# Data Analysis using Python

## Assignment...

By

M. Amrutha

3rd BSc Computer Science

Admno:- 19735

1. Explain Programming and Python in detail.

Programming: It is the process of designing, writing, testing and maintaining a set of instructions that a computer follows to perform specific tasks.

purpose of programming:

* To solve real-world problems logically.
* To automate repetitive tasks.
* To process and analyze data.
* To control hardware and machines.

Python Programming Language

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

-> It is created by Guido Van Rossum in 1991.

Characteristics of Python

* Simple and Easy to learn - which will be executable in fewer lines.
* Interpreted language - No need for compilation.
* Object Oriented - It supports classes and Objects.
* Portable - runs on any windows.

## Applications of Python:

* Web development.
* Data Science and Data Analysis.
* Desktop applications.
* AI and Machine Learning.

## Types of Comments in python with Syntax:

→ Comments are used to explain code and improve readability. They are ignored by interpreter.

1. Single line Comment :- Used for short explanation.

```
Syntax:- # This is single line comment
         code
```

2. Multi line Comment :- It is used for long explanation. Python uses the triple quotes.

```
Syntax:- """
         This is multi line comment
         it has several line of explanation
         """
         code
```

3. Inline Comment:- It will written on the same line as code

```
Syntax:- x = 10  # Assigning a value
```

Importance of Python in Modern Software development :-

* Faster development - less time consumes
* High demand in industry - like Google, social media websites.
* Strong Community Support.
* Integration Capability:

2. Describe data types and Operators in Python with Suitable examples.

In python, data types define the type of data variable can store.

While Operators are symbols used to perform operations on variables and values.

Built in data types in python

Python have several built-in data types

1. Numeric Data types

| Type | Description |
|---|---|
| int | integer values |
| float | Decimal Values |
| Complex | Complex numbers. |

Ex:-

```
a = 10    # int
b = 3.14  # float
c = 2+3j  # complex
```

2. Sequence Data types :- To store ordered collection data

| Type | Description |
|---|---|
| List | Mutable Sequence |
| tuple | Immutable Sequence |
| String | Sequence of characters |

Ex:- list1 = [1, 2, 3]

tuple = (4, 5, 6)

name = "python"

3. Set Data type :- Used to store unordered and unique elements.

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

print (set1)

4. Mapping Data Type:- Used to store data in key-value pairs

Ex:- student = {"name": "Amrutha", "age": 20}

5. Boolean Datatype: Used to store True or False values

Ex:- x = True

y = false

Type Identification using type() function

The type() function is used to identify the data type of a variable

Syntax:- type (variable)

Ex:-
```
a = 10
print (type(a))
b = "python"
print (type(b))
```

## Various Python Operators

Operators are symbols used to perform operations on variables

### 1. Arithmetic Operators:-

Used to perform mathematical operations.

| Operator | Operation |
|---|---|
| + | Addition |
| - | Subtraction |
| * | Multiplication |
| / | Division |
| % | Modules |

Ex:-
```
a=10
b=3
print (a+b)
print (a%b)
```

### 2. Assignment Operators:-

Used to assign values to variables

| Operator | Operation |
|---|---|
| = | Assignment |
| += | Addition assignment |
| -= | Subtraction assignment |

| | |
|---|---|
| *= | Multiplication Assignment |
| /= | division Assignment |
| %= | Modulus Assignment |

Ex:-
```
x = 10
x += 5
print(x)
```

3. Comparison Operators :- Used to compare values

| Operator | Operation |
|---|---|
| == | Equal |
| != | Not Equal |
| > | Greater than |
| < | Less than |

Ex:-
```
a = 5
b = 10
print (a<b)
```

4. Logical Operators :-

Used to combine conditional statements.

| Operator | Meaning |
|---|---|
| and | Logical AND |
| or | Logical OR |
| not | Logical NOT |

Ex:-
```
a = True
b = False
print (a and b)
```

5. Membership Operator :-

Used to test whether a value is present in a Sequence

| Operator | Meaning |
|---|---|
| in | Present |
| not in | not present |

Ex :- list 1 = [1, 2, 3]

print (2 in list1)

6. Identity Operators :-

Used to compare memory locations of object

| Operator | Meaning |
|---|---|
| is | Same object |
| is not | Different object |

Ex :- a = 10

b = 10

print (a is b)

Real World usage of Operators :-

* Used in billing systems and Calculators.
* Used in decision making programs
* Used in searching data.
* Used to update values.

