

CONDITIONAL AND ITERATIVE STATEMENTS

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Learning Outcomes

- Types of statements in Python
- Statement of Flow Control
- Program Logic Development Tools
- if statement of Python
- Repetition of Task - A necessity
- The range() function
- Iteration / Looping statement
- break and continue statement

Types of Statement in Python

- Statements are the instructions given to computer to perform any task. Task may be simple calculation, checking the condition or repeating action.
- Python supports 3 types of statement:
 - Empty statement
 - Simple statement
 - Compound statement

Empty Statement

- It is the simplest statement i.e. a statement which does nothing. It is written by using keyword – pass
- Whenever python encountered pass it does nothing and moves to next statement in flow of control
- Required where syntax of python required presence of a statement but where the logic of program does. More detail will be explored with loop.

Simple Statement

- Any single executable statement in Python is simple statement. For e.g.
- `Name = input("enter your name ")`
- `print(name)`

Compound Statement

- It represent group of statement executed as unit. The compound statement of python are written in a specific pattern:

Compound_Statement_Header :

indented_body containing multiple simple or compound statement

Compound Statement has:

- Header which begins with keyword/function and ends with colon(:)
- A body contains of one or more python statements each indented inside the header line. All statement in the body or under any header must be at the same level of indentation.

Statement Flow Control

- In python program statement may execute in a sequence, selectively or iteratively. Python programming support 3 Control Flow statements:

1. Sequence
2. Selection
3. Iteration

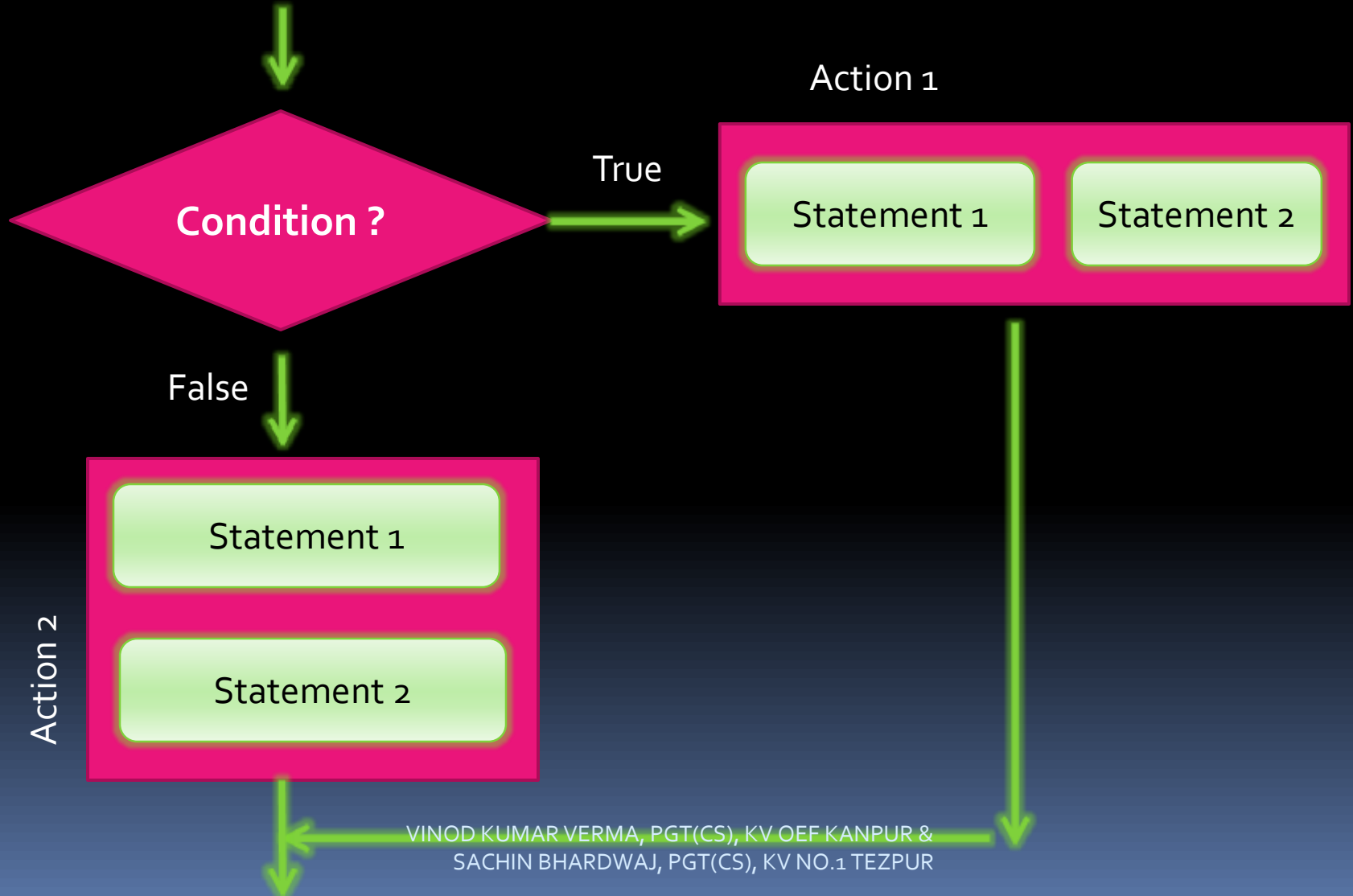
Sequence

- It means python statements are executed one after another i.e. from the first statement to last statement without any jump.

Selection

- It means execution of statement will depend upon the condition. If the condition is true then it will execute Action 1 otherwise Action 2. In Python we use **if** to perform selection

