DAY 2 | PYTHON BASICS FOR BEGINNER

↳ A simple introduction to Python Prog.

→ INTRODUCTION TO PROGRAMMING

Step1: Understand the Problem. 1st step in solving any pbm, algorithmic or otherwise, is to understand the pbm thoroughly..

Step 2: Design the Algorithm

step 3: Implement the Algo.

step 4: Test the Algo.

step 5: Analyze the Algo.

→ Developing an algorithm

⇒ How to create an Algo.

- ↳ Analyze the pbm
- ↳ Design the algo
- ↳ Select appropriate tools and technologies
- ↳ Implement the algo.
- ↳ Test the algo
- ↳ optimize the algo → (T.C. or S.C.)
- ↳ Document the algo. (comment).

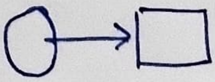
→ FlowChart AND PSEUDOCODE

↳ graphical or symbolic representation

* Common Flowchart symbols

↳ have some standard symbols → read and understood by wider grp of people.

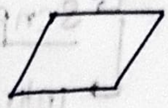
→ Terminal — start or End → Oval

↳ Flow Line → arrow form 1 symbol to othr to show process order 

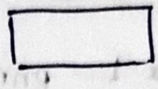
operation — flow of execution

- Universal
- Trace the path
- debugging
- flow of execution.

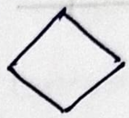
- → Input / Output → (input or output of data] → Rhomboid. parallelogram



- → Process → operation that manipulates data. → Rectangle



- → Decision making → Conditional operation to determine direction → Rhombus



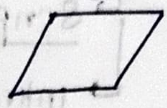
Example

```
Algo1: Add 2 nos. entered by the user
  S1: start
  S2: Input (Read) 1st num. A
  S3: Input (Read) 2nd num. B
  S4: Add A & B and assign the result to sum ← A+B
  S5: Display Sum
  S6: Stop
```

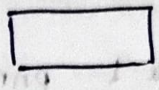
INTRODUCTION TO PYTHON :

- What is Python?
  - Popular programming lang. → Guido van Rossum – 1991 (release)
- Used For :
  - Web dev (server side)
  - software dev.
  - maths
  - system scripting
- What can python do ?
  - used on a server to create web apps
  - used alongside software to create workflow
- Why Python?
  - Platform friendly (Windows, Mac, Linux, Raspberry Pi, etc) (open source)
  - simple syntax to Eng lan
  - Libraries & modules

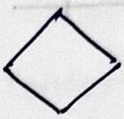
- → Input / Output – (input or output of data] → Rhomboid parallelogram



- → Process → operation that manipulates data. → Rectangle



- → Decision making – Conditional operation to determine direction → Rhombus



<u>Example</u>

Algo1: Add 2 nos. entered by the user

S1: start

S2: Input (Read) 1st num. A

S3: Input (Read) 2nd num. B

S4: Add A & B and assign the result to sum ← A+B

S5: Display Sum

S6: Stop

INTRODUCTION TO PYTHON :

- What is Python?
  - Popular programming lang. – Guido van Rossum – 1991 (released)
- Used For :
  - web dev (server-side)
  - software dev.
  - maths
  - system scripting
- What can python do?
  - used on a server to create web apps
  - used alongside software to create workflow
- → Why Python?
  - Platform friendly (Windows, Mac, Linux, Raspberry Pi, etc) (opensource)
  - simple syntax to eng lan
  - Libraries & modules

→ Interactive mode and script mode

- Script Mode → write your code in a .py file and run it
- Interactive Mode → type your code into python interpreter directly

Example 1

```
print(1)
x=2
print(x)
```

Interactive mode

```
>>> 1 } output
    2
```

Interactive mode	script mode
typing commands & expressions at the prompt	read & execute statements in a script
cant save & edit the code	can save and edit
expt. with the code	clear about code
can't save the statement for further use and we have to retype all the statements ~~to rerun them~~	we can save the statements for further use and we not need to retype all the statement to re-run them
see results immediately	cant see the output immediately

Indentation in Python

- A block is a group of statements in a program or script.
- Python do not use braces ({}) to ~~use~~ indicate block of code for class and function defn or flow ctrl.
- In python, Blocks of code are denoted by line indent.
- Python prog. get structured thru. indentation, i.e. code blocks are defined by their indentation.

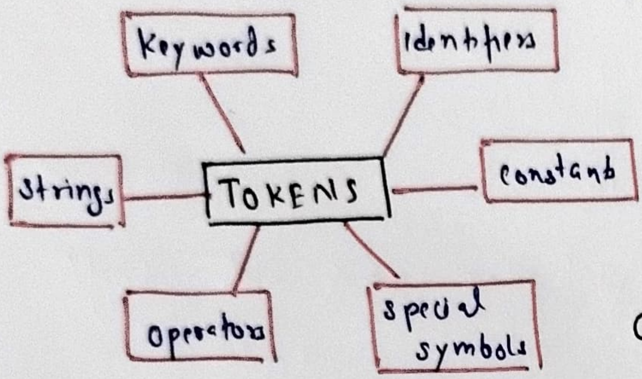
Comments → Non-executable statmernts in a prog.

↳ Comments make the prog. easily readable &understandable by the programmer as well as other users

#

TOKENS IN PYTHON : → smallest individual unit

L All statement & instructions in a prog. are built with tokens.



VARIABLES → containers for storing data

Creating Variable : No command for declaring a variable.

directly assign a value to it.

```
E.g. ·  x=5
        y="John"
        pirint (x)  —  5
        print (y)   —  "John"
```

LITERALS in python

L Fixed values - stored in source code of prog. and do not change during execution

L used to represent data that should not be altered by users, such as software e operating parameter.

CODE :

```
# String Literals
   a= "Hello"
   b=" students"
   c=" This is a learning platform"

# Driver code
   print (a)
   print (b)
   print (c)
```

First Python Program

code : print ("Hello World!")

- print () - display output on screen

Variables & Data Types

→ store data in memory using variable

→ common types : int, float, str, bool

→ e.g. name='Alice' , age = 25

Python Syntax Basics

↳ Indentation defines code block

L No semi colons or braces needed

L # for single line comment

''' ''' for multiline comment

or """ """

Operators

- Arithmetic : +, -, *, /, %
- Comparison : ==, !=, >, <
- Logical : and, or, not