

# Introduction to Python

Python Programming  
CT108-3-1-PYP

$\pi$



# Programming Language

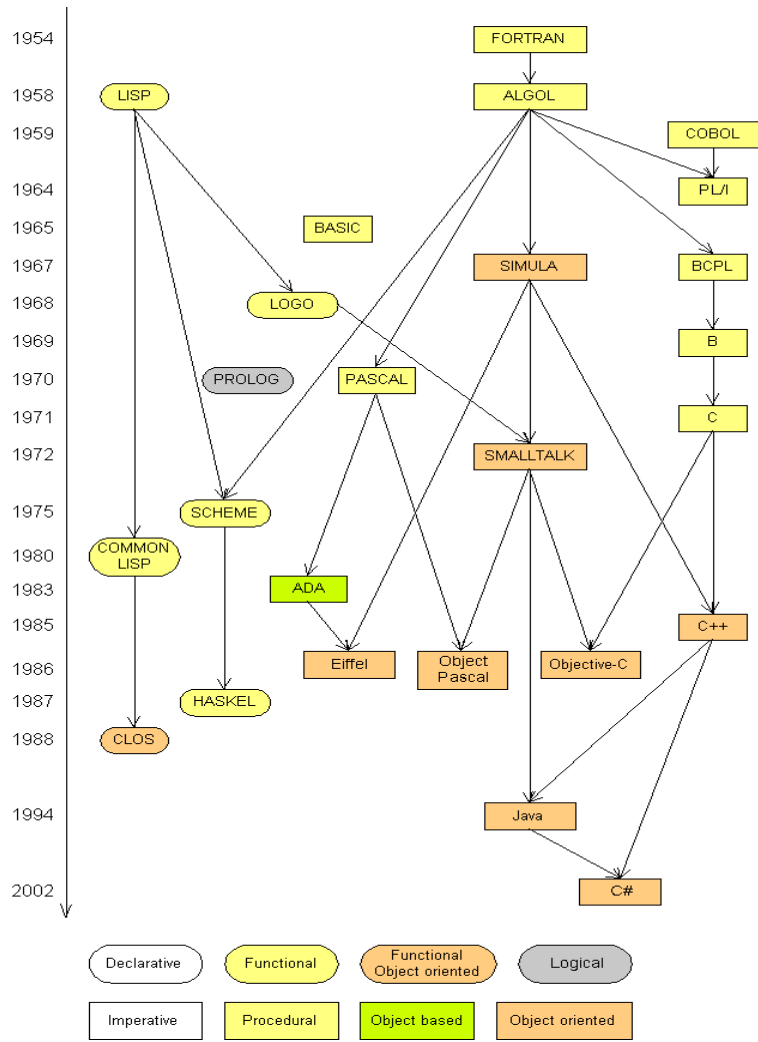


- › Set of rules used to instruct a computer on what operations to perform.
- › Acts as a bridge between humans and machines by offering a readable format for both.
- › Provides a structured way to express algorithms.
- › Helps describe computations in a clear, logical manner.
- › Serves as a tool to create executable models that solve problems in specific domains.
- › Without programming languages, it would be impossible to instruct computers effectively.

# Programming Languages

## › Some influential ones:

- FORTRAN
  - › science / engineering
- COBOL
  - › business data
- LISP
  - › logic and AI
- BASIC
  - › a simple language



# Basic types of instructions in programming

Input

- **Input** (get input from the user)

Math

- **Math** (perform basic mathematical operations)

Conditional  
statement

- **Conditional statement** (check for conditions and execute the appropriate sequence of statements)

Repetition

- **Repetition** (repeatedly performs some sequences of statements)

Output

- **Output** (display data to the user on a monitor or printout)

# Users .vs. Programmers

Who uses the computer or the smart-phone or the e-reader.

Computers are not his area of expertise



**Users**

- See computers as a set of tools - word processor, spreadsheet, map, todo list, etc.

Just owns the device and knows how it works.

# Users .vs. Programmers

Have tools to build new tools

Write tools to automate a task



Who understands computers from a technological point of view.

