PYTHON

Control Flow : Loop control flow statements are used to change the execution of the sequence.

- Loop iterates over a block of code until the condition is False. But, when we want to stop the present running iteration or complete loop without checking the condition we use control flow statements.

Python provide three control flow statements are:

- Break
- Continue
- Pass

Break: This statement is used to terminate the present running loop completely and go to next statements after the body of loop.

- If the break statement is written in nested loop then the inner loop will be terminated

. The syntax for break statement will be simply write **break** in body of loop where to stop iterating.

Syntax:

```
break
```

Example:

```
for i in [12, 16, 17, 24, 29]:
    if i % 2 == 1:
        break
    print(i)
print("done")
```

- In the example it first iterates through for loop and take 12 as I value and goes to if condition and check whether it is satisfied or not. If yes goes to break.

- Here if condition is false so it goes to next statement and print the value.

- When the value is 17 if condition becomes True and break statement will be executed and controller goes to next statement i.e, “done”.

Continue : This statement is used to skip the present running iteration only.

- Once a continue statement is executed it skip running iteration and go back to same loop for next iteration.

- Syntax for continue is writing **continue** in the body of loop.

Syntax :

```
Continue
```

Example :

```
for i in range(1,21):
    if i % 5 ==0:
        continue
    print(i, ”not divisible by 5)
```

- In the example when it reaches the continue statement it go back to **for loop** for next iteration and print the value from 1 to 20 which are not divisible by 5 as we specified the if condition as i%5 == 0.

Pass : This statement is used for empty control statements.

• When we use a pass it does not do any action. So it is called as null statement in python.

Syntax is same as other control flow as **pass**.

Syntax :

```
Pass
```

Example :

```
for letter in 'Python':
    if letter == 'h':
        pass
        print ('This is pass block')
    print ('Current Letter :', letter)
print ("Good bye!")
```

• In the example we specified the condition as letter == “h”. Now it iterates through the sequence once h is reached it goes inside if statement.

• As we used pass it doesn’t do any action and goes to print statement in side if statement is any.