

PYTHON ASSIGNMENT

1. Explain Programming and Python in detail.

• Definition and purpose of programming:-

Def:- A python programming is a set of statements and Expressions organized into logical structure, stored in a file with a

*py Extension. A python program includes Source Code, Execution, Readability, modularity.

Purpose:- The purpose of a python program is to solve problems and automate tasks across a wide variety of domains, general purpose language allows it to be used for several application

• characteristics and applications of Python:-

characteristics:- Python has plenty of features that make it the most demanding & Popular.

- Easy to read and Understand
- Interpreted language
- Object Oriented programming language
- Free and Open Source
- Multi-Platform
- Dynamically Typed
- Versatile
- Extensible
- Libraries
- Huge Community

Applications for Python: Python is suitable for wide array of applications. includes following

- Web Development: Used for server side development with powerful framework.

- Data Science analysis: Python is leading language due to libraries such as Numpy, Pandas and matplotlib.

- AI/ML: Essential for developing AI models using Tensor flow and PyTorch.

- Software Development: Serves a support language for software development tasks like testing & bug tracking.

* Types of Comments in Python With Syntax

Comments: Comments are the lines in the code that are ignored by the interpreter during the execution of the program. Comments enhance readability of code. Comments can be used to identify functionality or structure of code. It helps in understanding unusual scenarios.

Single line Comment:

In Python single line comments starts with hashtag symbol

Eg: # Sample Comment
name = "geeksforgeeks"
print(name)

output
geeksforgeeks

Multi-line Comments:

We Can Use multiple hashtags (#) to Write multiline Comment in Python. Each and Every line will be Considered as a Single line Comment. `/*.....*/` not Used in Python.

```
#multiline Comment
```

```
Print("Multiline Comments")
```

```
#output
```

Multiline Comments

* Importance of Python in modern Software development:-
Python has become a Cornerstone of modern Software development due to its simplicity, Versatility, and Extensive libraries and frameworks. It enables rapid development.

→ Readability

→ Flexibility

→ Extensive libraries

→ Data Science

Readability: Python human like syntax makes it Easy to learn, read, and maintain. It lowers the barriers to Entry for new developers and improves Collaboration among large teams by making it Easy to Understand.

Data Science: Numpy, Pandas, Tensor Flow, and PyTorch for data analysis, machine learning, deep learning.

Extensive libraries: Python has a Vast Collection of third party modules and a standard library. These tools accelerate the development Process.

2) Describe Datatypes and Operators in Python.
Datatype: Datatype in Python are a way to classify data items. They represent the kind of value that Operations. Since Everything is Perfor an object.

Numeric : int float, Complex

Sequence Type: string, list, tuple

Mapping Type: dict

Boolean : Bool

Set type : set, frozenset

Binary Type: bytes, bytearray.

i) (Python numbers): Numeric:-

These numbers represent data that has a numeric value. A numeric value can be integer, float number or any other Complex number. These values can be defined int, float, Complex classes.

integer:

Value represented by int class, it contains positive or negative numbers. There is no limit to how long an integer should be.

Float:

Value that can be represented by decimal numbers. It is real number represented by float presentation. or character e or E.

Complex: It is represented by Complex class. (Real Part) + (Imaginary Part) For Example $2+3j$ $2+3i$

Basic Data Types

Ordered Collection
Mutable [can change]

Ex:
numbers = [1, 2, 3, 4]

Range

Represents sequence of numbers
Commonly Used in loops.

Ex: x = range(1, 5)

4) Set Data Type

Set:

Unordered Collection

No duplicate values

It stores only Unique values

Ex: S = {5, 10, 15, 20, 25}

5) Mapping Data Type

dict.

Store values data in Key-Value Pairs

Key values must be Unique.

Ex: student = {"name": "Abhi", "class": 5th}

Accessing a value

Print (student["name"])

6) Boolean:

Stores only True or False

Used in decision making statements

Ex: is-valid = TRUE

Tuple

Ordered Collection
Immutable [cannot change]

Ex:
Colors = ("red", "pink", "skyblue")

Type identification using type():

The `type()` function in Python tells what kind of data is

It is mainly used to identify data types at Run time

Syntax:

`type(object)`

Ex:-

`x = 10`

`Print(type(x))`

Output:- int

Various Python Operators (Arithmetic Operators, Comparison Operators, Assignment, Logical, Bitwise, Membership, Identity)

Arithmetic Operators:

Arithmetic Operators are used to perform mathematical calculations. They work with members like integers and floats.

Operator meaning:

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
//	Floor division
**	Exponent

Ex:

`a = 20`

`b = 5`

`Print(a ** b)`

Assignment Operators

Assignment Operators in programming assign a value from the right side to a Variable on the left side. It includes $=$, $+=$, $-=$, $*=$, $/=$

* Simple assignment operator

$=$: Assigns the value on the right to the Variable on the left

$x=10$; Assigns the value 10 to Variable x .