Selection



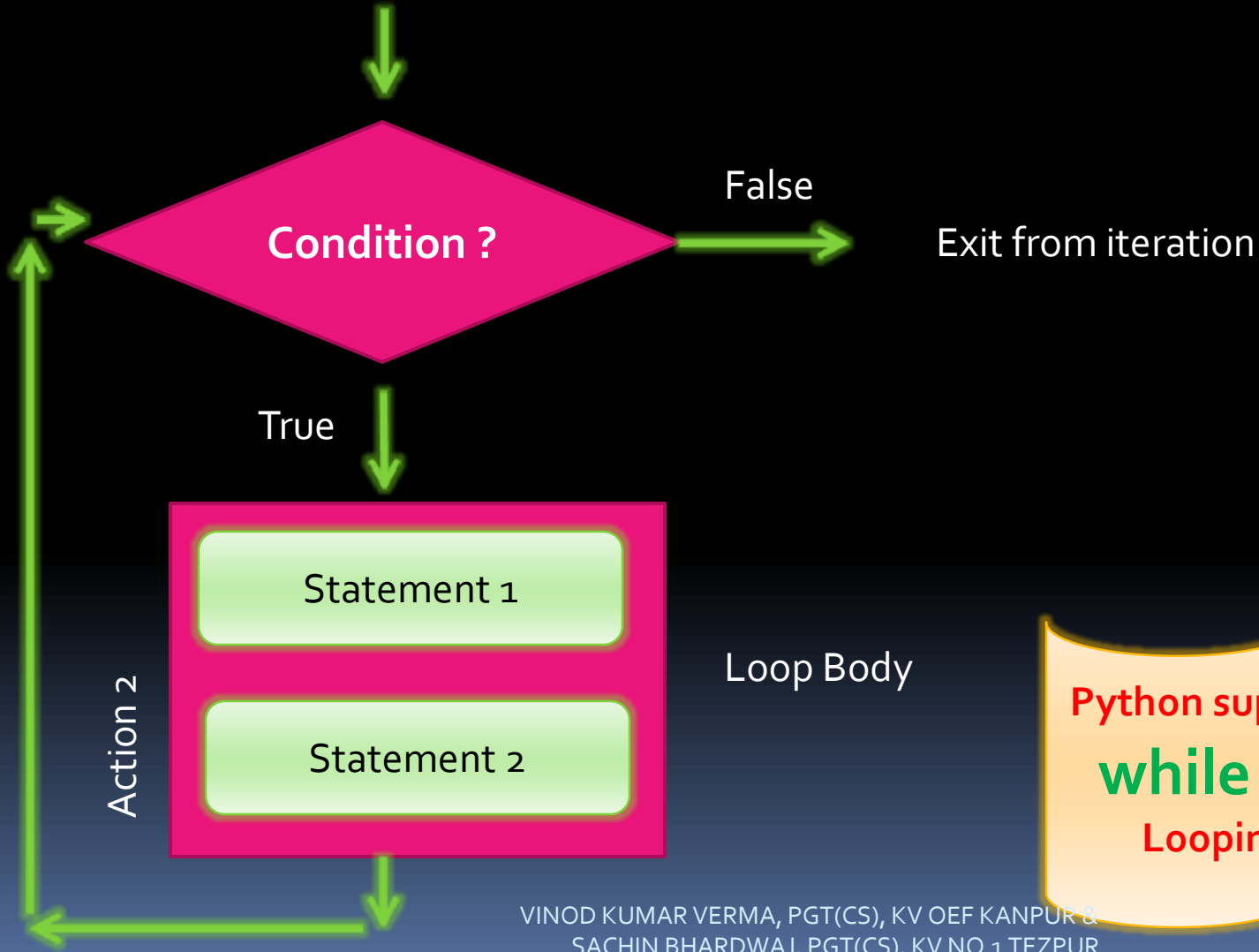
Selection

- We apply Decision making or selection in our real life also like if age is 18 or above person can vote otherwise not. If pass in every subject is final exam student is eligible for promotion.
- You can think of more real life examples.

Iteration - Looping

- Iteration means repeating of statement depending upon the condition. Till the time a condition is **True** statements are repeated again and again. Iteration construct is also known as **Looping construct**
- In real world we are using Looping construct like learning table of any number we multiply same number from 1 to 10, Washing clothes using washing machine till the time is over, our school hours starts from assembly to last period, etc.

Iteration - Looping



Python supports **for** and **while** statement for Looping construct

Error and Exceptions

- Error means 'bugs' in a program, i.e. either program is not running or running but not producing the expected output.
- Exceptions are those errors which appears during the execution time of program i.e. there are some exceptional condition in which program is not successfully executing.
- Types of Error:
 - Syntax Error
 - Semantic Error
 - Logical Error
 - Runtime Error

Syntax Error

- It occurs before the execution of program.
- It occurs due to violation of programming language rule for e.g. missing parenthesis, incorrect use of operators etc.

```
n = int(input("Enter any number "))  
r = n * n * n  
print("Cube of number ",r)
```

Syntax error due to
missing parenthesis

Semantic Error

- occurs when statements are not meaningful. For e.g. 'Sita Plays Piono' is syntactically and semantically correct but 'Piono plays Sita' is syntactically correct but semantically incorrect. Take another exampe : $X*Y=Z$ is semantically incorrect because expression cant comes before the assignment operator;

Logical Error

- occurs when program is executing successfully but not producing the correct output. It is one of the most difficult error to trace and debug. It may occur by use of wrong operators like use of '*' in place of '+' Or writing $C=B/A$ in place of $C=A/B$ etc.

```
# Example to divide the  
# first number with second number  
n1 = int(input("Enter first number "))  
n2 = int(input("Enter second number "))  
result = n2 / n1  
print(result)
```

```
Enter first number 20  
Enter second number 10  
0.5
```

With input 20 as n1 and 10 as n2, expected result was 2 but output is 0.5, because expression is $n2/n1$ not $n1/n2$

Runtime Error

- It occurs during the execution of program.
- It is also known as exception.
- It occurs due to some wrong operation, input, etc. the most common runtime error is "divide by zero"

```
# Example to divide the
# first number with second number
n1 = int(input("Enter first number "))
n2 = int(input("Enter second number "))
result = n1 / n2
print(result)
```

```
Enter first number 20
Enter second number 0
Traceback (most recent call last):
  File "C:/Users/vin/AppData/Local/Programs/Python
5, in <module>
    result = n1 / n2
ZeroDivisionError: division by zero
```

