

# PYTHON ASSIGNMENT

Programming and Python:

## Programming:

### Definition

programming can be defined as the process of writing instructions in order to perform particular tasks.

### Purpose of programming

- \* programming helps to automate tasks
- \* Solve complex problems
- \* Makes the work easier
- \* As computer doesn't understand language but understand instructions programming is useful.

### Characteristics of Python

- \* Easy to learn
- \* High level language
- \* Interpreted, Object Oriented
- \* Contains huge libraries, Open source

### Applications of Python

- \* Used in web development, Machine learning
- \* Automation and scripting, Game development
- \* Also used in Data science
- \* Finance and Trading
- \* Educational Tools

### TYPES OF COMMENTS IN PYTHON

1. Single line comment
2. Multi line comments

## Single Line comment

Single line comment can be written by using "#" symbol.

Syntax:

# statement

Example:

# This is python programming.

## Multi line comments

Multi line comments can be written by using three single quotation marks ('').

Syntax:

'''  
    Statement'''

Example:

''' python is the most used  
programming  
language'''

## Importance of python in modern software development

1. python has simple, readable syntax similar to English, so it is easy to use
2. python allows faster application development, used in high productivity.
3. python is used in many fields many fields such as Web development, Data Science and analysis, Artificial Intelligence and machine learning.
4. python has more libraries for Data Analysis like manipulation, visualisation
5. Unlike other programming languages python has more features like Data visualisation, Manipulation etc.

## Data Types and operators:

### Builtin data types in python

#### 1. Numeric Data Type :-

Numeric Data-type is used to define the integer values in python programming.

Example:-

1. marks = 85  
print(marks)

2. price = 4.89  
print(price)

#### 2. Sequence Data Type :-

Sequence Data-type include two types

- 1) List
- 2) Tuple

##### 1) List :-

Lists are ordered, mutable and can store different data types

Example:-

1. colors = ["pink", "red", "yellow"]  
print(colors)

2. mixed\_list = ["Kanya", 21, 7.58]  
print(mixed\_list)

##### 2) Tuple :-

Tuples are ordered, immutable and can store different types of data.

Example:-

1. mobiles = ("Samsung", "Realme", "Redmi")  
print(mobiles)

2. mixed\_tuple = ("Realme", 3, 6.2)  
print(mixed\_tuple)

#### 3. Set Data Type:-

Set datatype is used to store unordered collection of data which doesn't accept duplicate values (Repeated Values), represented in flower braces ({ })

Example:-

1. fruits = {'apple', 'banana', 'grapes'}

print(fruits)

2. numbs = {1, 2, 3, 2, 5, 3, 5, 5}

print(numbs).

#It doesn't show an error but gives the output as follows:

{1, 2, 3, 5}

#### 4. Mapping Data Type:-

Mapping Data Type also known as Dictionaries are represented as key-value pairs, the collection of different types of data like integer, float,

String.

Example:-

1. dict\_1 = {"name": "Siri", "age": 20, "marks": 72}

print(dict\_1)

2. student = {'roll-no': 123, 'name': 'Rahul', 'marks': 85.5}

print(student)

#### 5. Boolean Data TYPE :-

Boolean Data Type represents True or False

Values

\* Used for conditions and comparisons.

Example:-

1. a = 10

b = 10

a == b #Output : True

2. a = 20

b = 15

b >= a #Output : False

3. x = 15

y = 25

y <= x #Output : False

## TYPE IDENTIFICATION USING type():

Type Identification is defined as the identification of the datatype of variables.  
\*It returns the datatype of value / variable

Example:-

1. `x='hello'`

`y=20`

`z=4.8 print(type(x))`

2. `x=[5,"hello",5.2]`

`print(type(x))`

3. `y=(20,"name","marks")`

`print(type(y))`

4. `z={"name":"Kavya","age":21,"Grade":'A'}`

`print(type(z))`

## VARIOUS PYTHON OPERATORS:

python supports various operators like arithmetic, assignment, comparison, logical, membership, identity operators.

### 1. ARITHMETIC OPERATORS:

Arithmetic operators include Addition(+), subtraction(-), multiplication(\*), Division(/), modular division(%), floor division(//).

Addition(+): Addition(+) operator is used to add any two values.

Example:-

`a=2500`

`b=3510`

`print(a+b)`

Subtraction(-): Subtraction (-) operator is used to subtract two values.

Example:-

`a=4100`

`b=3250`

`print(a-b)`

**Multiplication:** Multiplication operator (\*) is used to multiply any two values.

Example:-

$a = 25$

$b = 30$

`print(a*b)`

**Division:** Division (/) operator is used to divide (1)

any two values

Example:-

$x = 25$

$y = 5$

`print(a/b)`

**Modular Division:** Modular division (%) is used (1.)

to get the remainder value for the given input.

