Page No.:
Date: 10/9/2025

History of Python

- 1980s (Late) - Guido van Rossum, a Dutch programmer at Centrum Wiskunde & Informatica in Netherlands started working on scripting language as a hobby project.

- The name Python was inspired by the British comedy group "Monty Python's Flying Circus", which guido loved.

Release notes :-

- Feb 1991 Python 1 release
- Oct 2000 Python 2 release
- Dec 2008 Python 3 release

Note :- Even though there are many new/simple programming exists today, C/C++ languages present and they are in boom why because OS are built using C language / program.

Key feature of Python :-

- Simple and easy to learn.
  ↳ It is similar to English language, making it beginner-friendly.

- Open source and free.
  ↳ It is free to download and use.
  ↳ Source code is open, supported by Python Software Foundation.

- Interpreted language.
  - ↳ Python code is executed line by line by the interpreter
  - ↳ No need of compilation (like c/c++)

- High level language.
  - ↳ Programmers no need to manage memory or low-level details.
  - ↳ Focus on solving problems rather than system internals.

- Platform independent (cross-platform) and more
  - ↳ Python runs on Windows, Linux, Mac OS without ~~more~~ code changes.

- Object-oriented programming
  - ↳ supports classes, ~~inter~~ inheritance, functional and procedural programming.

- Extensive libraries and framework
  - ↳ comes with a large standard library (math, os, datetime etc)
  - ↳ Rich ecosystem of external libraries.

- Dynamically typed.
  - ↳ No need to declare the variables in the start.

- Automatic memory management
  - Built in garbage collector handles memory allocation and cleanup

- Embeddable and extensible.
  - ↳ Python can be embedded in other languages.
  - ↳ We can also call c/c++ code from python

- Large community support.
  - ↳ Huge global community, active forums, tutorials.
  - ↳ Easy to find help, libraries and solutions

- Versatile and multi-purpose
  - ↳ Used in Web development (Django, Flask, FastAPI)
  - ↳ Data science & AI/ML, Deep learning
  - ↳ Automation and scripting
  - ↳ Gaming development
  - ↳ Cybersecurity, IOT, Desktop Apps
  - ↳ Web/mobile application.
  - ↳ GUI
  - ↳ Graphic design.

Note :-
- Python is a high-level, interpreted, object-oriented, general purpose programming language that is easy to learn and use.
- created by Guido van Rossum in 1991.

Comments :-
- They are the part of code but they won't get execute at the time of execution.
- They are used to explain code, make it readable or temporarily disable code.
- They improve code maintainability and teamwork.

Types of comments :-

- Single line comment
  ↳ Starts with # (hash)

eg:- # single line comment
print ("Hi")

- Multi line comments
  ↳ Python doesn't have a dedicated multi-line comment syntax.
  ↳ But we use :
  ↳ Multiple # lines
  ↳ or tripple qoutes (''' or """) as a workaround.

Keywords :-

They are the reserved words used for particular task and they cannot be used as identifiers (variable, function name)

eg:- True, False, for, while, if, def, else, return etc.

Categories of keywords :-

- Control flow
  ↳ if
  ↳ else
  ↳ elif
  ↳ for
  ↳ while
  ↳ break
  ↳ continue
  ↳ pass

- Boolean and logic
  ↳ True ↳ False ↳ AND ↳ OR ↳ NOT

- Functions and classes
  ↳ def ↳ class ↳ Return ↳ Lambda ↳ yield

- Variable handling
  ↳ global ↳ nonlocal ↳ del

- Exception handling
  ↳ try ↳ except ↳ finally ↳ raise ↳ assert

- Others
  ↳ Import ↳ from ↳ with ↳ as ↳ is, ↳ in
  ↳ None.

Variable :

- It is place/name where we can store any value.
- A variable is like a container (a name) used to store data in python.
- It points to a value stored in memory, and we can use that name to access or modify the value.

eg:- X = 10 # Integer value
name = Jai # String value
pi = 3.14 # float value

X, name, pi → variables.

- Rules for creating variables in python.

→ Starts with a letters or underscore (_)
age ✓ name ✓ _name ✗ @name ✗

→ Can contain letters, digits and underscore
Stud_name ✓ Stud-name ✗ stu name ✗

→ Case-sensitive
age, Age, AGE → These are 3 different variables.

→ Cannot use keywords as variables.
if = 5 ✗ for = 6 ✗

→ No special characters allowed except underscore
total$ ✗ name! ✗

→ Can be of any length.

Initialization :
• Initialization means assigning an initial value to a variable at the time of its creation.
• Ways to initialize variable :

→ Single variable Initialization
→ Multiple variable Initialization ⇒ same value / different value
→ Initialization with ~~var~~ Expression
→ Initialization using Input
→ Initialization using data structures

Single variable Initialization :-
• Assigning a value directly.
eg: a = 5
b = "Hello"
c = 3.14

Multiple variable Initialization :-
(Different values)
eg: Assigning multiple variables in a single line.
eg:- x, y, z = 10, 20, 30
print (x, y, z) # o/p = 10, 20, 30

(same variable)
Assigning same value to multiple variables at once.
a = b = c = 0
print (a, b, c) # o/p: 0 0 0

Initialization with expression :-
You can assign a value based on an expression.
eg: x = 5+10
y = x * 2
print (y) # o/p :- 30

Initialization using Input :-
Getting value from user i/p.
name = input ("Enter your name")
age = int(input ("Enter your age"))

Initialization with data structure.
You can initialize variables with list, tuples, dictionaries, set etc.
eg: fruits = ["apple", "mango", "banana"] # list
points = (10, 20) # tuple
students = {"name": "Chaitra", "age": 25} # dictionary

11/9/2025
Data types :-
• Data types define the type of value a variable can hold.
Built-in data types :-

1) Numeric types :-

- int → Integers numbers
- float → Decimal (floating-point) numbers
- complex → complex number (a+bj)

eg:- x = 10
y = 2.8
z = 2 + 5j

2) Text type -
- str → String (sequence of characters)

eg:- name = "PowerBI"

3) Sequence types -
- list → ordered, mutable collection.
Mutable collection can be changed (modified after it is created.

eg:- fruits = ["apple", "mango", "banana"]

- tuple → ordered, immutable collection.

eg:- coordinates = (10, 20)

- range → sequence of numbers (used in loops)

eg:- nums = range(5)

4) Set types -
- set - Unordered, unique elements

eg:- a = {1, 2, 3, 3} # set = {1, 2, 3}

- frozenset - Immutable set

eg:- b = frozenset ([4, 5, 6])

5) Mapping Type -
- dict - key-value pairs

eg:- student = {"name": "chaitra", "age": 25}

6) Boolean Type -
- bool → True or False

eg:- is_active = True
is_passed = False

7) Binary types -
- bytes - Immutable seq of bytes

eg:- b = b"Hello"

- bytearray → Mutable seq of bytes

eg:- arr = bytearray ([65, 66])

- memoryview → View object of bytes

eg:- mv = memoryview (b"Hello")

Checking data type:-
Use type() function:

eg:- x = 10
print (type(x)) # class 'int'

12/9/2025
Example programs:-

1) print ("Hi good morning")

o/p - Hi good morning