Comparison Operators

Comparison Operators are symbols used in Programming and math to compare two values, returning a Boolean result (True/False). It includes $==$ (equal), $!=$ (Not Equal), $>$ (greater than), $<$ (less than), $>=$ (greater than or equal), $<=$ (less than or equal)

Ex $5 == 5$ true

$5 != 8$ true

Logical membership Operators

Logical Operators (AND, OR, NOT) are used to modify boolean expressions (true/false), decision making, and control programming flow.

Membership Operators in and $not in$. These Operators are used to test whether a variable is found or not in a list, tuple, Dictionary, set, etc. Come under membership Operators.

Identity Operators

It includes is and $isnot$. Check whether a variable refers to the same object in memory.

Real World Usage of Operations

Arithmetic Operations include Computer Graphics, Data Compression, Cryptography, Games like Video Games, Baseball, Basketball, Basic Calculations, Education, Finance.

3) Explain Python Input & Output Operations in detail

* input() function & default data type.

Function Name: input()

Purpose: Pauses program execution and waits for the user to type text and press Enter key.

Syntax

Variable_name = input ("Optional prompt message.")

Return Value: Function captures entered data and returns it as a string.

* Default data type

Regardless of whether the user enters letters, numbers, symbols, the value returned by input()

```
age = input("Enter your age: ")  
print(type(age))
```

Output If user enters 25
it is stored as "25"
not the integer 25

Python allows taking multiple inputs in a single using the split function

Using Split:

Split() divides inputs based on spaces by default
The value can be converted to the requested data type

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

Output Operation in Python:

Python uses `print()` function to display output

Syntax `Print(Value1, Value2)`

Formatted Output Using Print()

a) Using Separators (Sep)

The `sep` parameter defines how multiple values are separate

Ex: `Print("python", "is", "Easy", sep = "-")`

#output python-is-Easy

b) Using End Parameter (end)

Controls what is printed at the end of the output

Ex: `Print("Hello", end = " ")` #output Hello World
`Print("World")`

c) Format Specifiers
Using f-strings
Available from Python
Allows embedding

Ex: `name = "Alice"`
`marks = 95`

`Print(f"Name {name}, Marks {marks}")`

* Type Conversion while Value
Type Conversion is the process of changing a Value

One data type to another.

There are 2 types of type Conversion:

1. Implicit

2. Explicit

Implicit type Conversion:

Implicit Conversion happens automatically when data type
Used together in an Expression.

Smaller data type into larger type

Commonly Occurs when integer, float are Combined

Ex

x = 10

y = 10.6

z = x + y

Print("x:", type(x))

Print("y:", type(y))

Print("z=", z)

Print("z:", type(z))

Explicit type Conversion:

Programmer manually changes a Value from one datatype
to another.

Python built-in functions like int(), float(), str(),

Ex S = "100"

a = int(S)

Print(a)

Print(type(a))

Decision making statements in Python:-

if-statement

Syntax: if Condition:
Statement(s)

Ex: age = 18
if age >= 18:
Print("Eligible
to Vote")

if-Else

Syntax: if Condition:
Statement(s)
Else:
Statement(s)

Ex: num = 5
if num % 2 == 0:
Print("Even number")
Else:
Print("odd number")

if-elif-Else

Syntax: if Condition 1:
Statement
elif Condition 2:
Statement
elif Condition 3:
Statement
Else:
Statement

Ex: marks = 72
if marks >= 90:
Print("Grade A")
elif marks >= 75:
Print("Grade B")
elif marks >= 50:
Print("Grade C")
Else:
Print("Fail")

Syntax flow & Execution

Nested-if Ex

```
num = 10
if num > 0:
    if num % 2 == 0:
        Print("Positive Even Number")
    Else:
        Print("Negative Positive odd Numbers")
```


4) Discuss Control statements & Decision-making

i) Meaning & importance of Control statements

Control statements are programming constructs that control the order in which instructions are executed in a program.

- Enable decision-making
- Allow repetition of code
- Improve program efficiency
- Logical conditions & real world scenarios

ii) Types of Control statements in Python

a) Decision making

Used to execute different blocks of code

if

if else

if-elif-else

b) Looping statements

Used to repeat block of code.

for

while

c) Jump statements

break

continue

pass

return

Write an Essay on Python Programming

Python Programming Fundamentals

Programming plays a crucial role in solving by helping humans to instruct computers to perform tasks efficiently & accurately. Python has gained popularity due to its simplicity, readability, & powerful features.

Role of Programming in Problem Solving:-

Programming is a process of designing solutions to express solutions in the form that a computer can execute. It helps in breaking complex tasks into simple problems.

Python Syntax readability & Simplicity

Python is known for its easy & simple syntax, which resembles English language. Unlike many other languages, Python does not use complex symbols.

Use Comments for Code Documentation:

Comments are used in Python to explain the purpose of code & improve readability. Python supports single line comments using # symbol & multi-line comments using ''' or """.

