

Assignment - III

Iteration in Python such as state, while, for, break, continue and pass with examples.

→ In Python, you can use various iteration and control flow constructs like 'while', 'for', 'break', 'continue' and 'pass' to control the flow of your program.

Iteration (or) Looping Statement:-

→ An Iterative statement allows us to execute a statement multiple times.

→ Repeated execution of a set of statement is called iteration (or) looping.

Types of Iterative statements:-

→ While loop

→ For loop

→ Nested loop

While :-

→ A while loop executes a block of statement again and again until the condition gets false.

→ The while keyword is followed by test expression and a colon.

→ Following the header is an indented body.

Syntax:- `while expression:
 true statement.`

→ The "while" loop continues to execute a block of code as long as, a given condition is "True".

Example of while using

```
n = int(input("Enter the number:"))
```

```
i = 1
```

```
fact = 1
```

```
while (i <= n):
```

```
    fact = fact * i
```

```
i = i + 1
```

```
print("The factorial is", fact)
```

→ Using the while will describe that if i is less than or equal to n.

→ Print statement will give the output.

INPUT = 5

The factorial is = 120

```
count = 0
```

```
while count < 3
```

```
printf("count is %d\n")
```

```
count += 1
```

→ The output will be

```
1  
2  
3
```

Using For loop (or) Statement

- The For loop is used to iterate a sequence of elements (list, tuple, string) for a specified number of times.
- For loop in Python starts with the keyword "for" followed by an arbitrary variable name, which hold its value in the following sequence object syntax:
**FOR iterating-variable in sequence:
statement.**

Example Using For loop

```
fruits = ["OPPIE", "banana", "cherry"]  
for fruit in fruits:  
    print(f"current fruit: {fruit}")
```

- A sequence represent a list or a string.
- The iterating variable takes the first item in sequence.

OUTPUT:-

current fruit: OPPIE

current fruit: banana

current fruit: cherry.

Nested Loop:

→ Python Programming allows using one loop inside another loop.

→ For example using a while loop or a for loop inside of another while for loop.

Syntax:

for iterating_variable sequence:

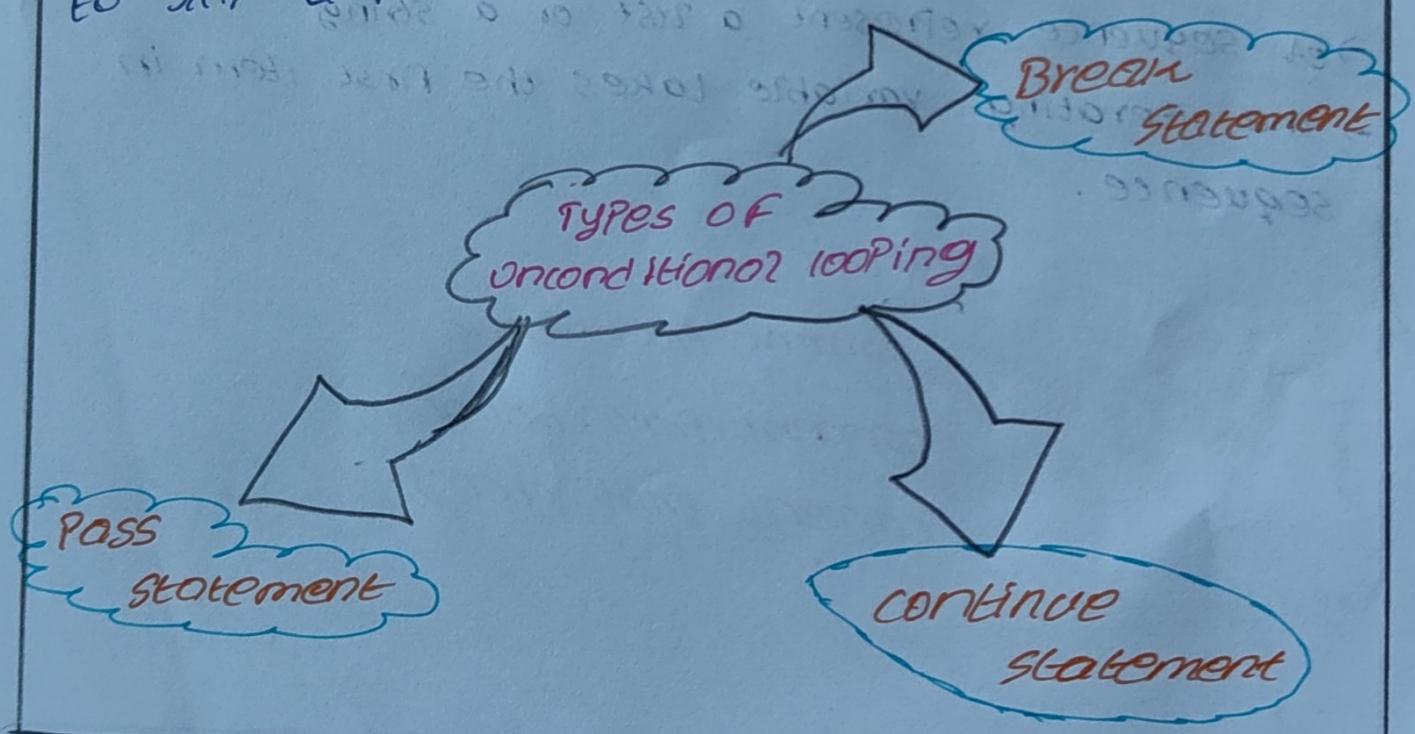
for iterating_variable sequence:

innerloop statement.

