Programming Languages

What are Programming Languages?

- Programming languages are algorithms designed for specific performers.

- Each programming language has its own command system executed by a computer.

- Commands are written in machine language or machine code.

- Assembly language is used to convert machine code into symbolic form for easier understanding.

- High-level programming languages use commands that combine sequences of machine instructions.

Photo by Pexels

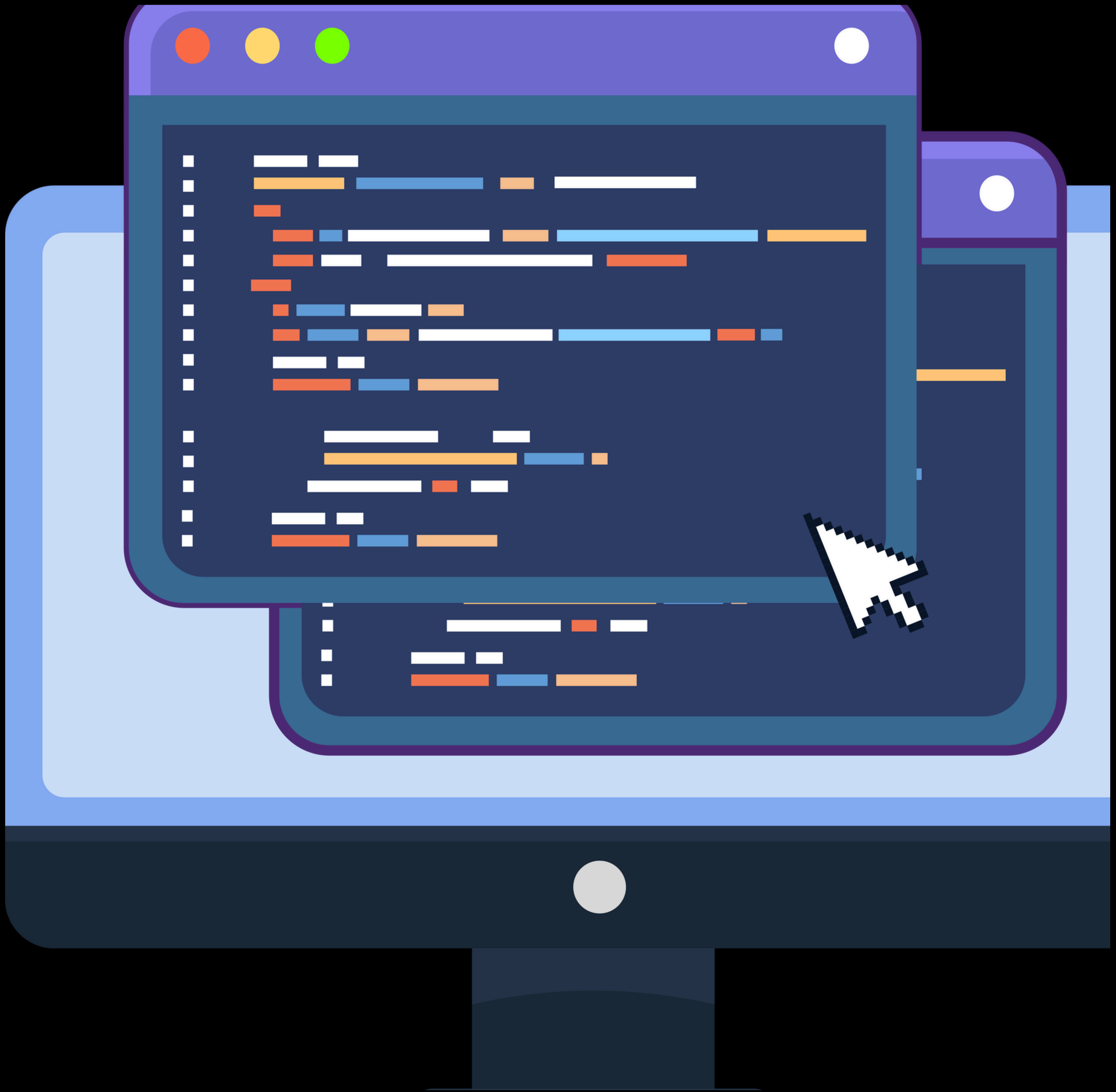
Programming Language Classifications

- Programming languages can be classified based on their semantic differences from machine code.

- Low-level languages are closer to machine code, like assembly languages.

- High-level languages are more abstract and easier for humans to understand.

- Languages can also be classified as compiled or interpreted, although many languages blur (non è chiara) this distinction.



Machine Language and Assembly Language

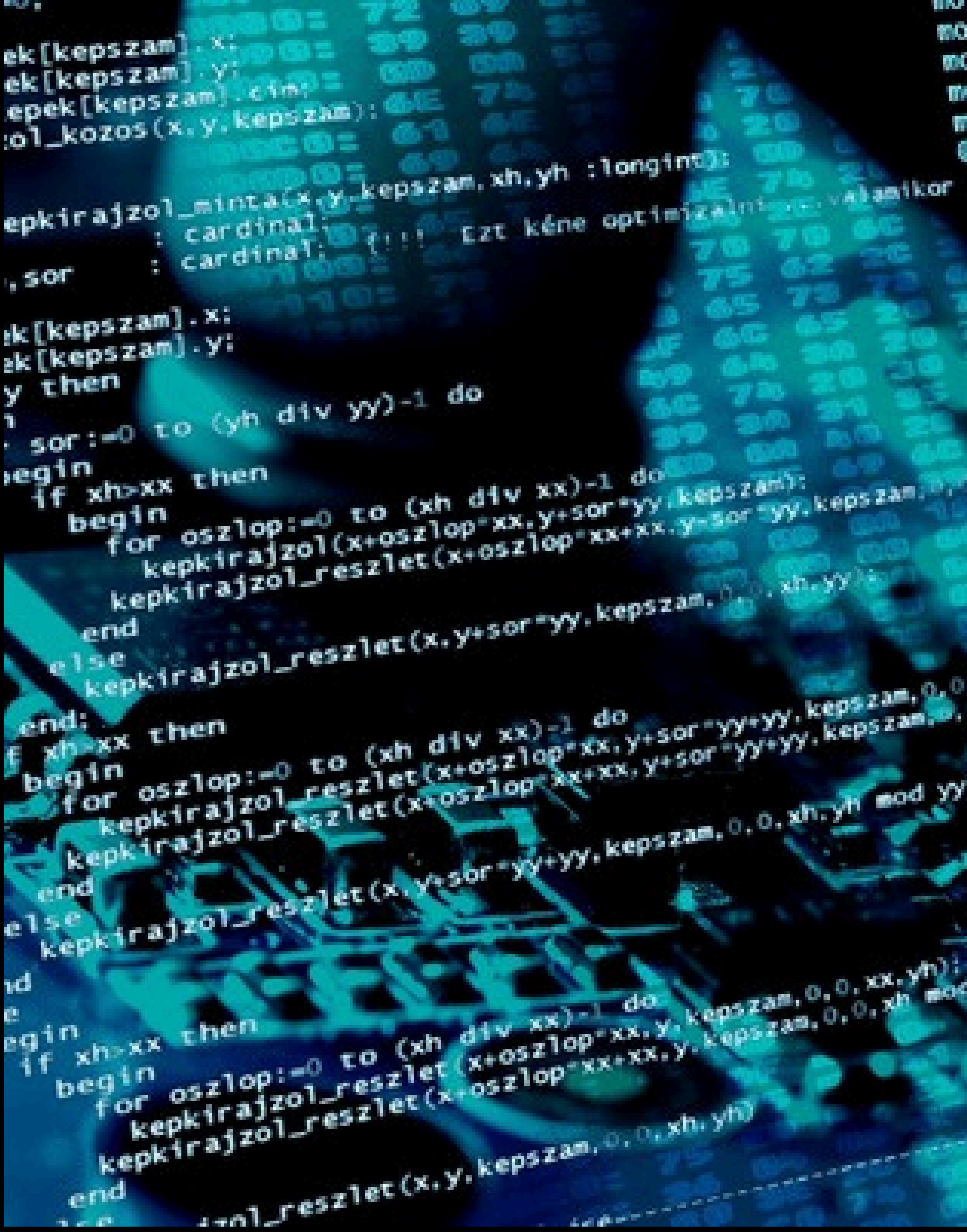
- Machine language commands are written in binary code. Programmers plan the set of instructions using an algorithm.