Exception handling (try & except)

- **try** : this block is used for writing those statements in which exception may occurs.
- **except** : this block is used for handling the exception occurs in try block i.e. decides what to do with exception.

```
try:
    n1 = int(input("Enter first number "))
    n2 = int(input("Enter second number "))
    result = n1 / n2
    print(result)
except:
    print("Cannot divide by zero")
```

```
Enter first number 20
Enter second number 10
2.0
>>>
= RESTART: C:/Users/vin/A
Enter first number 20
Enter second number 0
Cannot divide by zero
```

if Statement of Python

- 'if' statement of python is used to execute statements based on condition. It tests the condition and if the condition is true it perform certain action, we can also provide action for false situation.
- if statement in Python is of many forms:
 - if without false statement
 - if with else
 - if with elif
 - Nested if

Simple “if”

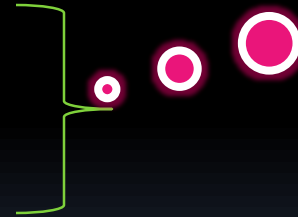
- In the simplest form if statement in Python checks the condition and execute the statement if the condition is true and do nothing if the condition is false.

- **Syntax:**

if condition:

Statement₁

Statements



All statement belonging to if must have same indentation level

**** if statement is compound statement having header and a body containing intended statement.**

Points to remember with “if”

- It must contain valid condition which evaluates to either True or False
- Condition must followed by Colon (:), it is mandatory
- Statement inside if must be at same indentation level.

Example - 1

Input monthly sale of employee and give bonus of 10% if sale is more than 50000 otherwise bonus will be 0

```
bonus = 0
```

```
sale = int(input("Enter Monthly Sales :"))
```

```
if sale > 50000:
```

```
    bonus = sale * 10 / 100
```

```
print("Bonus = " + str(bonus))
```

To Do..

1. WAP to enter 2 number and print the largest number
2. WAP to enter 3 number and print the largest number

To Do - 1

1. WAP to enter 2 number and print the largest number

```
n1 = int(input("Enter first number "))
n2 = int(input("Enter second number "))
large = n1
if (large<n2):
    large=n2
print("Largest number is ", large)
```

To Do - 2

2. WAP to enter 3 number and print the largest number

```
n1 = int(input("Enter first number "))
n2 = int(input("Enter second number "))
n3 = int(input("Enter third number "))
large = n1
if (large<n2):
    large=n2
if(large<n3):
    large=n3
print("Largest number is ", large)
```

if with else

- **if with else** is used to test the condition and if the condition is **True it perform certain action** and **alternate course of action** if the **condition is false**.
- Syntax:

if condition:

Statements

else:

Statements

Example - 2

Input Age of person and print whether the person is eligible for voting or not

```
age = int(input("Enter your age "))  
if age >= 18:  
    print("Congratulation! you are eligible for voting ")  
else:  
    print("Sorry! You are not eligible for voting")
```

To Do...

1. **WAP to enter any number and check it is even or odd**
2. WAP to enter any age and check it is teenager or not
3. **WAP to enter monthly sale of Salesman and give him commission i.e. if the monthly sale is more than 500000 then commision will be 10% of monthly sale otherwise 5%**
4. WAP to input any year and check it is Leap Year or Not
5. **WAP to Input any number and print Absolute value of that number**
6. WAP to input any number and check it is positive or negative number

if with elif

- **if with elif is used where multiple chain of condition is to be checked. Each elif must be followed by condition: and then statement for it. After every elif we can give else which will be executed if all the condition evaluates to false**

- Syntax:

if condition:

Statements

elif condition:

Statements

elif condition:

Statements

else:

Statement

Example - 3

Input temperature of water and print its physical state

```
temp = int(input("Enter temperature of water "))
if temp>100:
    print("Gaseous State")
elif temp<0:
    print("Solid State")
else:
    print("Liquid State")
```