**Programmers**

Writes the code to make the computer understand commands and perform various functions

# What is a Program/Code/Software?

A Program is a **set of instructions** to the computer that tell it how to perform a task. A programmer must know the basic concepts of mathematics to write a programs. Let's discuss programming basics:

- **Code** or **source code**: The sequence of instructions in a program.
- **Syntax**: The set of rules that define how code must be written
- **Output**: The messages printed to the user by a program.
- **console**: The text box onto which output is printed

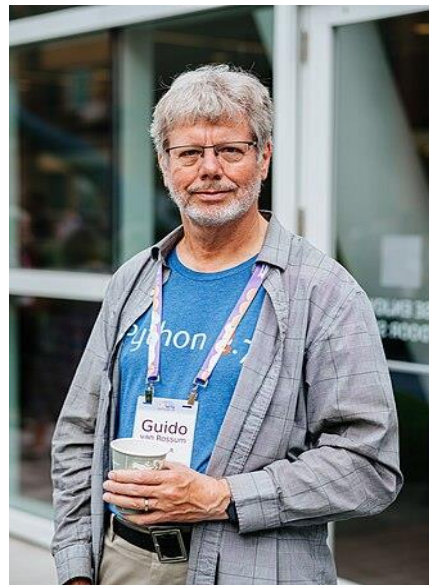
# What is Python?

**Python** is a high-level programming language designed by **Guido van Rossum** in 1991 and developed by the **Python Software Foundation**.

The name "Python" is inspired by the BBC show *Monty Python's Flying Circus* and has no relation to reptiles.

Python emphasizes **code readability**, and its simple syntax enables programmers to express concepts in fewer lines of code.

It is a **free, open-source** software with a strong **community-based development model**.



Guido van Rossum



## Why Python?

- It is easy to learn
- Relatively fast
- Object-oriented
- Strongly typed
- Widely used
- Portable
- Directly interpretable



$\pi$

## Why Python?



- C is much faster but much harder to use.



- Java is about as fast and slightly harder to use.



- Perl is slower, is as easy to use, but is not strongly typed.

## Python Features and Advantage

- **Object-Oriented:** Supports key concepts like polymorphism, operator overloading, and multiple inheritance.
- **Indentation:** Uses indentation for block structure, enhancing readability and simplicity.
- **Free & Open Source:** Easily downloadable, with accessible source code and community-driven development.
- **Powerful:** Dynamic typing, extensive libraries, and integration with compiled code for performance.

## Python Features and Advantage

- **Portable:** Runs consistently on all major platforms with a Python interpreter.
- **Mixable:** Easily integrates with other languages and technologies for hybrid solutions.
- **Easy to Use:** No intermediate compile/link steps; bytecode compilation ensures fast development.
- **Easy to Learn:** Intuitive syntax and structure make it beginner-friendly.

## Uses of Python

- System programming
- Graphical User Interface Programming
- Internet Scripting
- Gaming, Images, XML , Robot and more
- Database Programming
- Component Integration

# Datatypes

Text Type:	<code>str</code>
Numeric Types:	<code>int, float, complex</code>
Sequence Types:	<code>list, tuple, range</code>
Mapping Type:	<code>dict</code>
Set Types:	<code>set, frozenset</code>
Boolean Type:	<code>bool</code>
Binary Types:	<code>bytes, bytearray, memoryview</code>

# Python Program – Python IDLE

```
>>> 1 + 2
```

```
3
```

```
>>> 2 * 3
```

```
6
```

```
>>> x = 1
```

```
>>> print(x)
```

```
1
```

```
>>> x = x + 1
```

```
>>> print(x)
```

```
2
```

```
>>> exit()
```

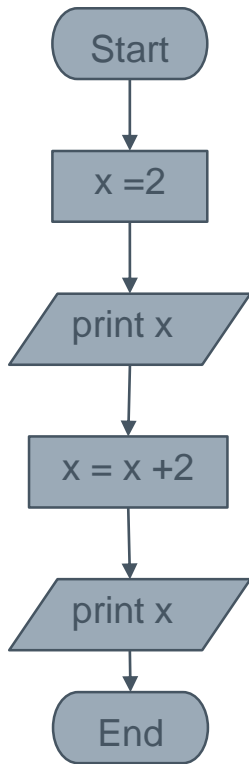
```
>>>quit ()
```

# WHITESPACE & COMMENTS

<b>Text Type:</b>	<code>str</code>
<b>Numeric Types:</b>	<code>int, float, complex</code>
<b>Sequence Types:</b>	<code>list, tuple, range</code>
<b>Mapping Type:</b>	<code>dict</code>
<b>Set Types:</b>	<code>set, frozenset</code>
<b>Boolean Type:</b>	<code>bool</code>
<b>Binary Types:</b>	<code>bytes, bytearray, memoryview</code>



