#lec 6

Rising codeers Era

Loops
For loop

- instead of wrighting code again & again we can use the concepts of loops
- to perform specific task repeatedly in our progeam.
- reduces lines of code?

⊙ For Loops ?

"To iterate (repeat a block of code) over a sequence of elements, such as lists, strings, tuples, or ranges of numbers."

Syntax

```
for item in sequence:
    # Code to execute for each item
```

for :→ Keyword indicating a for loop.

item :→ variable that takes on the value of each item in sequence during each iteration

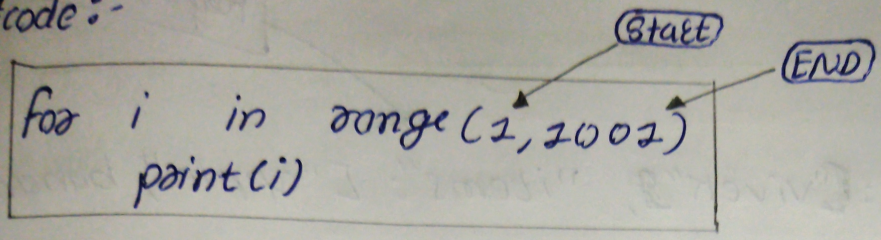
in :→ Keyword linking the item to the sequence being iterated over.

Sequence :→ iterable objects (list, string, tuple, range, etc)

Problem Statement 1

print numbers from 1 to 1000.

#code:-



Problem Statement 2

using for loop iterate over a list of names

#code:-

```
names = ["Omkar", "Sushant", "Durgesh"]
for name in names:
    print("Hello, " + name + "!")
```

explan:-

→ Creating a list named "names" using square brackets

→ List contain 3 names.

→ For loop iterates over each name in the names list

→ During each iteration, the current name is assigned to the variable "name".

Problem Statement 3

Processing customer order

#code:-

Key-value pairs

```
orders = [
    {"name": "vivek", "items": ["apple", banana"]},
    {"name": "Sahil", "items": ["orange", "grape"]},
]
for order in orders:
    print("Processing order for: ", order["name"])
    print("Items", order["items"])
```

expla":-

→ Creating a list named orders.

→ list contains 2 dictionaries
- → vivek → apple, banana.
- → Sahil → orange, grape.

→ for loop iterates over each order in the oeders list

→ during each iteration, the current order (which is a dictionary) is assigned to variable "oeder".

Problem Statement 4

checking for available usernames

code:-

```
usernames = ["Deepak", "Milind", "Sanket", "Aniket"]
desired_username = input("Enter user name: ")

for username in usernames:
    if username == desired_username:
        print("Username is already taken.")
        # Exit a loop when match is found
        break
# Execute only if the loop completes without a
# break
else:
    print("Username is not available!")
```

→ Creating a list named usernames

→ taking user input. using input() & store in variable desired_username.

→ checking for availability using forloop

→ comparing username with desired_username

→ If match ⇒ "Username is already taken!" & breaks out the loop using break keyword.

→ No match ⇒ found after all iterations, else block executes.

→ else block executes only when the loop completes without finding a match.

points ⇒ ("Usename is not available!")

Problem statement 5.

Create a multiplication table

#code

```
# taking i/p from user
table = int(input("Enter no: "))
for i in range(1,12):
    print(table, "*", i, "=", table*i)
```