Data Analysis using Python

Name : Tuttapogu Swathi

class : III B.sc (Computer Science)

Admission No : 19740

1. Explain Programming and python in detail.

Programming Definition :-

Programming is the process of writing instructions that a Computer follows to perform a task.

Examples of programming languages:

* C, C++, Java, python, Javascript
* Python is preferred for beginners because it is simple & readable.

Purpose of programming:

* To solve the problems
* To create apps, websites
* To reduce the time and effort
* To automate the tasks

What is python?

Python is a high-level, interpreted, general-purpose programming language

* It has simple syntax.
* length of the code is small.
* Python is Beginner-friendly.

characteristics of python

* Simple and easy to learn
* It is open-source
* High-level programming language
* Dynamically typed
* Platform Independent
* It is procedural and object-oriented

* It is interpreted programming language

* It is portable

* It is procedural & object-oriented programming language.

Applications of python:

* Web development

* Artificial Intelligence and Machine learning

* Data science and Analytics

* Automation

* Game Development

* Desktop Applications

Comments

* Comments are the non-executable lines used for explanation

* These are used to understand the purpose of the program.

* Comments are ignored by the python.

* It makes the program readable and understandable

Types of Comments in python:

Python supports two types of comments.

* Single-line Comment

* Multi-line Comment

Single-line Comment

Syntax: # This is a Single-line Comment

Multi-Line Comment

Syntax:

```
'''
This is multi-line
string, used as a
Comment
'''
```

Importance of python in modern Software development :-

* Python is important in modern software development because it is easy to read, write and understand.

* Python is easier for the beginners to learn faster because of its simple syntax.

* It helps the developers to build programs with less time and effort.

* Python works on various operating systems like windows, MacOS, Linux.

* It is widely used in various fields such as web development, data science, Artificial Intelligence etc.

* It has a wide range of standard libraries which allows us to develop the programs faster.

* Python is a interpreted language which allows the developers to detect the errors easier and makes the code cleaner.

* Python, because of its simplicity, flexibility and accessibility is one of the most popular programming languages.

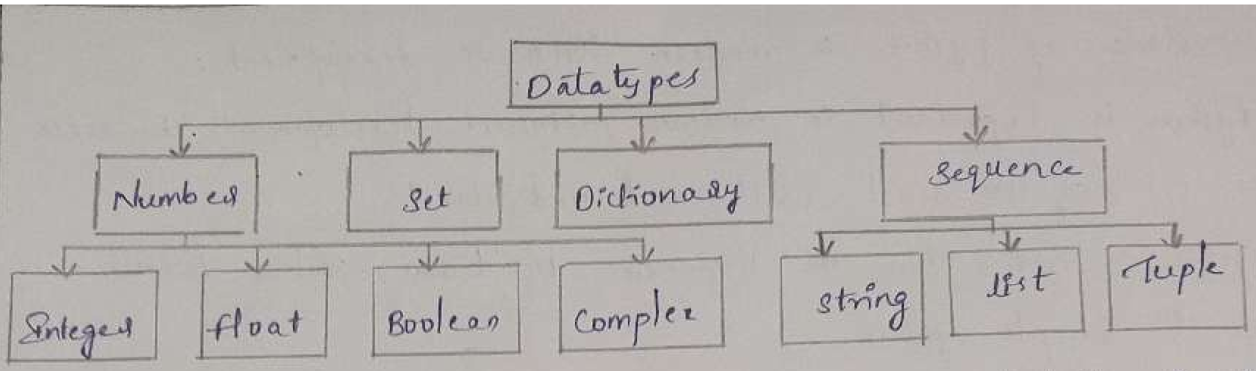
2. Describe data types and operators in python with suitable examples.

Built-in datatypes in python

Datatype :

A datatype tells the computer that what kind of data a value is.

Python provides several built-in data types to store and manipulate the data.



1. Numeric Data type: In python, the Numeric data type is used to store the numeric values.

-examples: Integer, floating number and complex numbers.

* Integer:- It is used to store whole numbers without fractions or decimals.

* float :- It is used to store the decimal point numbers.

