*Types of Comments in Python with Syntax:-

Comments in Python:-

- Comments are non-Executable lines used for Explanation
- Comments are ignored by Python
- Used to Explain code
- makes program readable and understandable.

Python Supports two types of Comments:-

1. Single-line Comment
2. Multi-line comment

Single-Line Comment:-

Ex:

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

Multi-Line Comment:-

Ex:

```
'''
This is multi-line
Comment.
```

* Definition & Purpose of Programming:-

Define:-

Programming is the process of designing & writting a set of instructions that a computer follows to perform a task.

Purpose of Programming:-

- computer cannot think on their own
- They only understand instructions
- To process and analyze data
- Program helps automate tasks & solve problems.

* CHARACTERISTICS AND APPLICATIONS OF PYTHON:-

CHARACTERISTICS:-

- Simple & Easy to Learn
- Interpreted language
- Platform independent
- open source
- object-Oriented

Applications:-

- Web development
- Data analysis & Science
- Artificial intelligence
- Machine learning
- Automation
- Gene development
- numerical computing

*Importance of Python in Modern Software Engineering:-

- Python increase developer productivity due to simple Syntax
- Widely used in web development AI, ML and data Science.
- Strong community Support and large number of libraries.
- Easily integrates with other language like C, C++ & Java
- used to top companies like Google, Netflix & Instagram.

2. Describe Data Types And Operators In Pyth

With Suitable Examples:-

A. Built-in Data Types In Python:-

1. Numeric Data Type:-

used to store Numerical values

- int - whole numbers
- float - decimal numbers
- complex - complex numbers

Ex:

```
a = 10        # int
b = 3.5       # float
c = 2+3j      # complex
```

2. Sequence Data Type:-

used to Store ordered collections.

- List - mutable
- Tuple - immutable
- String - Sequence of characters

Ex:-

```
list1 = [1, 2, 3]
tuple1 = (4, 5, 6)
name = Python
```

3. Set Data type :-

used to Store unordered and unique Elements.

Ex:

Set 1 = {1, 2, 3, 3}

4. Mapping Data type :-

Stores data in key-value pairs

- Dictionary

Ex :-

Student = {"name": "Thanusha", "age": 20}

5. Boolean Data type :-

Stores only true (or) False.

Ex:

x = True
Y = False

VARIOUS TYPES OF PYTHON OPERATORS :-

Types :-

1. Arithemetic operators
2. Assigment operators
3. Comparison operators.

4. logical operators
5. Bitwise operators
6. membership operators
7. Identify operators:

* Arithmetic operators:-

- Arithmetic operators are used to perform mathematical calculations.
- They work with numbers like integers and floats.
- used for mathematical calculations.

operator	meaning
+	Addition
-	Sub
*	multi°
/	Division
%	modulus
//	Floor division
* *	Exponent

* Comparison operators:-

- used to Compare 2-values
- comparision operators are used to compare two values.
- (They are commonly used in)
- the result is always a boolean value (T(or)f)

operator	meaning
==	Equal to
!=	Not Equal
>	Greater than
<	less than
>=	Greater than (or) Equal
<=	less than (or) Equal

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

Logical operators:-

- Logical operator are used to combine multiple conditions
- they are commonly used in decision-making statements.
- used to combine conditions

operator	meaning
and	True if both are True
or	True if any one is true
not	Reverse the result

```
a = 10
b = 20
Print (a<b & b>15)
```

Assignment operators:-

- Assignment operators are used to assign (or) update values in a variables

- They combine arithematic operation with assignment.
- used to assign values to variables.

operator	Examples
=	x = 5
+=	x += 3
-=	x -= 2
*=	x *= 2
/=	x /= 2

```
x = 10
x += 5
Print(x)
```

membership operators:-

- membership operators check whether a values exists in a sequence.
- used with lists, strings, tuples and sets
- Check whether a value exists in a sequence.

operator	meaning
in	present
not in	Not present

```
list 1=(1,2,3)
Print (2 in list)
Print ('on not in 'python')
```

Identity operators:-

- Identity operators check whether two variables refer to the same memory object.
- They do not compare values, but object identity
- check whether two variables refer to same objects.

