

Conditional And Iterative Statements

सीबीएसई पाठ्यक्रम पर आधारित
कक्षा -11

अध्याय -4



द्वारा:

संजीव भदौरिया

स्नातकोत्तर शिक्षक (संगणक विज्ञान)

के० वि० बाराबंकी (लखनऊ संभाग)

परिचय

- सामान्यतया program का execution शुरू से अंत तक होता है।
- लेकिन कई programs में सीधे क्रम में क्रियान्वयन (execution) नहीं होता है ।
- आवश्यकता के अनुसार program में शाखाएं निकलती हैं अथवा एक निश्चित बिंदु पर बार बार वापस आकर program आगे बढ़ता है ।
- ऐसे कार्यों के लिए पाइथन कुछ statements के रूप में tool प्रदान करता है ।
- जैसे if-else, for, while statements तथा jump statement जैसे break, continue इत्यादि ।

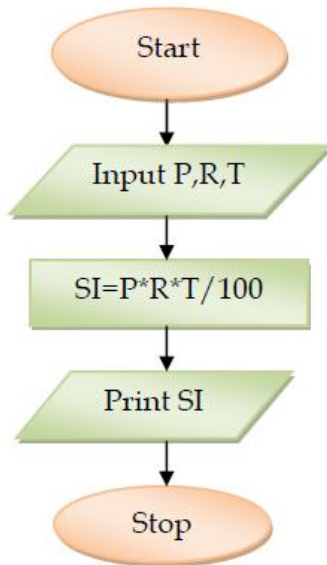
पाइथन में statements के प्रकार

- पाइथन में statement तीन प्रकार के होते हैं -
 - » Empty Statements
 - pass
 - » Simple Statements (Single Statement)
 - name=input (“Enter your Name “)
 - print(name) इत्यादि |
 - » Compound Statements
 - `<Compound Statement Header>`:
 - `<Indented Body containing multiple simple statements/compound statements>`
 - यहाँ Header लाइन keyword से शुरू होकर कोलन (:) पर समाप्त होती है |
 - एक body में एक से अधिक साधारण पाइथन statements या compound statements होते हैं |

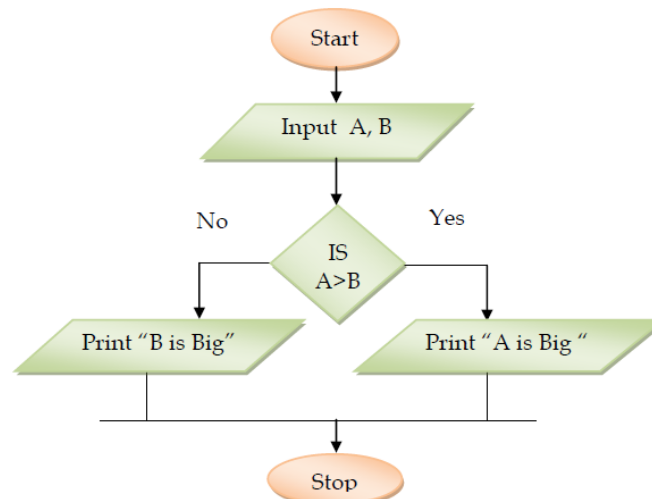
Statement Flow Control

- किसी भी program में statements या तो क्रम(sequential) से क्रियान्वित(execute) होंगे या चयन करके(Selective) या फिर पुनरावृत्ति(Iteration) में क्रियान्वित होंगे ।