* Complex :- It is used to store the complex numbers and is represented by complex class.

-ex:- 1+5i

* Boolean :- It is the python datatype with one of the two built-in values, True or false.

2. Sequence Datatype: The sequence datatype in python is the ordered collection of similar or Different Python data types.

The following are the sequence datatypes in python.

* string: string is the group of characters. It is represented by the str class. In python, string can be represented using single quotes, double quotes or even triple quotes all are treated as same.

Ex:- x = 'python'

* List : A list is the ordered collection of the mutable elements represented within square brackets. We can store multiple datatype values in list.

ex:- List = [1, 1.2, 'Python']

* Tuple : Tuple is the ordered collection of immutable elements stored in parenthesis . Tuples cannot be modified after it is created.

ex :- tuple = ("Bike", "car", "Bus")

3. Set :- It is an unordered mutable collection of unique items and should be written in curly braces.

Ex:- n = {1, 2, 3, 2, 1}

In the set , duplicate items will not be printed.

4. Dictionary : It is a ordered collection of key-value pairs and each key must be unique. Dictionary is mutable.

It is defined using curly braces { } with key-value pairs seperated by colons.

```
Ex:- P = {"name" : "swathi", "Group" : "B Sc", "Year" : 3}
      print(person['name'])
      Output : swathi
```

Type identification using type():

In python, type identification means finding the datatype of a value or variable.

the type( ) function is used to find the datatype of the value.

It returns the datatype of the value assigned.

```
ex :- s = "string"
      print( type(s))
output :- <class 'str'>
```

Operators :.

-An operator in python is a symbol that performs operations on values and python variables.

Types of python operators :

Python operators are categorized in the following categories.

1. Arithmetic operators

2. Comparison operators

3. Assignment operators

4. logical operators

5. Bitwise operators

6. Membership operators

7. Identity operators.

1. Arithmetic operators.

Arithmetic operators are used to perform addition, subtraction, multiplication, division, modulo division, exponential and floor division.

Operator	Name	example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus	22%10 = 2
**	Exponent	4**2 = 16
//	floor Division	11//6 = 1

2. Comparison operators :-

* Comparison operators are used to Compare two values.

* Based on the Comparison, they produce boolean value (True or false)

Operator	Name	Example
==	Equal	2==2 is true
!=	Not equal	3!=4 is true
>	Greater than	5>4 is true
<	less than	5<4 is not true
>=	Greater than or equal to	5>=4 is true
<=	less than or equal to	4<=5 is true

3. Assignment operators

* Assignment operators are used to assign values to variables.
* It assigns the value on the right side to the variable on the left side.

operator	Name	Example
=	Assignment Operator	x = 10
+=	Addition assignment	x+=1
-=	Subtraction assignment	x -=5
*=	Multiplication assignment	x*=5
/=	Division assignment	x/=5
**=	Exponent Assignment	x**=2
%=	Remainder assignment	x%=5
//	floor division Assignment	x//=3

4. logical operators:

operator	Description	Example
and	If both operands are true then the condition becomes true	(a and b) is true
or	If any one of the operands is true then the conditions becomes true	(a or b) is true

not	Used to reverse the logical state of the operand	not(a and b)

5. Bitwise operators

Bitwise operators are used to execute the operations on the bits of the binary integers.

operator	Name
&	Binary AND
|	Binary OR
^	Binary XOR
~	Binary ones Complement
<<	Binary left shift
>>	Binary Right shift

6. Membership operators

In python, Membership operators are used to determine whether a certain value occurs in a sequence or not.

operator	explanation	example
in	Present	a in b
not in	not present	a not in b

7. Identity operators

In python Identity operators are used to Compare two objects memory address. If both refer to same memory address they return true and false otherwise

operator	explanation	example
is	Same object	x is y
is not	different object	x is not y

Real-world usage of operators

Operators are used in daily life whenever we calculate, compare and make decisions.

* In shopping, operators help calculate total bill, discount and balance amount.
* In banks, they are used to add salary, check the minimum balance.
* In schools, they compare the marks to decide pass or fail.
* In traffic signals, decisions are taken, like stop or go based on conditions.
* In mobile apps, they check the login details and show the results accordingly and checks the authenticity.
* In games, they are used to increase the score.

In offices, they calculate the salary and the working hours.

so, in real life, operators are used to calculate things, compare values and make decisions accordingly like we do while shopping, studying or using mobile apps.

3. Explain Python input and output operations in detail.

What is input?

Input is the data given to the program by the user through the keyboard.

This helps in making the programs interactive and dynamic,

Input in python - input() function.

Purpose of the input()

* The input() function is used to take input from the user.
* Input is always read as a string

Syntax :

```
x = input("Message")
```

Example :-

```
x = input("enter your name :")
print("Hello", x)
```

Default datatype:-

The value taken by input( ) is string (str) by default, even if you enter a number.

Type conversion while taking input

when we use input ( ), the value is taken as string by default. If we want to use it as a number, we must convert it to the required data type.

example:- If we give 25
System reads it as "25" (string)

Since input is always string, conversion is required.

Integer input

```
a = int(input("enter a:"))
print(a)
```

Float input

```
b = float(input("enter b:"))
print(b)
```

Taking Multiple inputs:

Taking multiple inputs means getting more than one value from the user.

Using seperate statements

By using seperate input statements each value is taken one by one.

ex:-
```
a = int(input("enter a:"))
b = int(input("enter b:"))
```

Using split()

By using the split() method many values can be taken at once

Example:-

```
a, b = map(int, input("enter two numbers: ").split())
print(a, b)
```

What is output?

Output is the result produced by the program.

Output in python - output can be taken using print() function.

purpose of print()

- It is used to display output on the screen.
- It can print text, numbers, variables and expressions.

Syntax

```
print(value)
```

Printing text

```
print("Hello python")
```

Printing Numbers and expressions

```
print(10)
print(10+5)
```

Printing variables

```
x = "python"
print(x)
```

Printing multiple values

```
a = 10
b = 20
print("sum is :", a+b)
```

New line in output

- Each print() creates a new line.
- \n is used for line break.

Ex:- print("Hello \n Python")

Formatting output using seperators.

In python, sep is used to seperate the multiple values with the specified symbol.

Example:- print("Apple", "Banana", sep=",")

output:- Apple, Banana

formatted output using formatting specifiers.

Formatting specifiers are used to control how values are displayed in the output.

example:-
```
name = "swathi"
age = 21
print("Name: %s Age: %d" %(name, age))
```
Output:- Name: swathi Age: 21

4. Discuss control statements and Decision - Making statements in Python.

Control statements in python:

Control statements are used to control the flow of execution of a python program. By using control statements, we can write decision-making, looping and flow altemating programs.

Meaning of control statements:

* Control statements decide which part of a program runs, when it runs and how many times it runs.

Importance of control statements:

* They help the program take decisions, repeat tasks and control the flow of execution.
* Without control statements, a program would run line by line only, with no thinking or logic.

Types of control statements in python:

Python control statements are broadly classified into three categories.

1) Decision Making statements
2) Looping statements
3) Jump| control transfer statements

1. 1) Decision Making statements

* Decision making statements are used to execute the code based on conditions.

a) If statement

* It executes the block of code only if the condition is True and skips the block otherwise.

Syntax:
```
if condition:
        statement
```
Example:-
```
age = 20
if age >= 18:
    print("eligible to vote")
```
b) if-else statement

* It executes one block if condition is True and another block if condition is false.

Syntax:
```
if condition:
        statement
else:
        statement
```
Example:-
```
num = 5
if num % 2 == 0:
        print("Even number")
else:
        print("odd number")
```
c) if-elif-else statement.

* It is used to check multiple conditions.

Syntax:-
```
if condition1:
        statement
elif condition2:
        statement
else:
        statements
```

2. Looping statements

Looping statements are used to execute the particular block of code for a repeated number of times.

a) for loop:

A for loop is used to iterate over a sequence such as a list, tuple, string or range.

Syntax:- for item in sequence:

```
for item in sequence:
        statements
```

example:-

```
for i in range(1, 6):
        print(i)
```

b) while loop:

It repeats the block of code as long as the condition is True.