- Low – level languages are examples of machine languages. The main function is to operate, manage and manipulate the computer hardware. They are directly executable on the computer hardware.

- Assembly language is a low-level language that reflects machine code. Each line of the code usually corresponds to one machine instruction.

- High-level programming languages use English-like commands for easier human understanding. They are similar to a natural language.

- C++, and Python are examples of high-level programming languages.

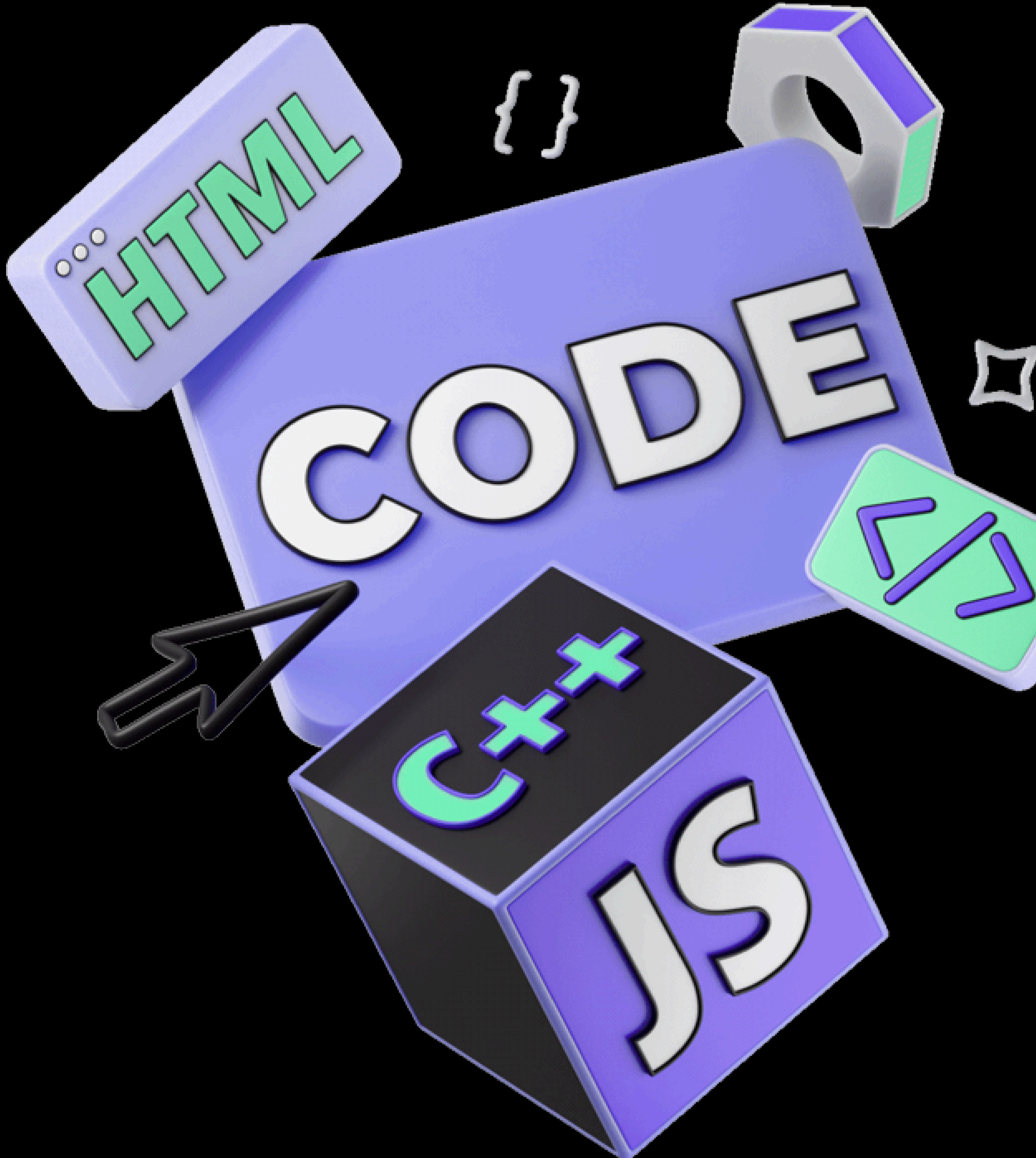


High-Level Programming Languages

- High-level programming languages abstract from hardware implementation.

- Object-oriented programming languages, like Java and Python, use objects and classes for code organization.

- Structured programming languages, like C, emphasize block structures.



Translation programs

Compilers and Interpreters are defined as translator programs and are used for high level languages

- Compilers translate source code into executable files directly run by the operating system.

- Interpreters read and execute code line-by-line or transform code into bytecode (machine code).

- Languages like JavaScript, and Python use interpreters to execute code.

Translation programs

Assemblers are classified as translation programs. They translate the assembly language into a machine code (bytecode). One instruction in assembly language is the equivalent to one machine intruction.

Object oriented languages

- Object-Oriented Programming (OOP) is a way of writing code where you model (shape again) real-world things as objects.

- Object-Oriented Programming (OOP) is designed to make complex code easier to manage.

Why do you use OOP (Object oriented languages)?

- Easier to organize and reuse code.
- Makes large programs easier to manage.
- Models real-world concepts more clearly.

Syntax, Semantics, and Operators

- Syntax and grammar define the rules for writing programs in a language.

- Semantics determine the meaning and interpretation of language constructs.

- Syntax and semantics are checked during program analysis.

