flow_controls

February 16, 2023

1 Flow Control Statements of Python

1. Conditional Statements

```
Syntax 1:

if condition:
    action
else:
    action
------------------------------------------
Syntax 2:
    if condition:
        action

    elif condition:
        action
    else:
        action


------------------------------------------
Syntax 3:

    if condition:
        action
```

```python
# Example: WAP to take string as input and compare it with 'Durga'. Are equal
 or not

s=input("Enter your name: ")

if s.lower()=='durga':
    print("Matched, What can I do for you Maalik??")
else:
    print("Not Matched, Get lost right now")
```

```
Enter your name:  durga

Matched, What can I do for you Maalik??
```

```
[6]: # Example: WAP to take 3 inputs and find which is maximum

     a,b,c=[int(x) for x in input("Enter 3 numbers: ").split(',')]

     if a>b and a>c:
         print("%i is Maximum" %a)
     elif b>a and b>c:
         print("%i is Maximum" %b)
     else:
         print("%i is Maximum" %c)
```

Enter 3 numbers: 10,20,30

30 is Maximum

[ ]:

2. Iterative Statements

Those statements executes a block of code for specified times or based on some condition. It is know as iterative statements. 1. For loop 2. While loop

```
[ ]: Syntax:

     # for loop

         for x in variable:
             action statements

     # while loop

     while condition:
         action statements
```

```
[3]: # for loop example
     # sum of 1 to 10 numbers
     n=10
     sum=0
     for i in range(1, n+1):
         sum=sum+i

     print(sum)
```

55

```
[4]: # sum of 1 to 10 using while loop

     n=10
     s=0
```

```
while n>0:
    s=s+n
    n-=1

print("The sum: ", s)
```

The sum: 55

[ ]:

3. Transfer Statements

These are the statements which are used to redirect or discontinue the iteration of loops 1. break
2. continue 3. pass

break : It stopped the execution of loop.

[5]:
```
# example:
a=[10,200,30,500,300,250]

for i in a:
    if i>=500:
        print("loop Stopped")
        break
    print("item {}".format(i))
```

item 10
item 200
item 30
loop Stopped

continue: It return the flow of execution back to the starting point of loop

[6]:
```
# example

a=[10,200,30,500,300,250,800,3000,200]

for i in a:
    if i>=500:
        print("item skipped")
        continue
    print("item {}".format(i))
```

item 10
item 200
item 30
item skipped
item 300
item 250

```
item skipped
item skipped
item 200
```

pass: it used when we don't want to give any action statement. It is like a none type

```
[7]: a=[10,200,30,500,3000,250]

for i in a:
    if i>=500:
        pass
    else:
        print("item {}".format(i))
```

```
item 10
item 200
item 30
item 250
```

[ ]: