Explain Programming and python in detail?

Programming is the process of writing instructions that tells a computer what to do. These instructions help the computer solve problem & perform tasks like calculation, data, storage / decision-making.

Example: A program to add two numbers

```
a = 10
b = 20
Print(a+b)
```

What is python?

Python is a high-level programming language that is easy to understand and learn. It was developed by "GUIDO VAN ROSSUM" and is widely used because of its simple Syntax and readability.

Charecteristics of Python:

→ Easy to read and write
→ uses simple English-like Syntax.
→ Free and open source.
→ Supports multiple programming styles.

Applications of Python:

⇒ Web development
⇒ Data Analysis
⇒ Automation
⇒ Scientific Research

Example:-

```
Print("Hello, World")
```

Types of Comments in Python:

Comments are statements in a Program that are not executed by python. They are used to explain the code and improve readability.

1. Single-line comments:

→ Used to explain a single line of code
→ Begins with the # symbol.
→ Python ignores this line during execution.

Syntax:-

```
# This is a Single-line Comment
```

Example:

```
# Assign Value to Variable.
x = 10
```

2. Multi-line comments:

→ Used to explain multiple lines at once
→ Written using triple quotes (""" """ """)
→ Commonly used for documentation.

Syntax:

```
""" """
This is a multi-line-comment
""" """
```

Example:

```
"""
This program calculates Sum of
        two Numbers
"""
a = 5
b = 6
Print (a+b)
```

Importance of comments:-

→ Make code easy to understand.
→ Helps in debegging.
→ useful for documentation.

Importance of Python in Modern Software Development

→ It is Easy to learn and use
→ Simple Syntax and reduce Coding time
→ Supports multiple programming styles.
→ Has many libraries and frameworks.
→ Used in AI, data science, web development, and automation.
→ Works and different operating systems.

Data types and operators in python.

Python provides different built-in data types to store various kinds of data and operators to perform operations on them.

Built-in Data Types :-

1. Numeric Data types

Used to store numbers such as integers and decimal values.

```
a = 10      # integer
b = 3.5     # float
```

2. Sequence Data types:

Tuple - ordered collection; immutable
→ Used to store a collection of values.

```
name = "Kalyani"  # string
Marks = [80, 90, 95]  # list - ordered collection; Mutable
```

3. Set Data type :-

Stores unique elements

```
nums = {1, 2, 3}
```

list - [ ]
Tuple - ( )
Set - { }

4. Mapping Data-type

Stores Data in key-value pairs

```
Student - {"name": "Kalyani", "age": 20}
```

5. Boolean Data type:-

Stores Data as True or False values

result = True

int => a = 10, b = -12, c = 1 2 3 4 5 6 7

float => x = 1.0 ; y = 12.3 ; z = 13.4

Complex = Alphabet with number; A = 2 + 5j
(2 → real number, 5j → imaginary)

String - Sequence of characters depicted in quotation marks.

In python we can use Single, double / triple quotes to define a string.

"hello python".

Type Identification Using type()

The type() function in Python is used to identify the data type of a variable. It tells what kind of values is stored in the variable.

```
Type (variable_name)
a = 10
Print (type(b)) # int
b = 3.5
Print (type(b)) # float
name = "Python"
Print (type(name)) # str
nums = [1, 2, 3]
Print (type(nums)) # list
flag = True
Print (type(flag)) # bool
```

Operators in Python: (x + y) — + : operator; x, y : operands

used to perform operations on variables & values. Python operators are special symbols used to perform specific operations on one a s more operands.

Unary operators:

Python operators that require one operand to perform a specific operation known as unary operators.

Binary operators:

Python operators that require two operands to perform a specific operation.

Types of Python operators:-

Python operators are categorized in the following categories.

- Arithmetic Operators.

Used for mathematical calculations.

operators	Meaning	Example
+	Addition	10 + 5 = 15
-	Substraction	10 - 5 = 5
*	Multiplication	10 * 5 = 50
/	Division	10/5 = 2.0
%	Modules(remainder)	10 % 3 = 1
//	floor Dirision	10 // 3 = 3
**	Expenent	2 ** 3 = 8

Real-World use:

Calculating total marks, salary, discounts, average score, etc

2. Assignment operators.

Used to assign values to variables.

operators	Meaning	Example
=	Assign value	x = 10
+=	x = x+5	x += 5
-=	x = x-5	x -= 5
^=	x = x^2	x^= 2

Real world use: updating bank balance, increasing counters, modifying source

3. Comparision operators:

used to compare two values. Result is True / false

Operator	Meaning
= =	Equal
! =	Not Equal
>	Greater than
<	less than
>=	Greater than & Equal
<=	less than & Equal

Real-world-use:-

checking pass / fail marks, age eligibility, login validation.

4. logical operators:-

used to combine condition

Operators	Meaning
and	True if both conditions are True
or	True if any one condition is True
not	Reverse the result

Real-world use:-

checking multiple conditions like age and qualification.

Membership operators:-

used to check whether a value exists in a sequence.

Operators	Meaning
in	value exists
not in	Value does not exist

Example - 'a' in apple # True

Real-word-use:-

checking username is database, item in list.

Identity Operators-

Used to compare memory locations

is — Same object

is not — Different object

Real-world use:

checking object identity in programs

Python Input and output operations

Input operation in python.

input() function

→ used to take input from the user.

→ Default data type is string (str)

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

Type conversion while Taking Input

Since input is always string, type conversion is required.

```
age = int (input ("Enter age:"))
salary = float (input ("Enter Salary:"))
```

Taking Multiple Inputs:

Multiple inputs can be taken in one line using split()

```
a, b = input ("Enter two numbers : "). split ()
a = int(a)
b = int(b)
```

Output Operation in Python

Print() Function

used to display output on the screen.

```
Print ("Hello Python")
```

Separator (sep)

used to separate multiple values

```
Print(10, 20, 30, sep = " , ")
```

output

10 , 20 , 30.

Format Specifiers / formatted Output

using format()

```
Print ("My age is {}". format (age))
```

using f-string

```
Print (f"My Salary is {Salary}")
```

Real-world use:

Displaying supports, student details, invoice, bills..,

Control Statement and Decision-Making Statements:-

Meaning of control statements.

They control the flow of execution of a program.

they decide which statement to execute and when.

Importance:-

Enables decision Making

Makes programs logical and dynamic

Avoids unnecessary Execution.

Types of control statements:-

1. Decision-making statements
2. looping statement
3. Jump statement

Decision-Making Statements:

1 If-statement

Executes a block of code only if the condition is true.

Syntax:

```
if condition:
    Statement
```

Example:

```
if marks >= 35:
    Print ("pass").
```

2. If - else statement :

Execute one block if condition is true, otherwise executes another block.

Syntax :-

```
if condition:
    Statement 1
else
    Statement 2
```

Example :

```
if age >= 18
Print ("eligible to vote")
else
Print ("Not eligible")
```

3. If - elif - else statement

Used to check multiple conditions

Syntax :

```
if condition 1:
    Statement 1
elif condition 2:
    Statement 2
else
    Statement 3
```

Example :

```
if marks >= 75;
Print ("Distinction")
elif marks >= 60;
Print ("Frist class")
elit marks >= 35
Print ("pass")
else :
Print ("Fail")
```

Execution Flow :-

→ conditions are checked top to bottom
→ first true condition is executed.
→ Remaining conditions are skipped.

Example : (if-else execution flow)

1. condition age >=18 is checked.
2. condition is false
3. else block executes.
4. Program continues after if-else.

Python execute only the block whose condition is true.
Indentation python a crucial role in controlling flow.

Write an Essay on Python programming fundamentals.

1. Role of Programming in problem solving.

→ Programming helps in solving real-world problems using logical steps.
→ It taks complex problems into smaller, manggable tasks.
→ Enables automation of repetitive work.
→ Helps in accurate calculations and data processing.
→ Improves Efficiency and Solves time.

2. Python Syntax Simplicity and Readability.

→ Python has simple and English-like Syntax
→ Easy to learn for beginners.
→ Does not use braces { } or Semi colons;
→ uses indentation to define code blocks.
→ Programs are Easy to read, write & maintain.

3. use of comments for code Documentation.

→ Comments Explain the Purpose of code statements
→ Improve readability and understanding of programs.
→ Helpful for debugging and future modifications.

→ Single-line comments use #
→ Multi-line comments use triple quotes (''' or """)
→ Comments are not Executed by the interpretor

4. Data types, operators and Input/output operation:

→ Python supports built-in data type like.
  int, float, str and bool.
→ Operators perform arithmetic, comparision, logical and assignment operations.
→ input() function is used to take input from the user.
→ Default data type of input() is "string".
→ print() function is used to display output.
→ Supports formatted output for better Prensentation.

5. Control Flow using Decision-Making Statements.

→ Control flow decides the order of Execution in a program.
→ Decision-making statements helps in making logical choices.
→ if statement execute code when condition is true.
→ if-else provides two-way decision making.
→ if-elif-else handles multiple condition.
→ makes programs trencible and dynamic.

Conclusion:-

→ Python fundamentals from the base for advanced programming.
→ Simple syntax and powerful features make python popular.
→ widely used in education, Science, data, analysis and software development.

---

1) Movie ticket Pricing.

A movie theatre charges:
₹150 for children (age < 13)
₹250 for adults (age 13-59)
₹200 for Session (age ≥ 60)
If the person is watching a 3D movie, add ₹50 Extra. write a program that takes age & is 3D (1 or 0). and print the final ticket price.

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

---

② College Attendance Rule: A Student is allowed to write the exam if:

attendance ≥ 75

or

attendance ≥ 60 and has medical certificate (1 = yes, 0 = no)

Take attendance precentage and medical certificate as input and print "Allowed" or "Not Allowed".

```
attendance = int (input ("Entey attendance percentage : "))
medical = int (input (" medical cultificate (1 for yes, 0 for No): "))
if attendance >= 75 or (attendance >= 60 and medical == 1);
      Print ("Allowed")
else :
      Print ("Not Allowed")
```

---

③ E.commerces Discount

A shopping site gives

20% discount if bill ≥ 5000

0% discount if bill blw 2000 and 4999

discount if bill < 2000

But if the customey is a prime numbey they gel extra 5% discount.

```
input : bill amount ; is prime (1 & 0)
print final amount to be paid.
bill = float (input (" Entey bill amount : "))
```

```
if bill >= 5000:
    discount = 0.20
elif bill >= 2000:
    discount = 0.10
else
    discount = 0.0
is prime = int(input("Is the customer a prime member? (1 for yes, 0 for No): "))
if is prime == 1;
    discount += 0.05
final_amount = bill - (bill discount)
Print("final amount to be paid : ", final_amount)
```

---

4 Smartphone Battery warning

If phase shows:

"low Battery" if battery ≤ 20

"Normal" if battery between 21-80

"full" if battery ≥ 80

But if phone is charging, it should show "charging" instead of any message.

Input : battery percentage, is charging (1 or 0)

```
Battery = int (input ("Enter battery percentage : "))
if charging = int (input ("Is the device charging? (1 for yes, 0 for No) : "))
    if ischarging == 1;
        Print ("charging")
else
    if battery <= 20;
        Print ("low Battery")
```

---

5 During licence check

If person can get during because if :

age ≥ 18

AND

Passed draining test (1 = yes)

But if age ≥ 60, driving test is not required.

Input age, test passed

Print "Eligible" or "Not Eligible".

```
age = int(input("Enter age : "))
test passed = int(input("passed driving test ? (1 for yes, 0 for No): "))
    if age >= 60
        Print("Eligible")
elif age >= 18 and test passed == 1:
    Print("Eligible")
else :
    Print("Not Eligible")
```

---

6 Q

Online food Delivery

A restaurant gives free delivery if :

order amount ≥ 500

OR

user is a gold member

But if the distance is more than 10 km, delivery is never free.

Input : amount is Gold (100), distance

```
amount = float(input("Enter order amount : "))
is Gold = int(input("Is Gold member ? (1 for yes, 0 for No) : "))
distance = float(input("Enter delivery distance (km): "))
    if distance > 10 :
        Print("Delivery charged")
elif amount >= 500 or is Gold == 1:
        Print("Free Delivery")
else :
        Print("Delivery charged")
```

7. Bank loan Approval

A bank approver a loan if;

Salary ≥ 30,000 AND credit Scole ≥ 700

OR

Salary ≥ 50,000 (credit Scole is ignored)

Input : Salary , Credit Scole

Print "loan Approved" or "loan Rejected".

```
Salary = int (input ("Entey Salary : "))
CreditScole = int (input ("Entey Credit Scole : "))
if Salary >= 50000 or (Salary >= 30000 and Credit Scole >= 700):
else :
    Print (" loan Rejected ")
```

---

8. Electricity Bill

units consumed

First 100 units = ₹2/unit

Next 100 units ⇒ ₹3/unit

Above 200 units ⇒ ₹5/unit

Note: No loops

Print final bill amount

```
units = int (input (" Entey units consumed : "))
    if units <= 100;
       bill = unit * 2
   Elif unit <= 200;
      bill = (100*2) + (unit - 100) * 3
  Else : bill = (100*2) + (100*3) + (units - 200) * 5
   Print (" Electricity Bill amount : ₹", bill)
```

---

9. Student Scholarship

A Student gets a Scholarship if:

marks ≥ 85

AND

family income < 500000

But if the Student is a Single Parent child, income condition is ignored.

```
Input : marks, income, Singleparent (1 or 0)
marks = int (input (" Enter marks : "))
income = int (input (" Enter family income : "))
Single parent = int (input (" Single parent child ? (1 for yes, 0 for No) : "))
if marks >= 85 and (income < 500000 or Single parent == 1);
    Print (" Scholarship Granted ")
else :
    Print (" Scholarship Not Granted")
```

---

10. Online Exam Result

- A Student passes if :
- theory ≥ 40 AND Practical ≥ 40

But if total (theory + practical) ≥ 100, Pass Even if one is less than 40.

```
input : theory, Practical
theory = int (input (" Enter theory marks : "))
Practical = int (input (" Enter practical marks : "))
if (theory >= 40 and practical >= 40) or theory + Practical >= 100);
        Print ("pass")
Else :
    Print (" fail ")
```

---

11 Hotel Room Pricing

- A hotel charges :
- ₹ 3000 per day for normal days
- ₹ 4000 per day on weekends

if Customer stays more than 3 days, give 15% discount

```
input : isweekend (1 or 0), days stayed
Print final bill.
```

```
is_weekend = int(input("Is it a weekend stay? (1 for yes 0. for No): "))
days_stayed = int(input("Enter number of days stayed: "))
if is_weekend == 1:
    rate = 4000
else:
    rate = 3000
bill = rate * days_stayed
if days_stayed > 3:
    bill = bill * 0.85
Print("Total bill amount: ", bill)
```

---

12. Graning level unlock

~~if~~ game unlocks next level if:

Score ≥ 100

OR

Player has a premium pass

But if player used cheating, access is denied

```
Input: Score, is premium, used cheat
Score = int(input("Is premium player? (1 for yes, 0 for No): "))
if used cheat == 1:
    Print("Access Denied")
Elif Score >= 100 or is premium == 1;
    Print("Next level unlocked")
else:
    Print("level locked")
```

---

13 Mobrule Data usage

-If network gives unlimited dates if:

daily usage ≤ 2 GB:

OR

usey has unlimited plan

But if roaming is on, unlimited plan does not work.

Input : data used, has unlimited plan, is Roaming

datalused = float (input ("Entey daily data usage (GB): "))

has unlimated plan = int (input ("Has unlimated plan ? (1 foryes 0 fq No): "))

is Roaming = int (input("Is roaming on ? (1 foyes for No): "))

```
if is Roaming == 1:
    if datalused <= 2 :
    Print ("unlimited Data")
Else :
    Print ("limited Data")
Else:
  if data used <= 2 or has unlimitedplan == 1:
        Print ("unlimited Data")
    Else
      Print ("limited Data")
```

---

14. office Entry System

-An Employee can oatey the office if:

ID card is valid

-AND

-fingerprint matches OR face Scan matches)

But if it is a holiday, entry is devied for every one.

Input : idvalid, fingerpint, face Scan, is Holiday

```
idvalid = Int (Input ("ID Cage valid ? (1 for yes, 0 for No): "))
-fingerpint = Int (input("fingerprint match ? (1 for yes, 0 fa No): "))
```

```
face Scan = int (input (" face Scan match ? (1 for yes, 0 for No ): "))
is Holiday = int (input (" Is today a holiday ? (1 for yes, 0 for No ): "))
if is Holiday == 1:
    Print ("Entry Denied")
elif id valid == 1 and (finger print == 1 or face Scan == 1):
    Print ("Entry allowed")
else:
    Print ("Entry Denied")
```

---

15. Movie Rating Display

If movie app shows rating based on average score:

Average ≥ 8.5 => "Excellent"

Average between 6.0 and 8.4 → "Good"

Average < 6.0 = "Average".

But if the movie is masked as editors choice always show "Recommended".

Input : average Rating, is Editorchoice (1 or 0)

Print the message.

```
average Rating = float (input (" Enter average rating : "))
is editors choice = int (input (" Is editors choice ? (1 for yes, 0 for No ): "))
if is Editorschoice == 1:
    Print (" Recommended")
elif average Rating >= 8.5 :
    Print (" Executed")
elif average Rating >= 6.0 :
    Print (" Good")
Else :
    Print (" Average")
```