Example:-

$a = 20$

$b = 6$

`print(a%b)`

**Floor Division:** Floor Division is used to (11)

get the integer value as output when divided rather than the float values.

Example :-

$a = 20$

$b = 7$

`print(a//b)`

## 2 Assignment Operators :

Assignment operators are used to assign values to the variables. Assignment operators include

1)  $= \rightarrow$  Simple assignment

2)  $+= \rightarrow$  Add and assign

3)  $-= \rightarrow$  Subtract and assign

4)  $*= \rightarrow$  Multiply and assign

5)  $/= \rightarrow$  Divide and assign

6)  $\% = \rightarrow$  Modulus and assign

- \* '=' operator assigns the values to the variables.
- \* '+=' operator first adds the given values and then assign to the variable
- \* '-=' operator first subtracts the given values and then assign to the variable
- \* '\*=' operator first multiplies the given values and then assign to the variable.
- \* '/=' operator divides the given values and then assign to the variable.
- \* '%=' operator does the modular division the result and remainder then is assigned to the variable

Example:-

```
a = 5    # =
print(a)

a += 10  # +=
print(a)  # +a

a -= 2  # -=
a *= 10 # *=
a /= 5  # ./=
a %= 2  # %.=

print(a)
```

### 3. Comparison operators:

These operators are used to compare the two values. Comparison operators are:

- 1.) == (Equal to) → checks whether the given values are equal, and given output will be boolean.
- 2.) != (Not equal to) → checks whether the values are not equal to each other and if they are not equal gives output as True if not False
- 3.) > (Greater than) → checks the value over other if it is greater it gives True if not False
- 4.) < (Less than) → checks the value over other value and if the given value is less than other it gives True if not False
- 5.) >= (Greater than or equal to) → checks the given values are greater than (or) equal to both conditions are satisfied it gives True if not False.

6.) `<=` (less than or equal to) → checks the given values are less than or equal to if both the conditions are satisfied it gives output as True if not False.

Example:-

`x = 5`

`y = 3`

```
print(x > y) # True  
print(x == y) # False  
print(x < y) # False  
print(x >= y) # True  
print(x <= y) # False  
print(x != y) # True
```

#### 4. Logical Operators:

Logical operators are used to combine the conditions, logical operators are:

- 1.) `and` (logical AND) → Both the conditions must be TRUE
- 2.) `or` (logical OR) → Atleast one condition must be TRUE
- 3.) `not` (logical NOT) → Reverses the condition (true becomes false and vice versa)

Example:-

`x = 5`

`y = 3`

```
print(x > 3 and y > 2)  
print(x > 6 or y > 2)  
print(not(x > 6))
```

#### 5. Membership Operators:

Membership operators are used to check the values are present or not present in given set of values. The membership operators are:

- 1.) `in` → checks if a value is present
- 2.) `not in` → checks if a value is not present.

Examples:-

```
fruits = ['apple', 'banana', 'cherry']  
print("banana" in fruits)  
print('orange' not in fruits)
```

## 6. Identity operators:

Identity operators in Python are used to check if two variables refer to the same object.

The identity operators are:

1) is → checks if the two variables refer to the same object.

2) is not → checks if both variables do not refer to the same object.

Examples:-

```
x = [1, 2, 3]
```

```
y = [1, 2, 3]
```

```
z = x
```

```
print(x is z) # True
```

```
print(x is y) # False
```

```
print(x is not y) # True
```

Python Input and Output Operations:

Input Operations:

- \* input() function is used to take user input.
- \* Default data type of input() is string.
- \* Use type conversion functions like int(), float() etc. to convert input to desired type
- \* Take multiple inputs using input().split() and store in variables.

Example:

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

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

```
height = float(input("Enter height: "))
```

# Multiple inputs

```
x, y = input("Enter two values: ").split()
```

Here split function splits a string into a list of substrings based on separator.

Output operations:

- \* print() function is used to display output
- \* Use separators like sep=' ' to separate values
- \* Use format specifiers like %d, .%f, %.s or f-strings for formatted output.

Example:-

```
print("Hello", name, sep = ',')
print(f"Age: {age}, Height: {height:.2f}")
```

## Control statements and Decision-making statements

### Control statements:

Control statements determine the flow of a program's execution. They're crucial for writing dynamic, flexible and efficient.

### Types of control statements:

1. Decision-Making statements: if, if-else, if-elif-else
2. Looping statements: for, while
3. Jump statements: break, continue, pass.