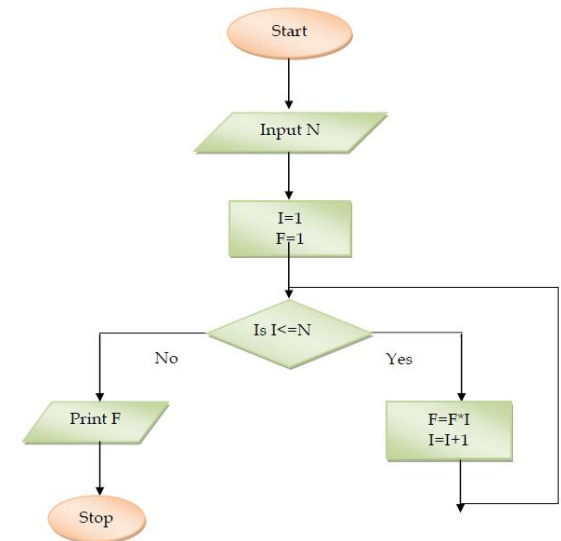
Sequential



Selective



Iterative



Program Logic Development Tool

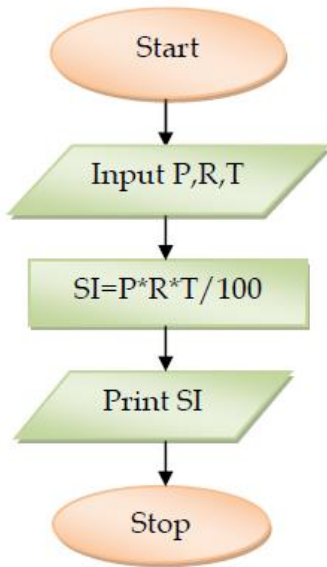
- प्रोग्रामिंग में किसी भी समस्या के समाधान के लिए निम्न चरण आवश्यक है –
 1. समस्या का निर्धारण या परिभाषण
 2. समस्या का विश्लेषण (analysis)
 3. Design अथवा प्रारूप बनाना (Flowchart या Algorithm)
 4. Code लिखना (प्रोग्रामिंग)
 5. Testing और Debugging
 6. कार्यान्वयन (Implementation)
 7. रख-रखाव (Maintenance)

Algorithm

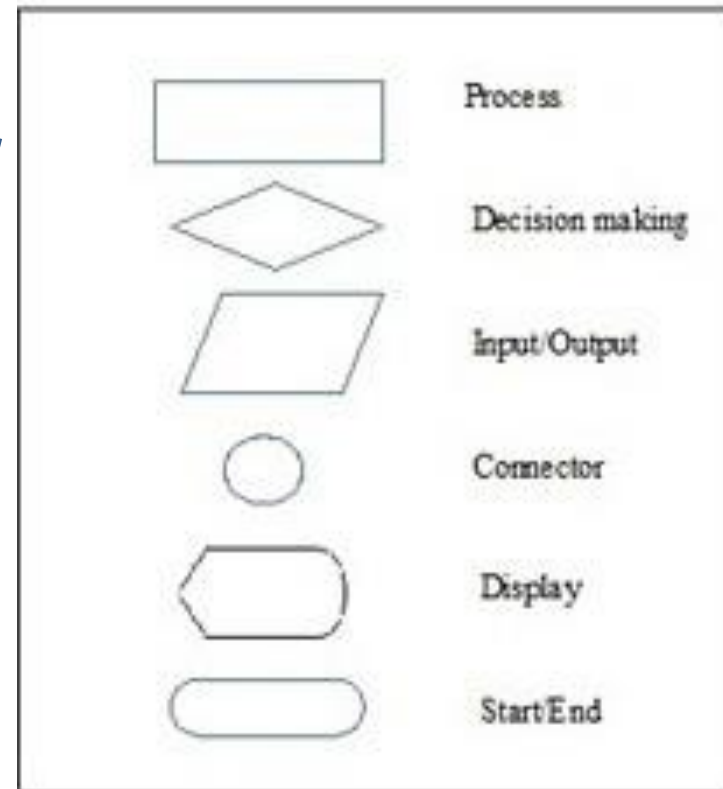
- Algorithm किसी भी program को लिखने का एक क्रमबद्ध procedure होता है | जैसे –
- दो संख्याओं को जोड़ने का algorithm निम्नवत है -
 1. *Input First Number*
 2. *Input Second Number*
 3. *Add First Number with Second Number and store into Third number.*
 4. *Display Third number*

Flowcharts

- Flowchart किसी भी program को लिखने का एक क्रमबद्ध सांकेतिक चित्रण के साथ प्रदर्शन होता है | जैसे –
- साधारण व्याज निकालने का flowchart निम्नवत है -



