

## 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 applications

• 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 Data types and Operators in Python with Examples.

Data type: Data type in Python are a way to classify data items. They represent the kind of value that operations can perform on an object. Since everything is 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)j For Example  $2+3j$   $2+3i$

# Data types

Ordered Collection

Mutable [can change]

Ex:

numbers = [1, 2, 3, 4]

Range

Represents Sequence of numbers  
Commonly Used in loops

Ex: r = range(1, 5)

## 4) Set Data type

Set:

Unordered Collection

No duplicate Values

It store 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": 5<sup>th</sup>}

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, Comparision 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:

<code>+</code>	Addition
<code>-</code>	Subtraction
<code>*</code>	Multiplication
<code>/</code>	Division
<code>%</code>	Modulus
<code>//</code>	Floor division
<code>**</code>	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 Value. returning a Boolean result (True/False) it includes  $==$  (equal),  $\neq$  (Not Equal),  $>$  (greater than),  $<$  (less than),  $\geq$  (greater than or Equal),  $\leq$  (less than or Equal)

Ex:  $5 == 5$  true

$5 \neq 8$  true

## logical membership Operators

Logical Operators (AND, OR, NOT) are to modify boolean Expressions (true/false), decision making Control programming flow:

Membership Operators in and not in . These Operators are Used to test whether Variable is found or not in list, tuple, Dictionary set Come Under membership Operators

Identity Operators  
It includes is and is not . check whether Variable refers to 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 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

Using the split function allows taking multiple inputs in a single line.

Using split:

`split()` divides inputs based on spaces by default.  
The value can be converted to the required 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 separated.

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 variables

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

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

\* Type Conversion is the process of changing a value from one datatype to another.

There are 2 types of type conversion:

1. Implicit
2. Explicit

### Implicit type conversion:

Implicit conversion happens automatically when data types are used together in an expression.

Smaller datatype 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))`

## revision making statements in Python:-

### if-statement

Syntax: if Condition:  
Statement(s)

Ex: age = 18

if age >= 18:

Print("Eligible")

### 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  
Handle 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 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 Purpose of Code & improve readability. Python supports Single line Comments Using # symbol & multi-line Comme

(..) or ("").

### Data types, Operators, & input/Output Operations

Python Supports Various data types such as integer & 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 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 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

bill = float(input("Enter bill amount"))  
isPrime = int(input("Is Prime Number? (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")
    else:
        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"))
isGold = 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 isGold == 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 <= 100:  
 bill = Units \* 2

Elif Units <= 200:  
 bill = 100 \* 2 + (Units - 100) \* 3

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

Print("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 >= 85 and (Single Parent == 1 or income < 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 >= 40 and Practical >= 40) or total >= 100 :

Print("Pass")

Else  
 Print("Fail")

11. Bill:  
Weekend = int(input("Is it Weekend? (1=Yes, 0=No): "))  
days\_stayed = int(input("Enter Weekend no. of days stayed: "))  
bill = 4000 if Weekend == 1 else 3000  
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 Std")

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")