3. Explain Python Input and Output Operation in detail.

Input and Output (I/O) operations :-

In python allow program to interact with the user

Input operations :- This is used to take data from the user.

Output operations :- This is used to display result to the user

Python mainly uses the input() function for input and the print() function for output.

1. input() Function and its default data type

The input() function is used to take input from the user through the keyboard.

Syntax :-

```
Variable = input ("Message")
```

Ex :-

```
name = input ("Enter your name :")
print (name)
```

→ The default data type of input() is string (str), even if the user enters numbers.

2. Type Conversion while taking input :-

Since input() returns a string, we must convert it to the required data type using type conversion functions.

Common Type Conversion functions

- int() → converts to integer
- float() → converts to decimal number.

- str() → converts to string

Ex:-

1) age = int (input("Enter your age"))
   print(age)

2) Price = float (input("Enter price"))
   print(price)

3. Taking Multiple Inputs

Python allows taking multiple inputs in a single line using the split() method.

Ex:- a, b = input ("Enter two numbers:").split()
   a = int(a)
   b = int(b)
   print (a+b)

4. Formatted Output Using print() function

The print() function is used to display output on the screen.

a) Using Separators(sep)

The sep parameter is used to seperate multiple values.

Ex:- print (10, 20, 30, sep=",")

b) Using End parameters(end)

The end parameter controls what is printed at the end.

Ex:- print ("Hello", end=" ")
   print ("World")

c) format() Method

This is used format (or) creating the formatted strings

Ex:- name = "python"

Version = 3.10

print ("Language : {} Version : {}". format (name, version))

d) f-string (Modern Method)

Ex:- marks = 85

print (f" marks obtained : {marks}")

4. Discuss Control Statements and Decision - Making Statements in Python

In python, control statements are used to control the flow of execution of a program. Normally, Python programs execute statements line by line from top to buttom. Control statements allow us to change this normal flow based on condition or repetitions.

Meaning and Importance of Control Statements

→ Control statement decide which statement should be executed.

→ How many times a block of code should run, when to stop or skip execution.

Importance :-

* Helps in decision-making.
* Reduces complexity of programs.
* Makes program flexible and dynamic.

Types of Control Statements in Python

Control Statements in Python are broadly classified into:

1. Decision - Making Statements
2. Looping Statements.
3. Jump statements.

1. Decision - Making Statement.

Used to execute different blocks of code based on conditions.

(a) if statement :-

The if statement executes a block of code only when the condition is True.

syntax :- if condition:
    statements

Ex : age = 18
    if age >= 18:
        print ("Eligible to vote")

(b) if - else statement :-

The if-else statement provides an alternative block when the condition is False.

Syntax :- if condition:
    statements
else:
    statements

Ex:-

```
num = 5
if num % 2 == 0:
    print ("Even number")
else:
    print ("odd number")
```

c) if-elif-else Statement

Used to check multiple conditions.

Syntax :-

```
if condition1:
    Statements
elif condition2:
    statements
else:
    statements
```

2. Looping Statements

Used to execute block of code repeatedly.

(a) For loop:-

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

Syntax:

```
for variable in sequence:
    statements.
```

Ex:- for in i in range (1, 6):
    print (i)

b) While loop

A while loop is used to repeat a block of code as long as a condition is True.

Syntax:- while condition:
    statements

Ex:- i = 1
while i <= 5:
    print (i)
    i += 1

3. Jump Statements

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

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

b) Continue Statement :-

The continue statement is used to skip the current iteration and move to the next iteration.

