

# TYPES OF PROGRAMMING LANGUAGES

By : Hend abdelnasser

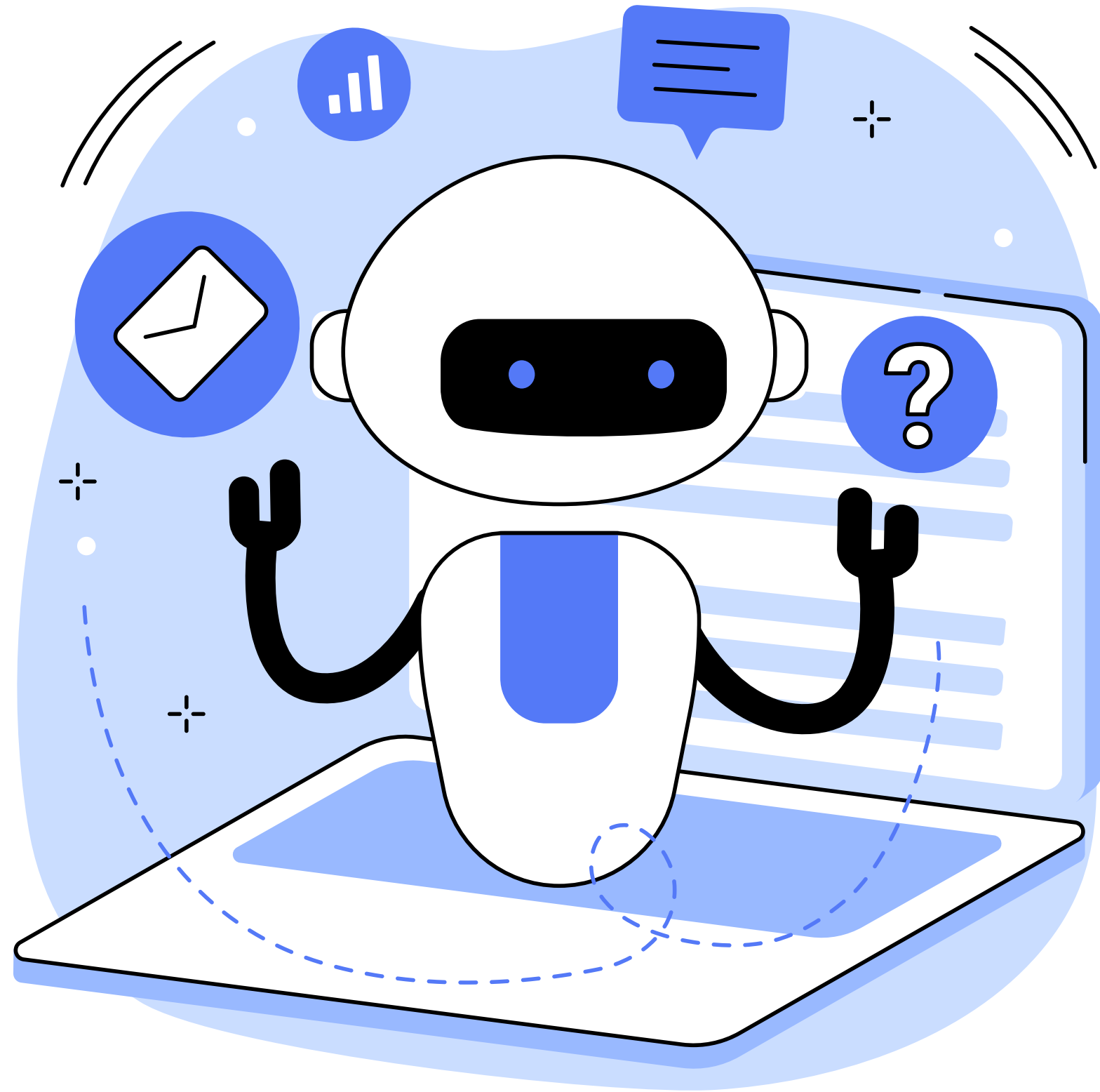
# CONTENTS



*By: Hend abdelnasser*

## TYPES OF PROGRAMMING LANGUAGES

- 1 Low level , High level languages
- 2 Interpreted , Compiled
- 3 Programming ,Scripted
- 4 Open source , not open source
- 5 Support OOP , not supporting OOP



