
    if name == "stop":
        break
    print ("Hello", name)
```

(5.5) Continue Statement

Skip one iteration and move to the next

```
i=0
while i<10:
    i += 1
    if i%2 ==0:
        continue
    print (i)
```

this prints only odd numbers.

(5.6) Input Validation Example Using while loop

This is how while loops are often used in real apps

```
age = int (input ("Enter age: "))
while age <= 0:
    print ("Invalid age. Try again.")
    age = int (input ("Enter age: "))
```

Repeats until user enters a valid age

(5.7) while loop password Retry Example

```
password = "Sakshi"
attempt = " "

while attempt != password:
    attempt = input ("Enter password: ")

print ("Access Granted! ")
```

(*)(*) Mini project - password Generator (*)(*)

This project uses:

- lists
- Random module
- loops
- String joining

Key takeaway :-

Day 5 is all about Controlling repetition in your program

you learnt how to loop through lists, strings and numeric sequence using range(