Photo by Pexels

Variables, Functions, and Procedures

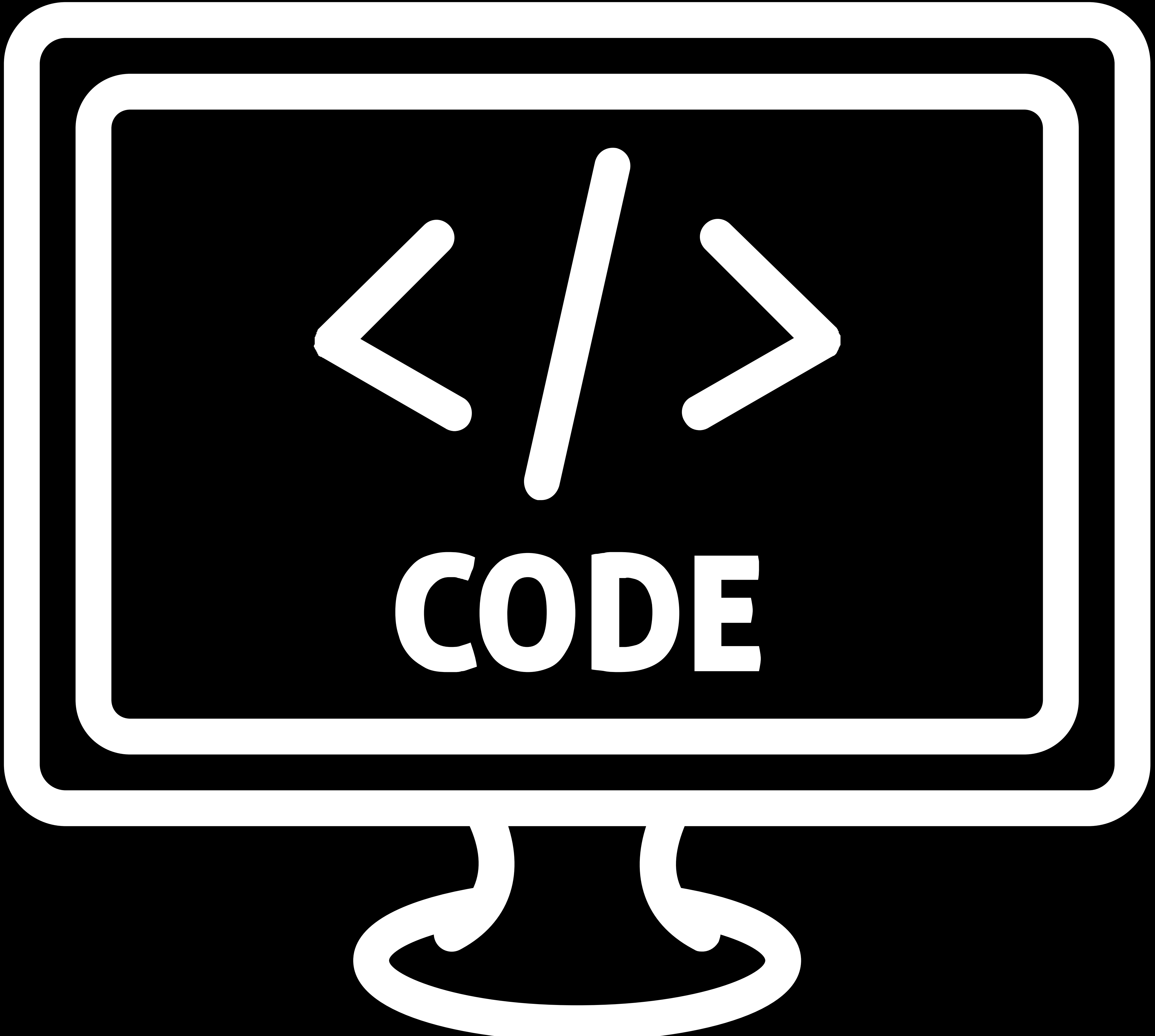
- Variables can be constants or changeable values.

- Functions represent processes that compute a specific value based on input arguments.

- Procedures are sets of commands executed in sequence without returning a value.

- High-level languages provide libraries of standard functions.

- Objects and classes are key concepts in object-oriented programming.