Syntax:-

```
while condition:
        statements
```

ex:-

```
i=1
while i<=5:
    print(i)
    i+=1
```

3. Jump statements

It's Used to alter the flow inside loops.

a) Break statement

The break statement is used to terminate a loop immediately when a condition is satisfied.

Syntax: break

Example:-

```
for i in range(1, 6):
        if i==3:
            break
        print(i)
```

b) Continue statement

It is used to skip the current iteration and move to the next iteration.

Syntax : Continue

```
        for i in range(1, 6):
            if i= =3:
                continue
            print (i)
```

c) Pass statement :-

The pass statement is used as a placeholder where a statement is required synptactically but no action is needed.

syntax : Pass

```
ex:- for i in range (1,6):
         if i= =3:
             pass
         print(i)
```

5. Write an essay on Python programming fundamentals.

Python is a popular and beginner-friendly programming language. It is widely used in various fields because it is easy-to-learn and understand.

It helps people solve real-would problems by giving clear instructions to the computer in a simple way.

Role of programming in Problem solving:

Programming plays an important role in problem solving. It helps us break a big problem into smaller steps and solve it logically. With programming we can automate tasks, reduce manual work and get accurate results quickly.

example:- calculating marks, managing bank accounts, or analyzing data becomes easy with the help of programs.

Python syntax simplicity and readability:

One of the biggest advantages of python is its simple syntax

Python code looks very close to the english language. which makes it easy to read and write.

Unlike other programming languages, Python does not use lengthy code. Because of this simplicity, beginners can understand

6 python programs without confusion.

Use of Comments for code documentation.

Comments are used in python to explain the code. They help others understand what the program is doing.

Comments are ignored by the computer, but they are very useful for programmers. With the use of them we can easily remember the purpose of the code in future.

Datatypes, operators and input/output operations:

Datatypes define the kind of data used in a program, such as numbers, text or decimal values.

operators are used to perform actions like addition, subtraction, Comparison and decision-making. Input and output operations allow users to enter data and see results on the screen. Together, these fundamentals help programs work correctly and interact with users.

Control flow using decision-making statements

Control flow statements decide how a program runs.

Decision-making statements like if, else, and elif allow the program to choose different paths based on conditions. These statements help the program think and act according to the situation, such as checking eligibility or displaying results.

Real-world problems using python programming.

1. Movie Ticket Pricing

```
age = int(input("enter age:"))
is3D = int(input("3D movie? (1 for yes, 0 for NO) : "))
if age < 13:
    price = 150
elif age < 60:
    price = 250
else:
    prince = 200
if is3D == 1:
    price = price + 50
print("final ticket price:", price)
```

2. College Attendance Rule

```
attendance = float(input("enter attendance percentage: "))
medical    = int(input("Medical Certificate(1=Yes, 0=No): "))
if attendance >= 75 or (attendance >= 60 and medical ==1):

    print("Allowed")

else:
    print("Not Allowed")
```

3. E-commerce discount

```
bill = float(input("Enter bill amount:"))
isPrime = int(input(" Prime Member? (1=Yes, 0=No): "))
if bill >= 5000:
    discount = 0.20
elif bill >= 2000:
    discount = 0.10
else:
    discount = 0
if isPrime==1:
    discount = discount + 0.05
final_amount = bill - (bill * discount)
print(" final amount to be paid: ₹ ", final_amount)
```

4. Smartphone Battery warning

```
battery = int(input(" enter battery percentage: "))
isCharging = int(input(" Is the phone charging? (1 = Yes , 0 = No) : "))
if ischarging==1:
    print(" charging ")
else:
    if battery <= 20 :
        print(" Low Battery")
    elif battery <= 80 :
        print(" Normal ")
    else:
        print(" fail ")
```

5. Driving license check

```
age = int(input(" enter age: "))
testpassed = int(input(" Passed driving test? (1=Yes, 0= No): "))

if age >= 60 :
    if age >=18 :
        print(" eligible ")
    else:
        print(" Not eligible ")
```

```
else :
    if age >= 18 and testPassed == 1 :
            print("eligible")
    else:
        print ("Not eligible")
```