Flowchart में प्रयुक्त होने वाले संकेत चित्र |

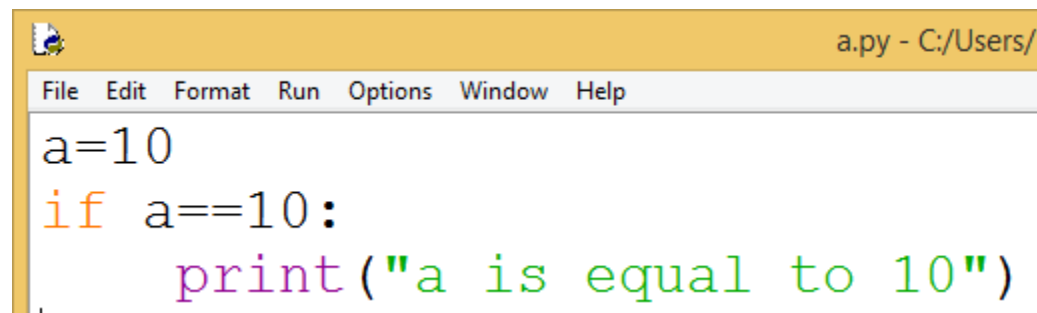


Python में if Statements

- Python में if statement का प्रयोग चयनित प्रोसेसिंग के लिए करते हैं | यदि किन्हीं दो statements में से कोई एक, किसी शर्त पर चयन करना हो तो if statement का प्रयोग किया जाता है | इसका syntax (प्रारूप) निम्नवत है -

```
if <condition>:  
    statement(s)
```

जैसे -

A screenshot of a Python IDE window titled 'a.py - C:/Users/'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

```
a=10  
if a==10:  
    print("a is equal to 10")
```


Python में if-else Statements

- Python में if-else statement का प्रयोग चयनित प्रोसेसिंग के लिए करते हैं | यदि किन्हीं दो statements में से कोई एक, किसी शर्त पर चयन करना हो तो if-else statement का प्रयोग किया जाता है | इसका syntax (प्रारूप) निम्नवत है -

if <condition>:

statement(s) when condition is true

else:

statement(s) when condition is false

जैसे -

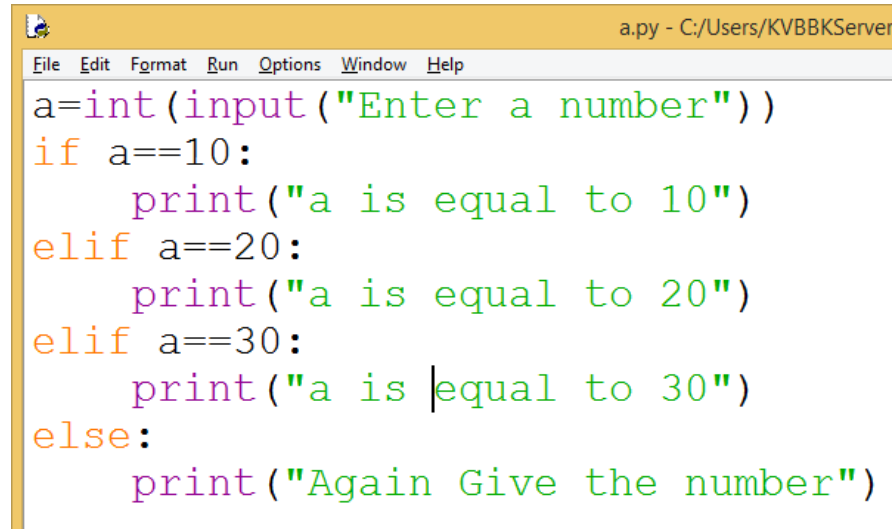
```
a.py - C:/Users/KVBBKSer
File Edit Format Run Options Window Help
a=10
if a==10:
    print("a is equal to 10")
else:
    print("a is not equal to 10")
```

Python में if-elif Statements

- Python में if-else statement का प्रयोग चयनित प्रोसेसिंग के लिए करते हैं | यदि किन्हीं दो से अधिक statements में से कोई एक, किसी शर्त पर चयन करना हो तो if-elif statement का प्रयोग किया जाता है | इसका syntax (प्रारूप) निम्नवत है -

```
if <condition1>:  
    statement(s) when condition1 is true  
elif <condition2>:  
    statement(s) when condition2 is true  
elif <condition3>:  
    statement(s) when condition3 is true  
else <condition4>
```

जैसे -



```
a.py - C:/Users/KVBBKServer  
File Edit Format Run Options Window Help  
a=int(input("Enter a number"))  
if a==10:  
    print("a is equal to 10")  
elif a==20:  
    print("a is equal to 20")  
elif a==30:  
    print("a is equal to 30")  
else:  
    print("Again Give the number")
```