# Sequential Steps or Flow - Example



Program:

```

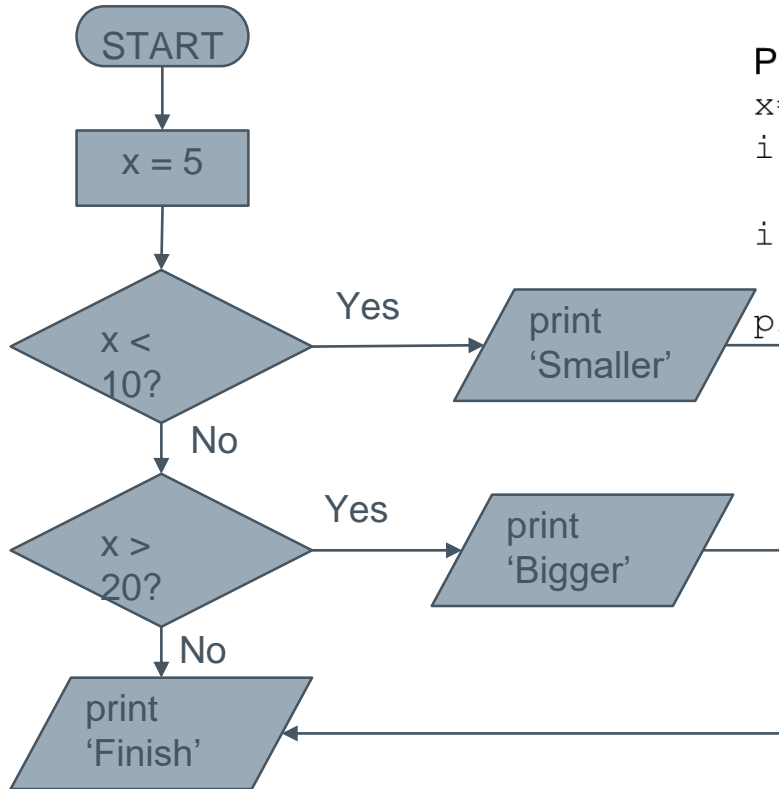
x = 2
print(x)
x = x + 2
print(x)
  
```

Output:

2  
4

When a program is running, it flows from one step to the next. We as programmers set up “paths” for the program to follow

# Conditional Steps or Flow - Example



Program:

x=5

if (x<10) :

    print('Smaller')

if (x>20) :

    print('Bigger')

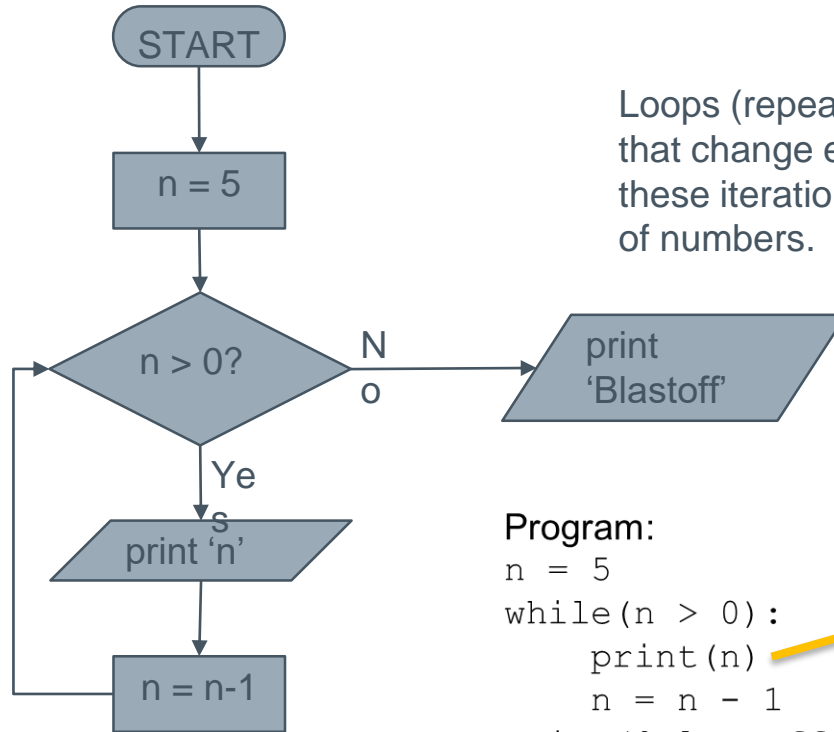
print('Finish')

Output:

Smaller

Finish

# Repetition Steps or Flow - Example



Loops (repeated steps) have iteration variables that change each time through a loop. Often these iteration variables go through a sequence of numbers.

**Program:**

```

n = 5
while(n > 0):
    print(n)
    n = n - 1
print('Blastoff!')
  
```

Output:

5  
4  
3  
2  
1

Blastoff!