

Introduction to Python

What is Python?

Python is a high-level, interpreted, general-purpose programming language.

It was developed by:

- Guido van Rossum

Released in:

- 1991

Python is designed to be:

- Simple
- Readable
- Easy to learn
- Easy to write

Unlike low-level programming languages, Python uses simple English-like syntax, which makes it beginner friendly.

Meaning of “High-Level Language”

A high-level language is a language that is easy for humans to understand.

Example:

- Python uses simple words like:
 - print
 - if
 - for

Low-level languages are closer to machine language and are harder to understand.

Meaning of “Interpreted Language”

Python is an interpreted language.

This means:

- Code executes line by line
- No separate compilation process is required

Advantages

- Easy debugging

- Faster testing
 - Beginner friendly
-

Meaning of “General-Purpose Language”

Python can be used in many fields:

- Web development
 - Artificial Intelligence
 - Machine Learning
 - Automation
 - Cybersecurity
 - Data Science
 - Business Analytics
 - Game development
-

Why Python is Popular

1. Simple Syntax

Python syntax is very easy compared to languages like C or Java.

Example:

```
print("Hello")
```

2. Less Code

Python programs require fewer lines of code.

3. Easy to Learn

Beginners can understand Python quickly.

4. Huge Community Support

Millions of developers use Python worldwide.

5. Large Libraries

Python has many built-in libraries.

Examples:

- NumPy
- Pandas
- Matplotlib

These libraries reduce coding effort.

Applications of Python

1. Web Development

Python is used to create websites and web applications.

Frameworks:

- Django
- Flask

Examples:

- Instagram
 - Spotify
-

2. Data Science

Python analyzes large amounts of data.

Used for:

- Data cleaning
 - Data visualization
 - Predictions
-

3. Artificial Intelligence

Python is widely used in AI applications like:

- Chatbots
 - Image recognition
 - Voice assistants
-

4. Automation

Python automates repetitive tasks.

Example:

- Sending emails automatically
 - File management
-

5. Cybersecurity

Python is used for:

- Ethical hacking
 - Security testing
 - Network analysis
-

Features of Python

1. Simple and Readable

Python code looks similar to English.

Example:

```
print("Welcome")
```

Readable code is easier to:

- Understand
 - Debug
 - Maintain
-

2. Platform Independent

Python programs run on:

- Windows
- Linux
- Mac

Without changing code.

3. Open Source

Python is free to use.

Anyone can:

- Download

- Modify
 - Share
-

4. Dynamically Typed

No need to specify data type.

Example:

```
x = 10  
name = "Python"
```

Python automatically identifies type.

5. Object-Oriented

Python supports Object-Oriented Programming (OOP).

Concepts:

- Class
 - Object
 - Inheritance
-

Python Syntax

What is Syntax?

Syntax means:

- Rules for writing programs

Every programming language has syntax rules.

Python syntax is simple.

Importance of Indentation

Python uses indentation instead of braces {}.

Indentation means spaces before code.

Example:

```
if 5 > 2:  
    print("Correct")
```

Why Indentation is Important

Indentation shows:

- Which statements belong together

Wrong indentation causes errors.

Comments in Python

What are Comments?

Comments are notes inside code.

Python ignores comments during execution.

Used for:

- Explanation
 - Documentation
 - Better understanding
-

Types of Comments

Single-Line Comment

Uses #

This is comment

Multi-Line Comment

Uses triple quotes.

```
"""
```

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

inline comment

```
Print("hello) # reminder
```

Variables in Python

What is Variable?

A variable is a container used to store data.

Example:

- Name
- Age

- Salary
 - Name ="pravalika"
 - Age=20
 - Print(name)
 - Print(age)
-

Why Variables are Used

Variables help:

- Store information
 - Reuse values
 - Process data
-

Creating Variables

```
name = "Ram"
```

```
age = 20
```

Rules for Variable Names

Allowed

- Letters
- Numbers
- Underscore _

Not Allowed

- Spaces
 - Special symbols
 - Starting with number
 - X = variable name
 - = assignment operator
 - 10 value
-

Data Types in Python

What is Data Type?

Data type defines:

- Type of value stored
-

Main Data Types

Data Type Meaning

int Integer numbers

float Decimal numbers

str Text/String

bool True/False

Integer (int)

Stores whole numbers.

Example:

x = 10

Float (float)

Stores decimal numbers.

Example:

price = 99.5

String (str)

Stores text.

Example:

name = "Python"

Boolean (bool)

Stores:

- True
- False

Example:

`is_passed = True`

5. `list()`

Stores multiple items in square brackets[].

Dictionary

“name”: key

“siri”: value

Input and Output

Output

Output means displaying result to user.

Python uses:

- `print()`

Example:

```
print("Hello")
```

Input

Input means taking data from user.

Python uses:

- `input()`

Example:

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

Type Conversion

What is Type Conversion?

Changing one data type into another.

Examples:

- String to integer
 - Integer to float
-

Types of Conversion

Integer Conversion

```
x = int("10")
```

Float Conversion

```
y = float("10.5")
```

String Conversion

```
z = str(100)
```

Operators in Python

What is Operator?