# DIFFERENCE BETWEEN HIGH LEVEL AND LOW LEVEL LANGUAGES

By : Hend abdelnasser

Parameter	High-Level Language	Low-Level Language
Basic	Programmer-friendly, easy to understand, debug, and widely used.	Machine-friendly, difficult for humans but easy for machines.
Ease of Execution	Difficult to execute.	Very easy to execute.
Process of Translation	Requires a compiler or interpreter.	
understanding	It is easy to understand.	It is tough to understand.
By : Hend abdelnasser		

Parameter	High-Level Language	Low-Level Language
Efficiency of Memory	Low memory efficiency (uses more memory).	High memory efficiency (uses less memory).
Debugging	Debugging is easy.	Debugging is complex comparatively.
Portability	Portable across different devices.	Not portable between devices.
Maintenance	Simple and easy to maintain.	Portable across different devices.
By : Hend abdelnasser		

Parameter	High-Level Language	Low-Level Language
Machine Dependency	Not machine-dependent.	Machine-dependent.
Usage	Very common and widely used.	Rarely used nowadays.
Abstraction	Provides high-level abstraction.	Provides little or no abstraction.
Hardware Knowledge	No need for hardware knowledge.	Requires hardware knowledge.
By : Hend abdelnasser		

Parameter	High-Level Language	Low-Level Language
Facilities Provided	Does not provide direct hardware access.	Provides direct hardware-level control.
Ease of Modification	Difficult to modify as a single statement may execute multiple instructions.	Easy to modify as statements map directly to processor instructions.

*By : Hend abdelnasser*



# DIFFERENCE BETWEEN COMPILED AND INTERPRETED LANGUAGE

By : Hend abdelnasser





## Compiled Language

A compiled language is a programming language that is translated into machine code before execution. This allows the processor to run the program directly. Examples include C, C++, C#, COBOL, and CLEO.