Datatypes, Operators, & Input/Output Operations

Python supports various data types such as integers, float - point numbers which are used to store

different kinds of data. Operators are Symbols

Perform Operations on data. (+, -, *, /) (<, >, =)

and OR NOT

Input/Output Operations enable interaction between User and Program

Control flow Using Decision-making statements

Control flow statements determine Order in which instructions are Executed in a Program to make decisions based on on Conditions


```
age = int(input("Enter age"))
is3D = 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 is3D == 1:
```

```
    Price += 50
```

```
Print("Final Ticket Price: ₹", Price)
```

2. College Attendance Rule:

```
attendance = float(input("Enter attendance Percentages:"))
```

```
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

```
bill = float(input("Enter bill amount"))
```

```
isPrime = int(input("Is Prime Member? (1=Yes, 0=No):"))
```

```
if bill >= 5000:
```

```
    discount = 0.2
```

```
elif bill >= 2000:
```

```
    discount = 0.10
```

```
else:
```

```
    discount = 0.0
```


4. Battery status:

```
battery = int(input("Enter battery Percentage:"))  
charging = int(input("Is Phone charging? (1=Yes, 0=No)"))
```

```
if charging == 1:  
    Print("charging")
```

Else:

```
if battery <= 20:  
    Print("Low Battery")
```

```
elif battery <= 80:  
    Print("Normal")
```

```
elif battery > 80:  
    Print("Full")
```

5> Rules:

```
if age >= 60:  
    Print("Eligible")
```

```
else if age >= 18 and test Passed == 1:  
    Print("Eligible")
```

```
Else:  
    Print("Not Eligible")
```



```
Delivery  
amount = float(input("Enter Order amount"))  
is Gold = int(input("Are you a gold member? (1=Yes, 0=No)"))  
distance = float(input("Enter delivery distance in km:"))  
if distance > 10:  
    Print("Delivery charged")  
Elif amount >= 500 or is Gold == 1:  
    Print("Free Delivery")  
Else:  
    Print("Delivery charged")
```

7. Bank loan

```
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 = float(input("Enter Units Consumed:"))

if Units \leq 100:

bill = Units * 2

elif Units \leq 200:

bill = 100 * 2 + (Units - 100) * 3

else:

bill = 100 * 2 + 100 * 3 + (Units - 200) * 5

Print(f"Electricity Bill: ₹{bill}")

9. Scholarship

marks = float(input("Enter marks:"))

income = float(input("Enter family income:"))

Single Parent = int(input("Single Parent child? (1=Yes, 0=no):"))

if marks \geq 85 and (Single Parent == 1 or income \leq 500000)

Print("Scholarship Granted")

Else:

Print("Scholarship Not Granted")

10. Online Exam result

theory = int(input("Enter theory marks:"))

Practical = int(input("Enter Practical marks:"))

total = theory + Practical

if (theory \geq 40 and Practical \geq 40) or total \geq 100:

Print("Pass")

Else
Print("Fail")

~~Weekend = int(input("Is it Weekend? (1=Yes, 0=No):"))~~
~~days stayed = int(input("Enter Weekend no. of days stayed"))~~
~~rate = 4000 if isWeekend = 1 else 3000~~
~~bill = rate * days stayed~~
~~if days stayed > 3:~~
~~bill *= 0.05~~

12. Gaming

Score = int(input("Enter Score:"))
 Premium = int(input("Has Premium Pass? (1=Yes, 0=No):"))
 Usedcheat = int(input("Used cheating? (1=Yes, 0=no):"))
 if Usedcheat == 1:
 Print("Access Denied")
 Elif Score >= 100 or isPremium == 1:
 Print("Level Unlocked")
 Else:
 Print("Level Locked")

13. Mobile Data Usage

data = float(input("Enter data :"))
 Unlimited = int(input("Unlimited Plan? (1=Yes, 0=no):"))
 Roaming = int(input("is roaming on? (1=Yes, 0=no):"))
 if Roaming == 1:
 Unlimited = data <= 2
 Else:
 Unlimited = data <= 2 or Unlimited == 1
 if Unlimited:
 Print("Unlimited Data")

Else:

Print ("Data Ltd")

14.

Office System

id = int(input("ID Card Valid? (1=Yes, 0=No): "))

finger = int(input("Finger Print? (1=Yes, 0=No): "))

face = int(input("Face (1=Yes, 0=no): "))

Holiday = int(input("Is it holiday? (1=Yes, 0=no): "))

if holiday == 1:

Print("Entry Denied")

elif id == 1 and (finger == 1 or face == 1):

Print("Entry Granted")

else:

Print("Entry denied")

15. Movie

avg = float(input("Enter avg rating"))

editor = int(input("Is editor choice (1=Yes, 0=no): "))

if editor == 1:

Print("Recommended")

elif avg >= 8.5:

Print("Excellent")

elif avg >= 6.0:

Print("Good")

else:

Print("Average")