### Decision-making statements:

#### if statement

\* Used for single condition check

\* Syntax: if condition:  
              statement.

#### Example:

```
x=5
if x>3:
    print("x is greater than 3")
```

#### if-else statement

\* Used for checking two decisions (conditions)

\* Syntax: if condition:  
              statement

```
    else:
        statement
```

#### Example:

```
x=5
if x>10:
    print("x is greater than 10")
else:
    print("x is less than 10")
```

**if-elif-else statement**

\* Used for multiple decision making

\* Syntax: if condition:

    statement

elif condition:

    statement

else:

    statement

Example:-

x=5

if x>10:

    print("x is greater than 10")

elif x==5:

    print("x is equal to 5")

else:

    print("x is less than 5")

Syntax and execution control:-

1. Conditions are evaluated from top to bottom

2. If a condition is true, the corresponding block is executed, and control jumps out of if-elif-else structure.

3. If no condition is true, the else block is executed (if present).

## Python Programming Fundamentals

Python is widely-used programming language that plays a vital role in solving complex problems. Programming is about breaking down problems into sub parts and writing code to execute specific tasks. Python's simplicity, readability, and extensive libraries make it an ideal language for beginners and also for experts.

**Role of programming in problem Solving**  
Programming is a powerful tool for solving real-world problems easily. By writing code, developers can automate tasks, analyze data and create new solutions. Python's syntax is designed to be intuitive and easy to learn, making it an excellent choice for problem solving.

## Python Syntax Simplicity and Readability

python's syntax is simple and concise, focusing on readability. Indentation defines block-level structure, eliminating the need for explicit delimiters. This makes python code easy to write, read and maintain.

\*In python the need for explicit delimiters like { } - or ; is eliminated due to its unique syntax design.

Example:

```
name = input("Enter your name:")
print(f"Hello, {name}!")
```

Use of comments for code Documentation.

Comments are essential for documenting code, explaining logic, and making it understandable to others. Python uses the # symbol for single-line comment and three single quotes for multi-line comments.

Example:

```
#This is a single-line comment
```

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

```

Data types, operators , input/output operations  
Python has various built-in data types, including:

- \* integers : int
- \* floats : float
- \* strings : str
- \* lists : list
- \* tuples : tuple

Operations are used to perform operations on data such as arithmetic, comparison, logical, assignment and identity operators.

\*Input/output operations are performed using input() and print() functions respectively.

**Example:-**

```
x = int(input("Enter a number:"))
y = int(input("Enter another number:"))
print(f"sum : {x+y}")
```

### Control Flow using Decision-Making Statements

Decision-making statements control the flow of a program's execution. Python's if, if-else, and if-elif-else statements enable developers to write conditional code.

**Examples:-**

```
x=5
if x>2:
    print("x is greater than 2")
elif x==5:
    print("x is equal to 5")
else:
    print("x is less than 2")
```

**Conclusion:-**

Python's simplicity, readability, and extensive libraries make it an ideal language for problem solving and development. Understanding Python's fundamentals, including datatypes, operators, input/output operations, and control flow, is essential for building robust and efficient applications.

## Real-world problems using Python programming

### 1. Movie ticket pricing

A movie theatre charges:

```
age = int(input("Enter age:"))
is3D = int(input("Is 3D movie? (1 for yes, 0 for no):"))

if age < 13:
    price = 150
elif 13 <= age < 60:
    price = 250
else:
    price = 200

if is3D == 1:
    price += 50

print(f"Final ticket price: rupees: {price}")
```

### 2. College Attendance Rule

A student is allowed to write the exam if:

```
att_per = int(input("Enter attendance percentage:"))
med_cer = int(input("medical certificate? (1 for yes, 0 for no):"))

if att_per >= 75:
    print("Allowed")
```

elif att\_per >= 60 and med\_cer == 1:

```
    print("Allowed")
```

else:

```
    print("Not Allowed")
```

### 3. E-commerce Discount

A shopping site gives:

```
bill_amt = int(input("Enter bill amount:"))
isprime = int(input("Is prime customer (+10):"))
if bill_amt >= 5000:
    discount = 20
elif bill_amt >= 2000:
    discount = 10
else:
    discount = 0
if isprime == 1:
    discount += 5
final_amount = bill_amt * (bill_amt * discount / 100)
print("Final amount:", final_amount)
```

### 4. Smartphone Battery warning

A phone shows:

```
battery_percentage = int(input("Enter battery percentage:"))
is_charging = int(input("Is charging (1 or 0):"))
if battery_percentage <= 20:
    print("Low Battery")
elif battery_percentage >= 21 and battery_percentage <= 80:
    print("Normal Battery")
elif battery_percentage > 80:
    print("Full Battery")
if is_charging == 1:
    print("charging")
```

### 5. Driving License check.

A person can get a driving license if:

```
age = int(input("Enter age:"))
tp = int(input("Passed driving test(1 for yes):"))

if age >= 18 and tp == 1:
    print("Eligible")
elif age >= 60:
    print("Driving test is not required")
else:
    print("Not Eligible")
```

### 6. Online Food Delivery

A restaurant gives free delivery if:

```
orderamount = int(input("Enter amount:"))
isgold = int(input("Is Gold member (1 or 0):"))

distance = int(input("Enter the distance:"))

if orderamount >= 500 or isgold == 1:
    print("Free Delivery")
elif distance > 10:
    print("Delivery is never free")
else:
    print("Delivery charged")
```

### 7. Bank Loan Approval

A bank approves loan if:

```
sal = int(input("Enter salary:"))
creditscore = int(input("Enter credit score:"))

if sal >= 30000 and creditscore >= 700:
    print("Loan Approved")
elif sal >= 50000:
    print("Credit score ignored")
else:
    print("Loan Rejected")
```

### 8. Electricity Bill

Units consumed:

```
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: RS", bill)
```

### 9. Student Scholarship

A student gets a scholarship if:

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

```
income = int(input("Enter income:"))
```

```
isSingleParent = int(input("Is single parent child (1 or 0):"))
```

```
if marks >= 85 and income < 500000:
```

```
    print("Gets scholarship")
```

```
if isSingleParent == 1:
```

```
    print("Income condition is ignored")
```

```
else:
```

```
    print("Does not get scholarship")
```

### 10. Online Exam Result

A student passes if:

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

```
practical = int(input("Enter practical exam marks:"))
```

```
if (theory >= 40 and practical >= 40) or (theory + practical  
                                            >= 100):
```

```
    print("Result: pass")
```

```
else:
```

```
    print("Result: fail")
```

## 11. Hotel Room Pricing

A hotel charges:

isWeekend = int(input("Is weekend stay? (1 for yes,  
0 for no):"))  
daysStayed = int(input("Enter number of days  
stayed:"))

if isWeekend == 1:  
 price = 4000

else:  
 price = 3000

total = price \* daysStayed

if daysStayed > 3:  
 total \*= 0.85 # 15% discount.

print(f"Final bill: RS {total}")

## 12. Gaming Level Unlocks

A game unlocks next level if:

score = int(input("Enter score:"))  
isPremium = int(input("Do you have a premium  
pass? (1 or 0):"))

usedCheat = int(input("Did you use cheating?  
(1 or 0):"))

if usedCheat == 1:  
 print("Access denied due to cheating!")

elif score >= 100 or isPremium == 1:

print("Next level unlocked!")

else:

print("Level locked. Try again!")

### 13. Mobile Data Usage

A Network gives unlimited data if:

```
dataUsed = float(input("Enter daily data used (inGB):"))
```

```
hasUnlimitedPlan = int(input("Do you have an unlimited plan?(1 or 0):"))
```

```
isRoaming = int(input("Is roaming on?(1 or 0):"))
```

```
if isRoaming == 1 and hasUnlimitedPlan == 1:
```

```
    unlimited_works = 0
```

else:

```
    unlimited_works = hasUnlimitedPlan
```

```
if dataUsed <= 2 or unlimited_works == 1:
```

```
    print("Unlimited data applies: ")
```

else:

```
    print("Data limits apply: ")
```

### 14. Office Entry System

An employee can enter the office if:

```
idValid = int(input("Is ID card valid?(1 or 0):"))
```

```
fingerprint = int(input("Fingerprint match?(1 or 0):"))
```

```
faceScan = int(input("Face scan match?(1 or 0):"))
```

```
isHoliday = int(input("Is it a holiday?(1 or 0):"))
```

```
if isHoliday == 1:
```

```
    print("Entry denied. Holiday today.")
```

```
elif idValid == 1 and (fingerprint == 1 or faceScan == 1):
```

```
    print("Entry allowed.")
```

else:

```
    print("Entry denied.")
```

### 15. Movie Rating Display

movie app shows rating based on average scores

```
averageRating = float(input("Enter average rating:"))
isEditorsChoice = int(input("Is it Editor's choice? (1 or 0):"))

if isEditorsChoice == 1:
    print("Recommended")
elif averageRating >= 8.5:
    print("Excellent")
elif 6.0 <= averageRating < 8.5:
    print("Good")
else:
    print("Average")
```