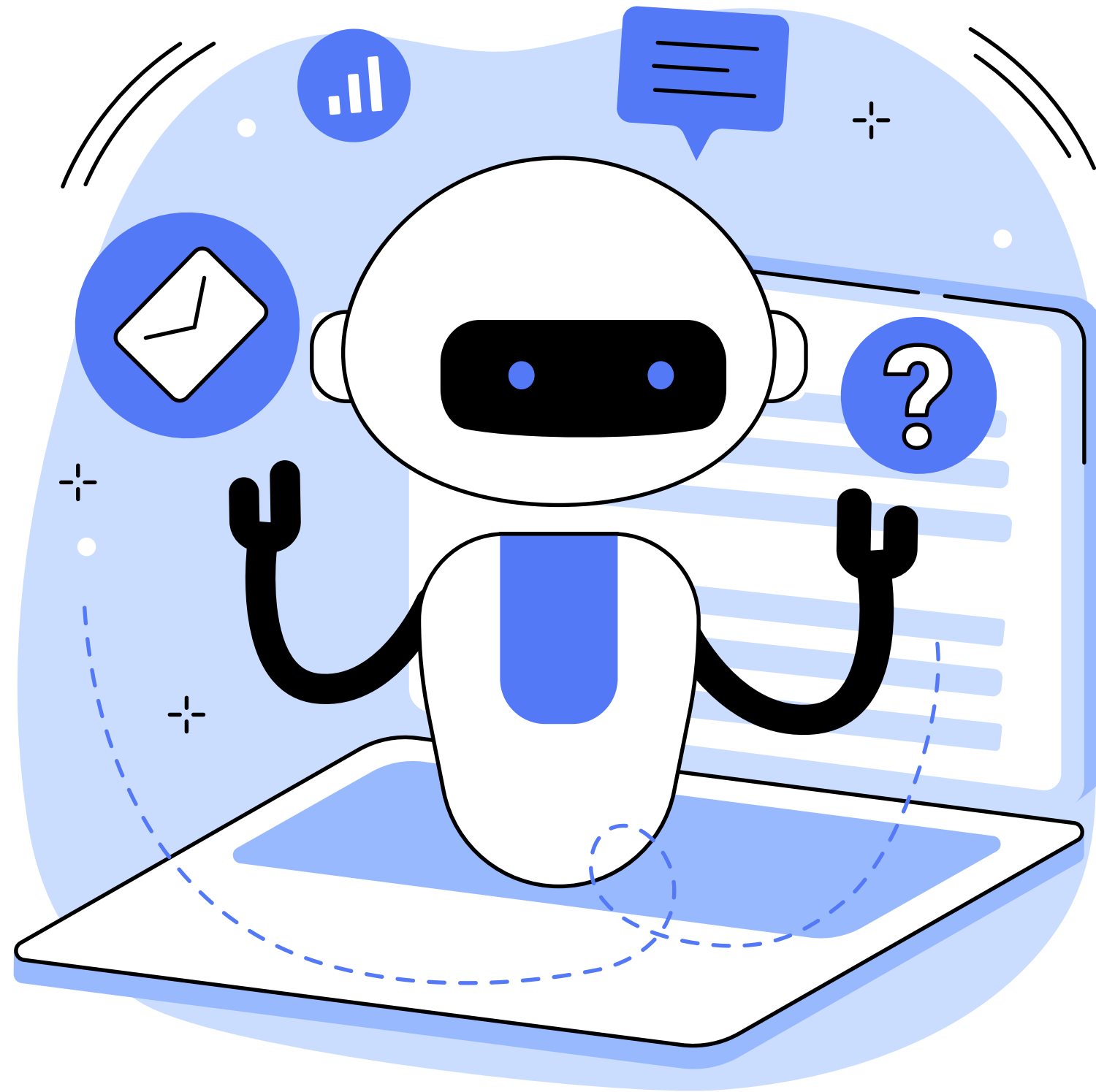


## Interpreted Language

An interpreted language is a programming language where instructions are executed directly without being compiled into machine code first. Examples include Python, JavaScript, BASIC, and Perl.

Aspect	Compiled Language	Interpreted Language
Execution Process	Delivers better performance.	Executes instructions directly without prior compilation.
Machine Code	Translates the entire code into machine language before execution.	Executes instructions without converting them into machine code beforehand.
By : Hend abdelnasser		

