WELCOME

Python  Series

AGENDA

Introduction to Programming Language
Examples
Code Flow
Types of Programming Languages
Characteristics
Terminologies
Advantages & Disadvantages

Python  Series

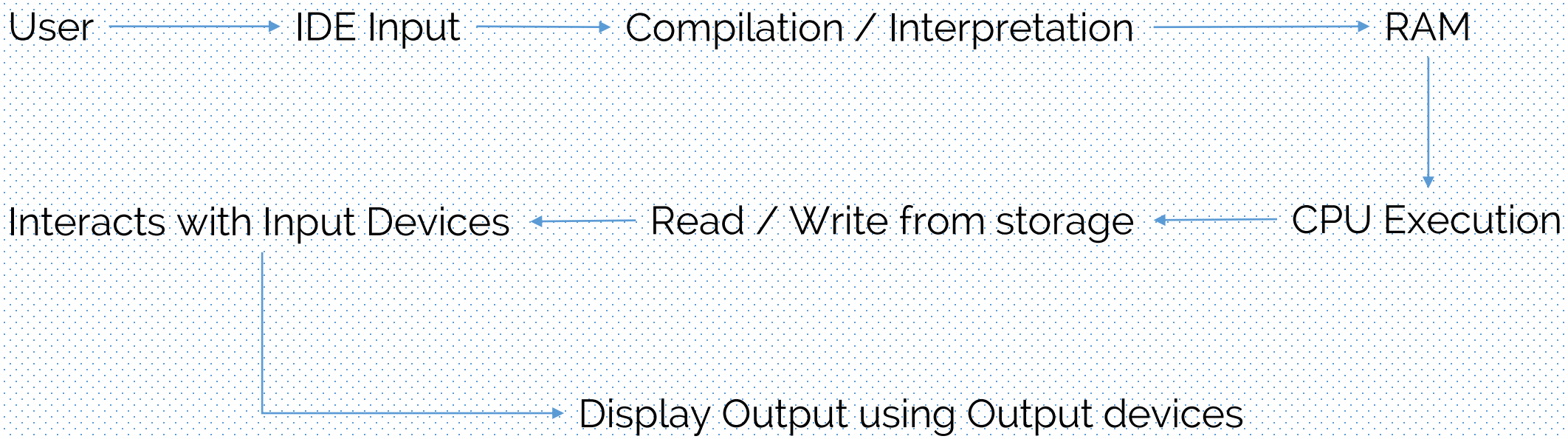
INTRODUCTION TO PROGRAMMING LANGUAGE

- **Programming Language**: Set of instructions and syntax to perform specific task
- **Purpose**: Specify instructions for computers to perform tasks.
- **Variety**: Many different languages exist, each with unique syntax, structure and command
- **Choice of Language**: Depends upon project requirements, platforms, audience and desired outcomes

Examples

Python  Series

CODE FLOW



Python Series

TYPES OF PROGRAMMING LANGUAGES

Types

- Low Level Language
  - Machine Language
  - Assembly Language
- Middle Level Language
- High Level Language
  - Procedural Oriented Language
  - Object Oriented Language
  - Functional Programming Language
  - Markup Languages

Python Series

LOW LEVEL LANGUAGE

Machine Languages

- Machine Dependent (hardware specific)
- Consists of binary codes (0s and 1s)
- Runs directly by processor (hence very fast)

Assembly Language (ASL)

- Set of instructions in symbolic and human understandable form
- Uses assembler to convert ASL code to Machine Code

HIGH LEVEL LANGUAGE

Procedural Oriented Programming Language (POP)

- Follow a sequence of commands or instructions to achieve the desired output.
- C, FORTRAN, Pascal

Object Oriented Programming Language (OOP)

- Programs are divided into small objects called **objects**
- Objects contains **data** (field) and **code** (procedures)
- Java, C, C++, Python

HIGH LEVEL LANGUAGE

Functional Programming Language

- Treat computation as evaluation of mathematical functions.
- Focuses on “what to solve” instead of “how to solve”
- Haskell, Lisp, Erlang

Markup Language

- Not programming language in traditional sense
- Included in programming language due to their role in web development and document formatting
- HTML, XML

CHARACTERISTICS

- **Syntax**: The set of rules that define how programs in a language must be written.
- **Semantics**: The meaning of the instructions written in a language.
- **Abstraction**: The ability to simplify complex systems and manage them at a high level.
- **Efficiency**: The performance and speed of programs written in a language.
- **Readability**: How easy it is to understand the code written in a language.
- **Portability**: The ability of code to run on different types of systems without modification.
- **Expressiveness**: The ease with which a language can express complex operations.
- **Maintainability**: How easy it is to update and modify code written in a language.
- **Safety**: The ability of a language to prevent or minimize errors.
- **Interoperability**: The ability to interact and integrate with code written in other languages.

TERMINOLOGIES

- **Variable**: A named space in the memory that stores values.
- **Data Type**: What kind of data a variable can hold (Array, String, Integer, Boolean).
- **Function**: A block of code designed to perform a specific task.
- **Loop**: A sequence of instructions that is continually repeated until a certain condition is met.
- **Syntax**: The set of rules that defines how programs in a language must be written.
- **Compiler**: A program that translates code from a high-level programming language to a lower-level language to create an executable program.
- **Debugging**: The process of finding and resolving defects or problems within a computer program.
- **Algorithm**: A step-by-step procedure for solving a problem or accomplishing a task.
- **Statement**: A single line that perform specific operation.
- **Operators**: Symbol that performs operations: +,-,*,/ etc.
- **Comment**: Ignored by computer, but valuable for humans.

ADVANTAGES & DISADVANTAGES

Advantages

- **Expressiveness**: Perform complex task in relatively few steps.
- **Ease Of Use**: Easy to learn and use
- **Portability**: code can be easily converted to machine code for variety of different platforms.
- **Debugging Tools**: Help coders to find and fix errors.

Disadvantages

- **Learning Curve**: Challenging to Learn at first
- **Compatibility**: Not all programming languages are compatible with all systems.
- **Performance**: HLL are slower than LLL

Python  Series

UPCOMING...

Interpreter Vs Compiler Vs Assembler
Python 2 Vs Python 3
Pypl
Pip
venv
Libraries, Frameworks, Tools

Python  Series