Outer loop statement.

Unconditional Statement:

→ A situation in which need to exit a loop completely when an external condition is triggered (or) need to skip a part of the loop.

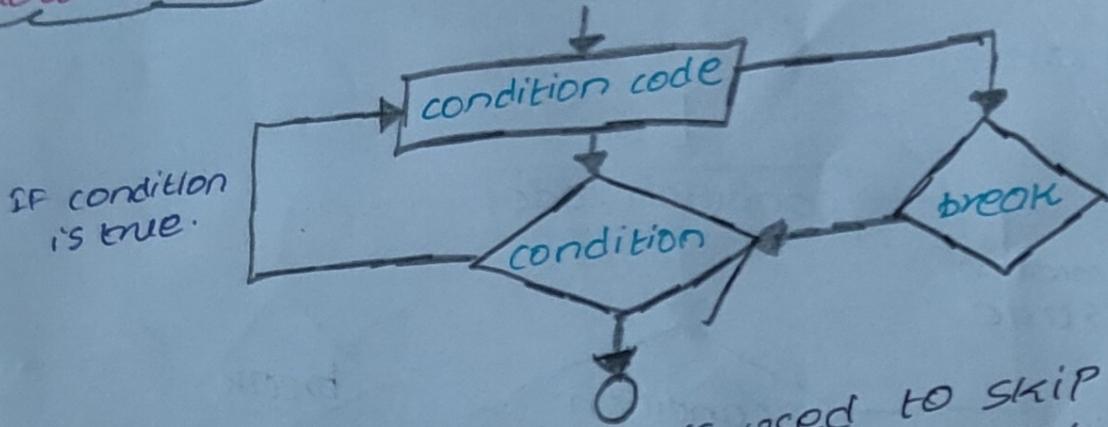


Continue Statement:-

- the continue statement returns the control to the beginning of the loop statement.
- the continue statement rejects all remaining statements and move back to top of the loop.

Syntax: continue

Flow-chart:



- The 'continue' statement is used to skip the current iteration of a loop and move next one.

Example:-

```
for num in range(5):
```

```
    if num == 2:
```

```
        continue
```

```
    print(num)
```

- continue statement will move forward if num is equal to 2

OUTPUT:- 0, 1, 3, 4.

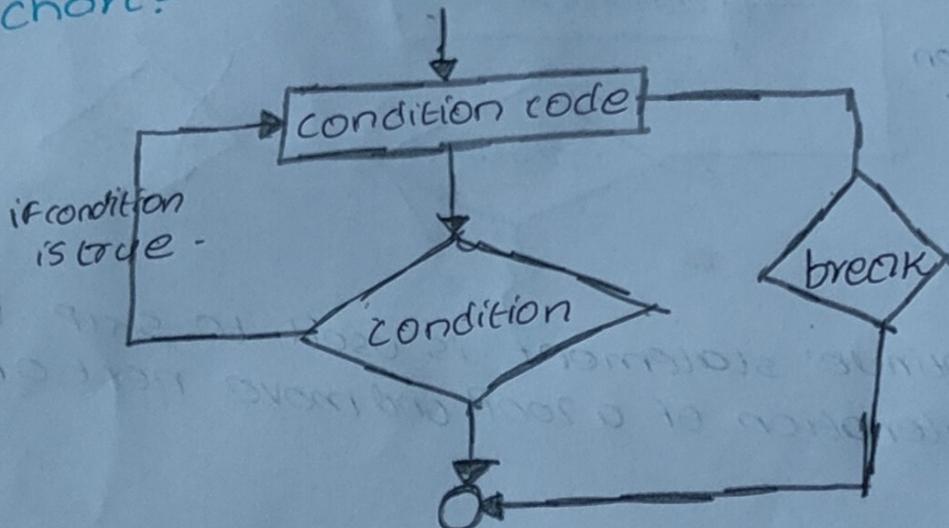
- print will get the output and print in compiler.

Break statement

- A break statement terminates the current loop and transfers the execution of statement immediately following the loop.
- The break statement is used when some external condition is triggered.

Syntax: break.

Flowchart:



- The "break" statement is used to exit a loop prematurely when a certain condition is met.

Example:-

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

- The break point will function the and if loop is one it continue the break function

Output: 0, 1, 2, 3, 4.

PASS statement:-

→ The Pass statement is a null operation, and nothing happens when it's executed.

→ It can be used when a statement is required syntactically but the program requires no action.

Syntax:- PASS

→ The "PASS" statement is a placeholder that does nothing.

→ It's often used when a statement is syntactically required but you don't want to execute any code.

Example:-

for letter in "Python"
 if letter == 'h'
 PASS
 print(letter)
 print("bye")

→ In this example the 'PASS' statement is used to indicate that you plan to add code inside the loop later, but for now, it does nothing.

OUTPUT IS = PYTHON
 bye.

- These iteration and control flow constructs are fundamental to Python and allow you to create flexible and powerful programs.
- You can use them in various combinations to control the flow of your code based on different and requirement