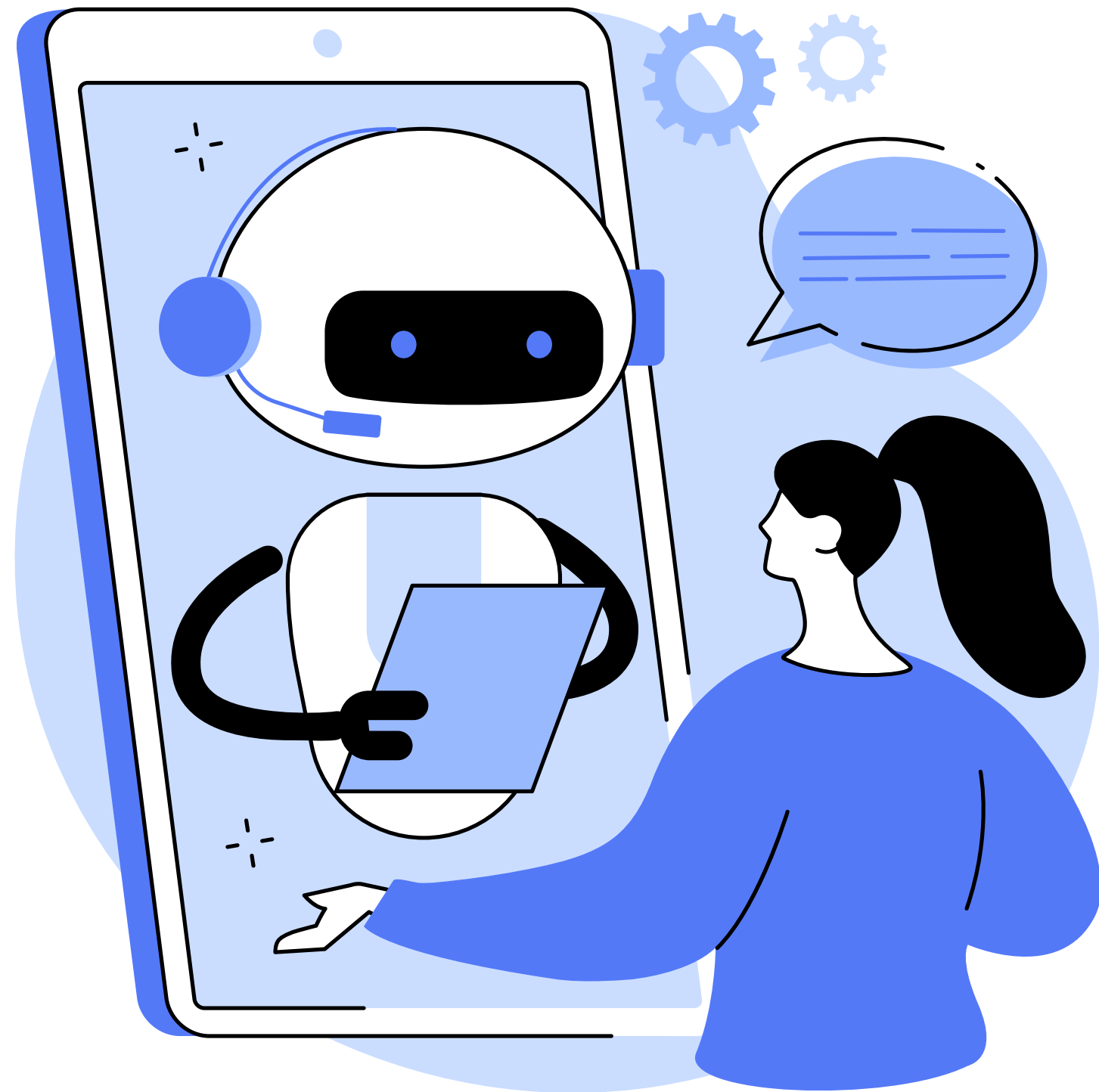
Aspect	Compiled Language	Interpreted Language
Performance	Requires at least two steps: compilation and execution.	Generally slower compared to compiled languages.
Speed	Faster execution due to precompiled code.	Slower execution as instructions are interpreted in real-time.
CPU Execution	Runs directly on the CPU.	The interpreter executes instructions instead of compiling them first.
By : Hend abdelnasser		



# DIFFERENCE BETWEEN OPEN SOURCE , NOT OPEN SOURCE LANGUAGE

By : Hend abdelnasser

Key Differences	Open Source Language	Not Open Source Language
Source Code Access	Publicly available	Restricted or unavailable
Modification	Allowed	Not allowed without permission
Cost	Usually free	Often requires a license or payment
Community	Large community	Controlled by a single entity
Transparency	High	Low
By : Hend abdelnasser		



# DIFFERENCE BETWEEN PROGRAMMING LANGUAGE, SCRIPTING LANGUAGE

By : Hend abdelnasser

## Programming

A programming language is a formal language used to develop standalone software applications. These languages typically require compilation before execution.  
Examples: C, C++, Java, Rust.

## Scripting

A scripting language is a type of programming language used to automate tasks and execute scripts within larger applications. These languages are usually interpreted rather than compiled. Examples: Python, JavaScript, Perl, Bash.



Language	Low/ High- Level	Interpreted / Compiled	Programming / Scripting	Open Source / Not Open Source	Supports / Not Support OOP
C	High	Compiled	Programming	Open Source	Does Not Support OOP
C++	High	Compiled	Programming	Open Source	Supports OOP
Python	High	Interpreted	Scripting	Open Source	Supports OOP
JavaScript	High	Interpreted	Scripting	Open Source	Supports OOP
Assembly	Low	Compiled	Programming	Open Source	Does Not Support OOP

*By : Hend abdelnasser*