

Control Statements

Control statements are used to control the flow of execution of program.

These control statements are classified into 3 categories

1. Conditional Control Statements
 - a. If
 - b. match
2. Looping Control Statements
 - a. While
 - b. for
3. Branching Statement
 - a. Break
 - b. Continue
 - c. Return
 - d. Pass (Not a branching statement)

Conditional Control Statements

Conditional control statements are used to execute block of statements based on condition or selection.

These conditional control statements are two types

1. If
2. Match

If statement

if statement is used to execute block of statements based on condition.

1. Simple if
2. If..else
3. If..elif..else (if..else ladder)
4. Nested if

Simple if

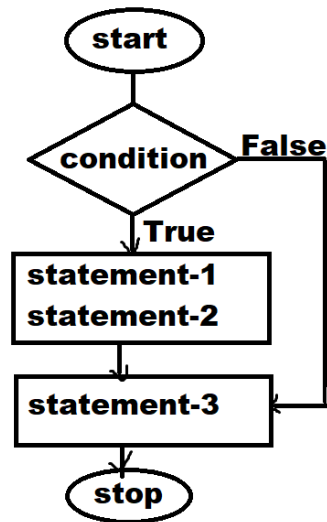
“if” without else is called simple if

Syntax:

```
if <condition>:  
    statement-1  
    statement-2  
statement-3
```

if condition is True, PVM executes statement-1, statement-2 and statement-3

if condition is False, PVM executes statement-3



“pass” keyword

Empty blocks are not allowed in python.

A block must have one statement.

“pass” is keyword used for representing empty blocks.

“pass” keyword do nothing (OR) it acts like a place holder.

Example:

```
if 10>5:  
    print("Hello")  
  
print("Bye")  
  
if 10>20:  
    print("Java")  
  
print("Python")  
  
if 10>5:  
    pass
```

Output

```
Hello  
Bye  
Python  
.Net  
Oracle
```

print("Oracle")	
-----------------	--

Note: all the statements inside the block must be at same indentation level.

Example: <pre># Banking Application (withdraw) accno=int(input("Account No :")) balance=float(input("Balance :")) tamt=float(input("Amount to Withdraw :")) if tamt>=50000: print("Input PAN Details ") if tamt<balance: balance=balance-tamt print("AccountNo ",accno) print("Balance ",balance)</pre>	Output Account No :1 Balance :90000 Amount to Withdraw :50000 Input PAN Details AccountNo 1 Balance 40000.0
---	--

If..else

This syntax define 2 blocks

1. if block
2. else block

Syntax: <pre>If <condition>: Statement-1 Statement-2 else: Statement-3 Statement-4 Statement-5</pre>	<pre>graph TD Start([start]) --> Condition{condition} Condition -- True --> Statements1[statement-1
statement-2] Condition -- False --> Statements2[statement-3
statement-4] Statements1 --> Statement5[Statement-5] Statements2 --> Statement5 Statement5 --> Stop([stop])</pre>
---	---

<p>If condition is True, PVM executes Statement-1, Statement-2 and Statement-5</p> <p>If condition is False, PVM executes statement-3, statement-4 and Statement-5</p>	
--	--

<p>Example:</p> <p># Write a program to check whether a person is eligible to voting or not</p> <pre>age=int(input("Enter Age of the Person ")) if age>=18: print("Eligible to Vote") else: print("Not Eligible Vote")</pre>	<p>Enter Age of the Person 20 Eligible to Vote</p> <p>Enter Age of the Person 15 Not Eligible Vote</p>
<p># Write a program to check whether a input number is even or odd</p> <pre>num=int(input("Enter Number ")) if num%2==0: print(num,"is even") else: print(num,"is odd")</pre>	<p>Enter Number 6 6 is even</p> <p>Enter Number 9 9 is odd</p>
<p># Write a program to check whether a input number is divisible by 7</p> <pre>num=int(input("Enter any number</pre>	<p>Enter any number 21 21 is divisible with 7</p> <p>Enter any number 20 20 is not divisible with 7</p>

<pre> ")) if num%7==0: print(num,"is divisible with 7") else: print(num,"is not divisible with 7") </pre>	
<p>Example:</p> <p># Write a program to check whether a input year is leap or not</p> <pre> year=int(input("Enter Year ")) if (year%4==0 and year%100!=0) or (year%400==0): print(year,"leap") else: print(year,"not leap") </pre>	<p>Output</p> <p>Enter Year 2000 2000 leap</p> <p>Enter Year 2001 2001 not leap</p> <p>Enter Year 2004 2004 leap</p>