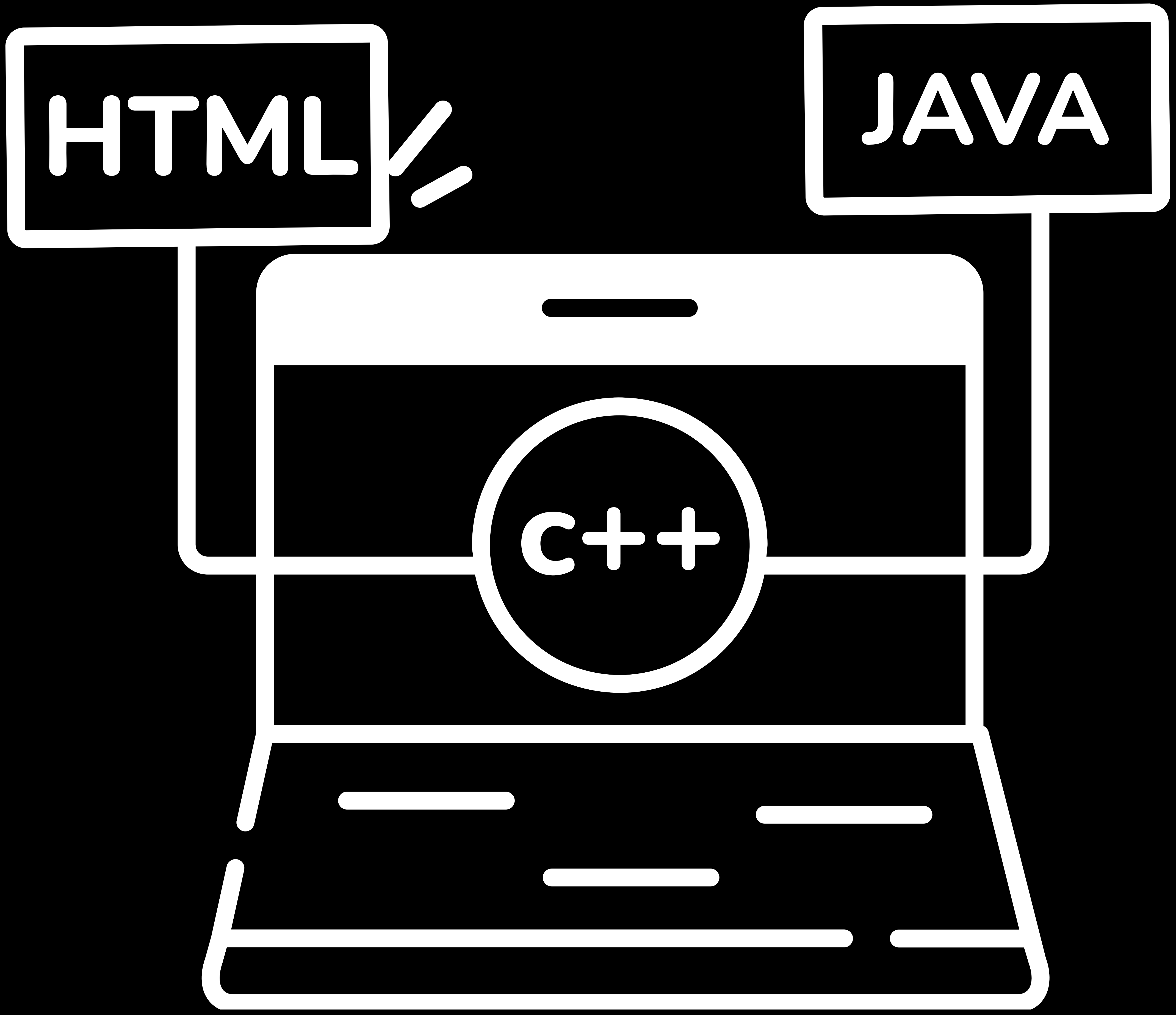
Conclusion

- Programming languages are fundamental tools for writing computer programs.