operator	meaning
is	Same object
is not	Different

a=10
b=10

D. Real-World Usage of Operators:-

- Arithmetic operators - calculating marks, salary, interest
- comparsion operators - Eligibility checking (age >18)
- logical operators - login authentication System
- membership operators - searching data in lists (or) databases
- Assignment operators - updating values in programs

~~Python~~ Conversion while Taking Input

PYTHON / INPUT & 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 IN INPUT:-

Since input is always string, conversion is required

Enterger input

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

Float input

```
Price = Float(input("Enter price:")
Print(price)
```

TAKING MUITIPLE INPUTS:-

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

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

* FORMATTED OUTPUT:-

using F- String (Recommended)

```
name = "Thanushree"
age = 20
Print ( f "name: {name}, Age: {age}")
```

Input & output with Expression

```
a = int (input ('Enter number:"))
Print ("Sequence :", a*a)
```

CONTROL STATEMENT & DECISION MAKING STATEMENT IN PYTHON:-

CONTROL STATEMENT:-

- control Statement are used to control the flow of Execution of a python program.
- using control Statement we can write decision-making looping and flow-altering programs.

Type of control Statement in python:-

3 types:- * Decision making
* looping
* Control tranfer Statement

Importance:-

=> Enables decision making
=> makes programs logical & dynamic
=> Avoids unnecessary execution.

DECISION-MAKING STATEMENTS:-

1. If-Statement:-

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

Syntax

```
if condition :
    Statement
```

Example

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

2. If-else Statement:-

Executive one block if condition is true.

otherwise Executive another block.

Syntax

```
if condition:
    Statement1
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 ("First class")
elif marks >=35
    Print ("pass")
Else:
    Print ("fail")
```

Execution Flow:-

- Conditions are checked top to bottom
- First true condition is Executed
- Remaining condition are skipped

Example: (If -Else Execution Flow)

- Condition age >=18 is checked
- condition is False
- Else block Executes

4. Program continues after if-else
Python execute only the block whose conditions is tru
Indentation plays a crucial role in controlling flow

WRITE AN ESSAY ON PYTHON PROGRAMMING FUNDAMENTA

1. Role of Programming in problem Solving
=> Progamming helps in Solving real-world problems usi
logical steps
=> It breaks complex problems into smaller, managabl
tasks
=> Enables automation of repetitive work
=> Helps in accurate calculations and data processing
=> Improves Efficiency & saves time.

2. Python Syntax Simplicity & Readability:-
=> Python has simple & English like Syntax.
=> Easy to learn for beginners
=> Does not use brackets { } (or) Semicolons',
=> uses indentation to define code blocks

3. use of Comments for Code Documentation:-
=> Comments Explain the purpose of code statements.
=> Improve readability & understanding of programs.
=> Helpful for debugging & future modifications
=> Single-line comments use #
=> multi-line comments use triple Quotes (''' or """)

4. Data types, operators and Input/output operations:-

=> Python supports built-in data type like int, float, str & bool.

=> operators perform arithmetic, comparision, logical & 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 presentation.

5. Control flow using Decision-making statements

=> control flow decides the order of execution in a program

=> Decision-making statements help in making logical choices

=> If statement executes code when condition is true.

=> If-else provides two-way decision making

=> If-Elif-else handles multiple conditions

=> makes programs flexible & dynamic.

Conclusion:-

=> python fundamentals form the base for advanced programing

=> Simple Syntax & powerful features make a python popular.

=> widely used in Education, Science, data analysis and Software development.