6. Online food Delivery

```
amount = float(input(" enter order amount : "))
isGold = int(input(" Is user a Gold number? (1=Yes, 0=No): "))
distance = float(input(" enter distance in km:"))
if distance >10:
    print(" Delivery charged")
else:
    if amount >= 500 or isGold == 1:
        print (" free delivery")
    else:
        print (" Delivery charged")
```

7. Bank Loan Approval

```
salary = float (input (" enter salary: "))
creditScore = int (input (" enter credit score: "))
if salary >= 50000:
    print(" Loan Approved")
elif salary >= 30000 and creditScore >= 700:
    print ("Loan Approved")
else:
    print ("Loan Rejected")
```

8. electricity Bill

```
units = int(input (" Enter units Consumed :"))
if units <= 100:
    bill = units * 2
elif units <= 200:
    bill = (100 * 2) + ((units-100) * 3)
else:
    bill = (100 * 2) + (100 * 3) + ((units - 200) * 5)
print(" final bill amount: ₹ ", bill)
```

9. Student scholarship

```
marks = int(input (" enter students marks: "))
income = float(input(" Enter family income:"))
singleParent = int (input (" enter single parent. status (1 for Yes, 0
                for No)))
scholarship = (marks >= 85) and (singleParent ==1 or income< 500000
if scholarship:
    print (" the student is eligible for scholarship.")
else:
    print(" the student is not eligible for a scholarship.")
```

10. Online Exam Result

```
theory = int(input (" enter theory marks : "))
practical = int (input (" enter practical marks:"))
total = theory + practical
if (theory >= 40 and practical >= 40) or total >= 100:
    print (" Result: PASS")
else:
    print (" Result : FAIL")
```

11. Hotel Room Pricing

```
isWeekend = int(input(" enter 1 for weekend, 0 for Nomal day: "))
daysStayed = int(input(" enter number of days stayed:"))
if isWeekend ==1:
        rate = 4000
elee:
        rate = 3000
totalBill = rate * daysStayed
if daysStayed > 3:
        totalBill = totalBill - (totalBill * 0.15)
print("final Bill Amount : Rs", totalBill)
```

12. Gaming Level Unlock

```
score = int(input("enter score:"))
isPremium = int(input("enter 1 if premium pass, 0 otherwise:"))
usedCheat = int(input("enter 1 if cheating used, 0 otherwise:"))
if usedcheat ==1:
        print("Access Denied")
elif score >= 100 or isPremium ==1:
        print("Next level unlocked")
else:
        print("level locked")
```

13. Mobile Data usage.

```
dataUsed = float(input(" enter daily data used (in GB): "))
Unlimitedplan = int(input(" enter 1 if Unlimited plan, 0 otherwise: "))
isRoaming = int(input("enter 1 if roaming is ON, 0 otherwise: "))
if isRoaming ==1:
        print(" Unlimited Data Not Available")
```

```
elif dataUsed <= 2 or Unlimitedplan == 1:
    print (" Unlimited Data Available")
else:
    print ("Limited Data only")
```

14. Office entry System

```
idvalid= int(input("Enter 1 if ID card is valid, 0 otherwise :"))
fingerprint = int(input ("Enter 1 if finger matches , 0 otherwise :"))
facescan = int(input ("Enter 1 if face scan matches, 0 otherwise:"))
isHoliday = int(input("Enter 1 if today is a holiday, 0 otherwise:"))
if isHoliday ==1:
    print("Entry Denied")
elif idvalid ==1 and (fingerprint ==1 or facescan ==1):
    print (" Entry Allowed")
else:
    print("Entry Denied")
```

15. Movie Rating Display

```
AverageRating = float(input("Enter average rating:"))
isEditorsChoice = int(input("Enter 1 if editor's choice, 0 otherwise:"))
if isEditorsChoice ==1:
    print("Recommend")
elif averageRating >=8.5:
    print (" Excellent")
elif 8.0 <= averageRating <=8.4
    print (" Good")
else:
    print("Average")
```