Operators perform operations on values.

Types of Operators

Type	Purpose
Arithmetic	Calculations
Comparison	Compare values
Logical	Combine conditions
Assignment	

Arithmetic Operators

Used for mathematical operations.

Operator Meaning

+	Addition
-	Subtraction

Operator Meaning

* Multiplication

/ Division

Comparison Operators

Used to compare values.

Result:

- True
- False

Example:

- >
- <
- ==

• Assignment Operators

- Used to assign values.

Operator Example Meaning

=	x = 5	Assign
+=	x += 2	x = x + 2
-=	x -= 2	x = x - 2
*=	x *= 2	x = x * 2
/=	x /= 2	x = x / 2

Logical Operators

Used with conditions.

Operator Meaning

and Both conditions true

or Any one condition true

not Reverse condition

Conditional Statements

What is Conditional Statement?

Used for decision making.

Program checks condition and executes code.

Conditional check

True ----- execute block

False-----skip block

if Statement

Executes code only if condition is true.

Example:

```
if age >= 18:  
    print("Adult")
```

if-else Statement

Provides alternative execution.

Example:

```
if age >= 18:  
    print("Adult")  
else:  
    print("Minor")
```

if-elif-else

Checks multiple conditions.

Used when many choices exist.

Nested if

an if statement inside another if statement.

Loops in Python

What is Loop?

Loop repeats code multiple times.

Helps avoid writing same code again and again.

Types of Loops

	Purpose
for	Fixed repetitions
while	Condition-based repetition
Nested loops	Loop inside another loop

Start



Check condition



True execute block



Repeat again



False stop

for Loop

Used when number of repetitions is known.

Example:

- Printing numbers 1 to 10

For variable in sequence:

statements

while Loop

Runs until condition becomes false.

Used when repetitions are unknown.

While condition:

statements

Loop Control Statements

break

Stops loop immediately.

continue

Skips current iteration.

pass

Does nothing temporarily.

Functions in Python

What is Function?

A function is a reusable block of code designed to perform a specific task.

Advantages of Functions

- Code reuse
 - Better organization
 - Reduces duplication
 - Easy debugging
-

Function Definition

Uses:

- `def` keyword

Example:

```
def greet():  
    print("Hello")
```

Function Call

Calling means executing function.

Example:

```
greet()
```

Parameters and Arguments

Parameters

Variables inside function definition.

Arguments

Values passed during function call.

Return Statement

return sends result back.

Example:

```
def add(a, b):  
    return a + b
```

program starts

|

Function defined

|

Function call

|

Function executes

|

Output produced

Built in function: already provided by python

Lists in Python

What is List?

A list stores multiple values in single variable.

Example:

- Student marks
 - Product names
-

Features of List

- Ordered
 - Mutable
 - Allows duplicates
-

Tuples in Python

What is Tuple?

Tuple is similar to list but immutable.

Immutable means:

- Cannot modify data
-

Sets in Python

What is Set?

Set stores unique values.

Features:

- No duplicates
 - Unordered
-

Dictionaries in Python

What is Dictionary?

Dictionary stores data in:

- Key-value pairs

Example:

- Student details
-

1. Introduction to Modules

What is a Module?

A module is a file containing Python code such as:

- Variables
- Functions
- Classes
- Statements

A module helps organize code and allows code reuse.

Why Use Modules?

- Reduces code duplication
- Makes programs easier to maintain
- Improves readability
- Supports teamwork
- Organizes large projects

Real-Life Example

Imagine a library:

- Mathematics books → Math Module
- Science books → Science Module
- History books → History Module

Similarly, Python stores related functions in modules.

Object-Oriented Programming (OOP)

What is OOP?

Programming based on:

- Objects
- Classes

Class

Blueprint for creating objects.

Object

Real instance of class.

Example:

- Car is class
- Specific car is object

Encapsulation

Encapsulation means binding data and methods together into a single unit.

It also protects data from unauthorized access.

Data hiding

Encapsulation supports data hiding.

Users cannot directly access sensitive information.

2. abstraction

Abstraction means hiding implementation details and showing only essential features. \

3. inheritance

Inheritance allows one class to acquire properties and methods of another class.

File Handling

What is File Handling?

Python can:

- Read files
- Write files
- Update files

File modes

Mode	description
r	read
w	write
a	append
x	create

r+	read and write
w+	write and read
a+	append and read

Why File Handling is Important

Used for:

- Data storage
 - Reports
 - Logs
-

Exception Handling

What is Exception?

Errors occurring during program execution.

Example:

- Division by zero
-

Why Exception Handling?

Prevents program crash.

Uses:

- try
 - except
-

Modules in Python

What is Module?

A module is a file containing Python code.

Benefits:

- Reusability
 - Better organization
-

Libraries in Python

What is Library?

Collection of modules.

Examples:

- NumPy
 - Pandas
-

Advantages of Python

- Easy syntax
 - Beginner friendly
 - Huge community
 - Powerful libraries
 - Cross-platform
-

Limitations of Python

- Slower execution compared to C
- High memory usage
- Less suitable for mobile apps