Day-4

4/7

loops in python.

loop → repeat a block of code multiple times until a condition is met.
→ automate repetitive tasks.

for loop
→ iterate over sequence like lists, tuple, strings

Syntax: for variable in sequence
print (variable)

```
for i in range (1,6)
    print (i)        o/p : 1 2 3 4 5
```

While loop → Executes as long as condition is True.

Syntax: while condition
#code.

```
i=1
while i<=4:
    print (i)        o/p. 1 2 3 4
    i+=1
```

Nested loop
→ loop inside another loop.

```
for i in range (1,4) / Outer loop runs 3 times
    for j in range (1,4) / inner loop runs 3 times for each Outer loop.
        print (i,j)
```

for loop with string
→ loop through each character in a string
→ analyzing text char by char
- Counting vowels.

range() with for loop → range() function generates a sequence of numbers.

Syntax: range (start, stop, step)
↓
how much to increase

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

O/P: 1 2 3 4 5

Control Statements

→ Break - stops loop if a condition is met -

```
for i in range(1,6):
    if i==3:
        break
```

O/P: 1 2

→ Continue - Skip printing or logic when cond. is met.

```
for i in range (1,6)
    if i==3:
        continue
```

O/P: 1 2 4 5

→ pass - placeholder

```
for i in range(1,5)
    if i==3:
        pass
```

O/P: 1 2 3 4

→ else - Executes after the loop completes normally (without break)

```
for i in range(1,4):
    print(i)
else
    print("loop finished")
```

O/P 1 2 3 loop finished.

→ enumerate - gives both index and value when looping.

```
text = "python"
for index, char in enumerate(text):
print(f"Index {index} : {char}")
```

O/P : Index 0 : p
Index 1 : y
Index 2 : t