Nested If -else

```
*a.py - C:/Users/KV
File Edit Format Run Options Window Help
a=int(input("Enter a number"))
b=int(input("Enter a number"))
c=int(input("Enter a number"))
if a>b:
    if a>c:
        print("a is greater")
    else:
        print("c is greater")
else:
    if b>c:
        print("b is graeter")
    else:
        print("c is greater")|
```

Loop/Repetitive Task/Iteration

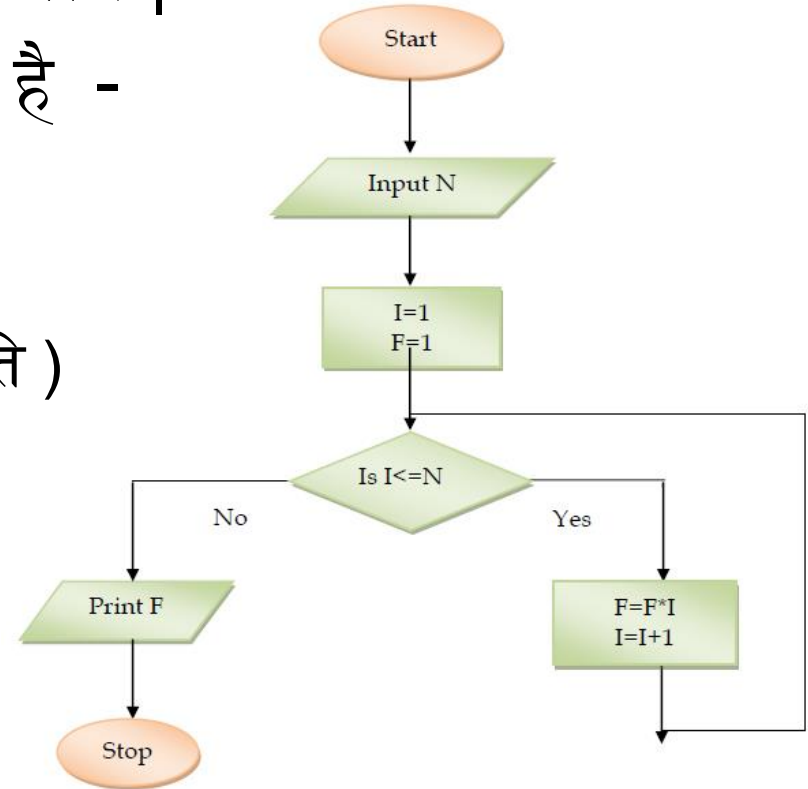
इस प्रकार के कार्यों में कुछ statements की तब तक पुनरावृत्ति होती है जब तक की सही स्थिति न आजाये ।

Loop में मुख्यतः 3 अवयव आवश्यक है -

1. Start (प्रारम्भिक मान देना)
2. Step (loop को आगे बढ़ने वाला)
3. Stop (loop को समाप्त करने की स्थिति)

पाइथन में loop निम्न हैं

- for loop (Counting based)
- while loop (Conditional Based)



range () Function

- **Python** में `range()` फंक्शन बहुत काम का है | इसके कार्य को समझने के लिए इसका `syntax` समझते हैं

`range (<lower limit>,<upper limit>)`

यदि हम लिखें - `range (0,5)`

तो यह एक list बनाता है `[0,1,2,3,4]` अर्थात यह list में lower limit से लेकर upper limit से 1 कम तक की सारे values को लेलेगा |

`range (0,10,2)` तो यह लिस्ट में `[0,2,4,6,8]` रखेगा |

`range (5,0,-1)` तो यह लिस्ट में `[5,4,3,2,1]` रखेगा |

`range (0,10,2)` तो यह लिस्ट में `[0,2,4,6,8]` रखेगा |

in तथा not in ऑपरेटर

- Python में *in* operator का निम्न कार्य है -
 - 3 in [1,2,3,4] में ये statement **True** return करेगा |
 - 5 in [1,2,3,4] में ये statement **False** return करेगा |
- Python में *not in* operator का निम्न कार्य है -
 - 5 not in [1,2,3,4] में ये statement **True** return करेगा

```
for a in [1, 2, 3]:  
    print(a)  
    print(a*a)
```

For loop के द्वारा किसी संख्या का पहाडा (Table)

For Loop का syntax

```
for <var> in <sequence>:  
    <statements to repeat>
```

```
num=int(input("Enter a number"))  
for a in range(1,11):  
    print(num, "x", a, "=", num*a)
```