(1) (a) making Ticket pricing

A movie theatre charges:

₹150 for children (age <13)
₹250 for adults (age 13-59)
₹200 for seniors (age ≥60)

If the person is watching a 3D movie, add ₹50 extra

write a program that takes age and is 3D (1 or 0)

& prints 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 wr

the exam if;

attendance ≥ 75

(or)

attendence ≥60 and has medical artificate (1=yes, 0=no)
Take attendance percentage and medical certificate as input & print "Allowed (or) not Allowed"

```
attendance = int (input ("Enter attendance percentage:"))
medical = int (input ("medical certificate (1 for yes, 0 for no):"))
if attendance >= 75 or (attendance >=60 and medical==1):
    Print ("Allowed")
Else:
    Print ("Not Allowed")
```

E. Commers Discount

A Shopping Site gives

20% discount if bill ≥ 5000

10% discount if bill between 2000 & 4999

No discount if bill < 2000

But if the customer is a prime number they get extra 5% discount

Input : bill amount ; is prime (1 or 0)

Print final ~~input~~ amount to be paid

```
bill = float (input ("Enter bill amount:"))
if bill >= 5000:
    discount = 0.20
elif bill = 2000:
    discount = 0.10
```

```
Else
    discount = 0.0
ie.. Prime = int(input("Is the customer a prime member? (1 for yes, 0 f
if is prime == 1:
  discount += 0.05
final amount = bill - (bill * discount)
Print ("Final amount to be paid : ₹", final-amount)
```

(H) Smart phone Battery warning

A phone 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 : "))
is charging = int(input("Is the device charging ? (1 for yes, 0 f
if  is charging == 1:
      Print ("charging")
Else if battery <= 20:
      Print ("Low Battery")
```

Driving License check :-

A person can get a driving license if:

age ≥ 18
AND
Passed driving 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")
```

---

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 10km, delivery is never free

Input : amount is Gold (1 or 0) 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 isGold == 1:
    Print ("Free Delivery")
Else:
    Print ("Delivery charge")
```

(7) Bank Loan Approval:-

A bank approves a loan if:
Salary ≥ 30,000 AND credit score ≥ 700
OR
Salary ≥ 50,000 (credit score ignored)
Input: salary, creditscore
Print "Loan Approved" or "Loan Rejected"

```
Salary = int(input("Enter Salary:"))
Creditscore = int(input("Enter credit Score:"))
if  Salary >= 50000 (or) (salary >= 30000 and credit Score >= 700):
    Print ("Loan Approved")
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("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 Amount: ₹", bill)
```

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, single parent (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")
```

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 : marks, income, single parent (1 or 0)

```
theory = int(input("enter theory marks:"))
Practical = int(input("enter practical marks:"))
if (theory >=40 and practical>40) and (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 : is weekend (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
```

Gaming level unlock:-

A 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("Enter score:"))
is premium = int(input("Is premium player? (1 for yes, 0 for no):"))
used cheat = int(input("used cheating? (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")
```

Mobile Data usage:

A network gives unlimited data if:

Daily usage ≤ 2GB

OR

User has unlimited plan

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

Input: data used, has unlimited plan, is Roaming

```
data_used = float(input("Enter daily data usage (GB): "))
has_unlimited_plan = int(input("Has unlimited plan? (1 for yes, 0 for No): "
is_Roaming = int(input("is Roaming on? (1 for yes, 0 for No): "))

if is_Roaming == 1:
    if data_used <= 2:
        print("unlimited Data")
    else:
        print("limited Data")

else:
    if data_used <= 2 or has_unlimited_plan == 1:
        print("unlimited Data")
    else:
        print("limited Data")
```

(14) Office Entry System:

An Employee can Enter the office if:
ID card is valid
AND
(finger print matches (OR) Face Scan matches)
But if it is a holiday, Entry is denied for everyone
Input: id_valid, finger_print, face_scan, is_holiday

```
id_valid = int(input("ID card valid? (1 for yes, 0 for No): "))
finger_print = int(input("Finger print match? (1 for yes, 0 for 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:
    Point("Entry Denied")
```

```
Elif id valid == 1 and (finger print == 1 or face Scan == 1):
    Print ("Entry Allowed")
Else:
    Print ("Entry Denied")
```

---

movie Rating Display :-

A 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 marked as editor's choice always show "Recommended"

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

Print the message

```
average Rating = float(input("Enter average rating :"))
is Editors Choice = int(input("is Editor's choice ?(1 for yes, 0 for no) :"))
if is Editors Choice == 1:
    Print ("Recommended")
Elif average Rating >= 8.5:
    Print ("Excellent")
Elif averageRating >= 6.0:
    Print ("Good")
Else:
    print ("Average")
```