Example - 4

Input 3 side of triangle and print the type of triangle –
Equilateral, Scalene, Isosceles

```
s1 = int(input("Enter side 1"))
s2 = int(input("Enter side 2"))
s3 = int(input("Enter side 3"))
if s1==s2==s3:
    print("Equilateral")
elif s1!=s2!=s3!=s1:
    print("Scalene")
else:
    print("Isosceles")
```


Program to calculate and print roots of a quadratic equation: $ax^2+bx+c=0(a\neq 0)$

```
import math
print("For quadratic equation,  $ax^{**2} + bx + c = 0$ ,  
enter coefficients ")
a = int(input("enter a"))
b = int(input("enter b"))
c = int(input("enter c"))
if a==0:
    print("Value of ",a," should not be zero")
    print("Aborting!!!")
else:
    delta = b*b - 4 * a * c
```

Program to calculate and print roots of a quadratic equation: $ax^2+bx+c=0(a\neq 0)$

```
if delta > 0:
```

```
    root1=(-b+math.sqrt(delta))/(2*a)
```

```
    root2=(-b-math.sqrt(delta))/(2*a)
```

```
    print("Roots are Real and Unequal")
```

```
    print("Root1=",root1,"Root2=",root2)
```

```
elif delta==0:
```

```
    root1=-b/(2*a)
```

```
    print("Roots are Real and Equal")
```

```
    print("Root1=",root1,"Root2=",root1)
```

```
else:
```

```
    print("Roots are Complex and  
Imaginary")
```

To Do...

1. WAP to enter marks of 5 subject and calculate total, percentage and also division.

<u>Percentage</u>	<u>Division</u>
>=60	First
>=45	Second
>=33	Third
otherwise	Failed

2. WAP to enter any number and check it is positive, negative or zero number

3. WAP to enter Total Bill amount and calculate discount as per given table and also calculate Net payable amount (total bill – discount)

<u>Total Bill</u>	<u>Discount</u>
>=20000	15% of Total Bill
>=15000	10% of Total Bill
>=10000	5% of Total Bill
otherwise	0 of Total Bill

To Do...

4. WAP to enter Bill amount and ask the user the payment mode and give the discount based on payment mode. Also display net payable amount

<u>Mode</u>	<u>Discount</u>
Credit Card	10% of bill amount
Debit Card	5% of bill amount
Net Banking	2% of bill amount
otherwise	0

5. WAP to enter two number and ask the operator (+ - * /) from the user and display the result by applying operator on the two number

6. WAP to enter any character and print it is Upper Case, Lower Case, Digit or symbol

7. WAP to enter 3 number and print the largest number

8. WAP to input day number and print corresponding day name for e.g if input is 1 output should be SUNDAY and so on.

Nested if

- In this type of “if” we put if within another if as a statement of it. Mostly used in a situation where we want different else for each condition. Syntax:

if condition1:

if condition2:

statements

else:

statements

elif condition3:

statements

else:

statements

WAP to check entered year is leap year or not

```
year = int(input("Enter any year "))
if year % 100 == 0:
    if year % 400 == 0:
        print("Year is Leap Year ")
    else:
        print("Year is not Leap Year")
elif year % 4 == 0:
    print("Year is Leap Year ")
else:
    print("Year is not Leap Year")
input()
```

Program to read three numbers and prints them in ascending order

```
x = int(input("Enter first number "))
y = int(input("Enter second number "))
z = int(input("Enter third number "))
if y>=x<=z:
    if y<=z:
        min,mid,max=x,y,z
    else:
        min,mid,max=x,z,y
elif x>=y<=z:
    if x<=z:
        min,mid,max=y,x,z
    else:
        min,mid,max=y,z,x
elif x>=z<=y:
    if x<=y:
        min,mid,max=z,x,y
    else:
        min,mid,max=z,y,x
print("Numbers in ascending order = ",min,mid,max)
```

Storing Condition

- In a program if our condition is complex and it is repetitive then we can store the condition in a name and then use the named condition in if statement. It makes program more readable.
- For e.g.
- `x_is_less=y>=x<=z`
- `y_is_less=x>=y<=z`
- `Even = num%2==0`