Output

```
Enter a number10  
10 x 1 = 10  
10 x 2 = 20  
10 x 3 = 30  
10 x 4 = 40  
10 x 5 = 50  
10 x 6 = 60  
10 x 7 = 70  
10 x 8 = 80  
10 x 9 = 90  
10 x 10 = 100
```

while loop के द्वारा किसी संख्या का पहाडा (Table)

While Loop का syntax

```
While <LogicalExpression>:  
    <loop body with increment  
    or decrement>
```

```
n=int(input("Enter a number"))  
c=1 → Start  
while c<11: → Stop  
    print(n, "x", c, "=", c*n)  
    c=c+1 → Step
```

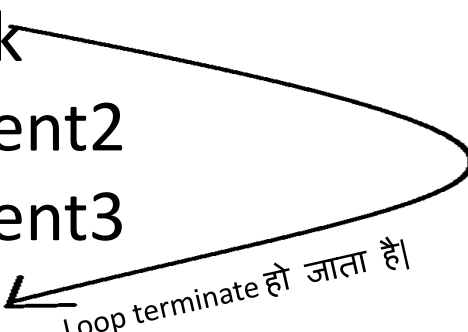
Output

```
Enter a number5  
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50
```


Jump Statements

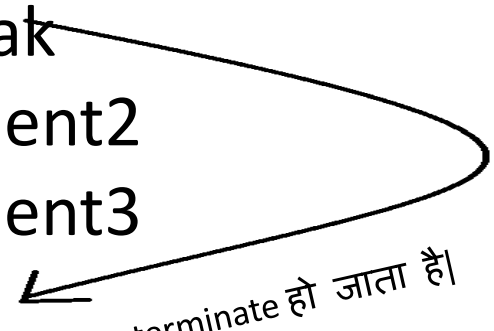
break Statement

```
while <test-condition>:  
    statement1  
    if <condition>:  
        break  
    statement2  
    statement3  
Statement4  
statement5
```



Loop terminate हो जाता है।

```
for <var> in <sequence>:  
    statement1  
    if <condition>:  
        break  
    statement2  
    statement3  
Statement4  
statement5
```



Loop terminate हो जाता है।

Jump Statements

break Statement

```
n=int(input("Enter a number"))
c=1
while c<11:
    if c==5:
        break
    print(n, "x", c, "=", c*n)
    c=c+1
```

Output

```
Enter a number4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
```

```
n=int(input("Enter a number"))
c=1
for c in range(1,11):
    if c==5:
        break
    print(n, "x", c, "=", c*n)
|
```

Output

```
Enter a number5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
>>>
```

Jump Statements

continue Statement

```
while <test-condition>:  
    statement1  
    if <condition>:  
        continue  
    statement2  
    statement3
```

```
Statement4  
statement5
```

```
for <var> in <sequence>:  
    statement1  
    if <condition>:  
        continue  
    statement2  
    statement3
```

```
Statement4  
statement5
```

Loop continue हो जाता है।

Loop continue हो जाता है।

Jump Statements

continue Statement

```
n=int(input("Enter a number"))  
for c in range(1,11):  
    if c==5:  
        continue  
    print(n, "x", c, "=", c*n)
```

```
n=int(input("Enter a number"))  
c=0  
while c<11:  
    c=c+1  
    if c==5:  
        continue  
    print(n, "x", c, "=", c*n)
```

दोनों program का output

```
Enter a number5  
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50  
5 x 11 = 55  
>>> |
```

संजीव भदौरिया, के० वि० बाराबंकी

Nested Loop

```
n=int(input("Enter the number"))  
for r in range(1,n+1):  
    for c in range(1,r+1):  
        print("*", end="")  
    print("")
```

OUTPUT

```
Enter the number5  
*  
**  
***  
****  
*****
```

Assignments

1. WAP to find greatest among three numbers.
2. WAP to print the result on the basis of marks entered of a student.
3. WAP to print counting up to n.
4. WAP to print even numbers up to n.
5. WAP to print odd numbers up to n.
6. WAP to print Fibonacci series.
7. WAP to calculate x^n .
8. WAP to calculate $n!$.
9. WAP to print different patterns.

धन्यवाद

- अधिक पाठ्य-सामग्री के लिए निम्न लिंक पर क्लिक करें –
- www.pythontrends.wordpress.com