Syntax: continue

```
Ex:- 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 syntactically 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.

Introduction

Python is one of the most popular and widely used programming languages in the world today. It is a high-level, interpreted, general purpose language known for its simplicity and readability. Python programming fundamentals provide the

basic concepts required to develop efficient and reliable software applications.

## Role of Programming in Problem Solving

Programming plays a crucial role in problem solving by breaking complex problems into smaller, manageable steps. A programmer analyzes the problem, designs an algorithm and implements it using a programming language. Programming helps automate tasks, reduce human effort, and provide accurate solutions. Python with its simple syntax, allow programmers to focus more on solving problems rather than worrying about complex languages rules.

## Python Syntax Simplicity and Readability

One of the major strengths of python is its simple and readable syntax. Python uses English-like keywords and relies on indentation instead of braces, which makes the code easy to understand and maintain. Even beginner can quickly learn python and write programs with fewer lines of code compared to other languages.

## Uses of Comments for Code Documentation :-

Comments are used to explain the code and improve readability. They help other programmers understand the logic

of the program and are useful for future maintenance. Python supports single-line comments using # and multi-line comments using triple quotes.

## Data Types, Operators, and input/output Operations

Python provides various built-in data types such as integers, floating-point numbers, strings, lists, tuples, sets, dictionaries. Operators are used to perform operations on data, including arithmetic, comparison, logical and assignment operations.

Input and output operations allow interaction with users. The input() function is used to take input from the user, and the print() function is used to display output.

## Control flow Using Decision-Making Statements.

Control flow statements control the execution of a program. Decision-making statements such as if, if-else and if-elif-else allow the program to make decision based on conditions. These statements are essential for implementing logic in real-world applications like grading systems and validation programs.

# Real-World problems using python programming

## Movie Ticket Pricing

```
age = int(input("Enter age:"))
is3D = int(input("Is it a 3D movie? (1 for Yes, 0 for No):"))
price = 0
if age < 13:
    price = 150
elif 13 <= age <= 59:
    price = 250
else:
    price = 200
if is3D == 1:
    price += 50
print("Final Ticket Price: ₹", price)
```

## 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("Is Prime Member? (1=Yes, 0=No): "))

discount = 0
if bill >= 5000:
    discount = 0.20
elif bill >= 2000:
    discount = 0.10
else:
    discount = 0

if isPrime == 1:
    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 ("Full")
```

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")
```

Online Food Delivery

```
amount = float (input ("Enter order amount: "))
isGold = int (input ("Is user a Gold member? (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 student's marks: "))
income = float (input ("Enter family income: "))
singleParent = int (input ("Enter single parent status (1 for Yes, 0 for N
```

```
scholarship = (marks >= 85) and (singleParent == 1 or income < 500000)
if scholarship:
    print("The student is eligible for a scholarship.")
else:
    print("The student is not eligible for a scholarship.")
```

## 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")
```

## Hotel Room Pricing

```
isWeekend = int(input("Enter 1 for Weekend, 0 for Normal day:"))
daysStayed = int(input("Enter number of days stayed:"))
if isWeekend == 1:
    rate = 4000
else:
    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 :"))
is premium = int (input (" Enter 1 if premium pass, 0 otherwise: "))
usedCheat = int (input (" Enter 1 if Cheating used, 0 otherwise :"))

if usedCheat == 1:
    print ("Accen 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): "))
hasUnlimitedPlan = 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 hasUnlimitedPlan == 1:
    print (" Unlimited Data Available")
else:
    print ("Limited Data Only")
```

## Office Entry System

```
idValid = int (input ("Enter 1 if ID card is valid, 0 otherwise :"))
fingerprint = int (input ("Enter 1 if fingerprint 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")
if idValid == 1 and (fingerprint == 1 or facescan == 1):
    print ("Entry Allowed")
else:
    print ("Entry Denied")
```

## Movie Rating Display

```
averageRating = float (input ("Enter average rating :"))
isEditorChoice = int (input ("Enter 1 if editor's choice, 0 otherwise :"))

if isEditorChoice == 1:
    print ("Recommend")
if averageRating >= 8.5:
    print ("Excellent")
if 6.0 <= averageRating <= 8.4
    print ("Good")
else:
    print ("Average")
```