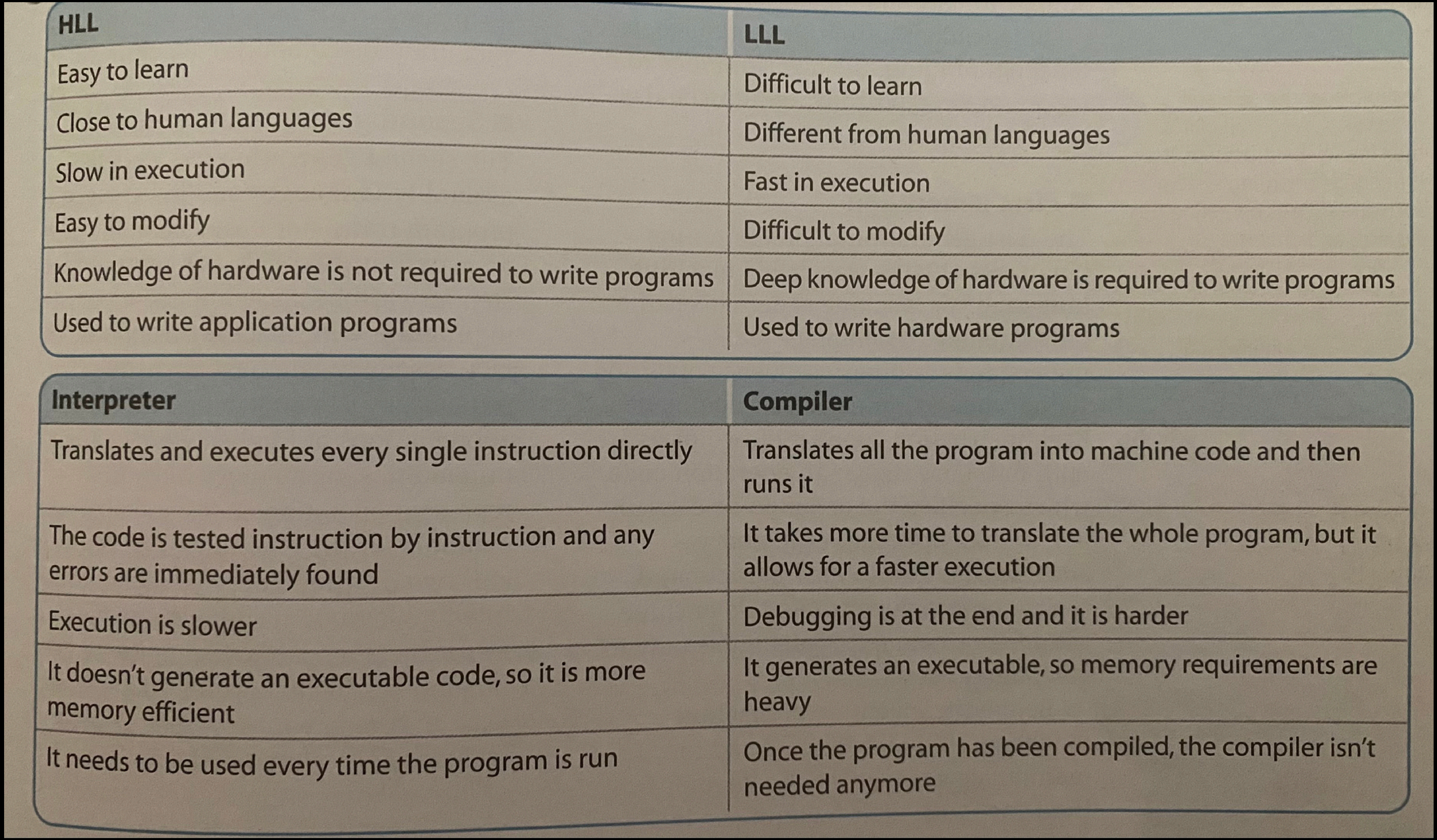
- They provide the means to translate high-level algorithms into executable code.

- Understanding different programming languages and their features is essential for effective coding.

- Continuous learning and exploration of new languages is crucial in the dynamic field of programming.



TO SUM UP



HLL	LLL
Easy to learn	Difficult to learn
Close to human languages	Different from human languages
Slow in execution	Fast in execution
Easy to modify	Difficult to modify
Knowledge of hardware is not required to write programs	Deep knowledge of hardware is required to write programs
Used to write application programs	Used to write hardware programs

Interpreter	Compiler
Translates and executes every single instruction directly	Translates all the program into machine code and then runs it
The code is tested instruction by instruction and any errors are immediately found	It takes more time to translate the whole program, but it allows for a faster execution
Execution is slower	Debugging is at the end and it is harder
It doesn't generate an executable code, so it is more memory efficient	It generates an executable, so memory requirements are heavy
It needs to be used every time the program is run	Once the program has been compiled, the compiler isn't needed anymore

HLL (HIGH LEVEL LANGUAGES) LLL (LOW LEVEL LANGUAGES)

TO SUM UP: ASSEMBLY LANGUAGES

POSITIVE ASPECTS OF ASSEMBLY LANGUAGES	NEGATIVE ASPECTS OF ASSEMBLY LANGUAGES
They are fast	They are difficult to learn, read and understand
They can be a powerful tool to write better programs	They can be time consuming
They can do what High level languages can’t do	Computers have eliminated the need for assembly language

Programming languages - C family

C is a high-level programming language that was developed in the early 1970s. It is simple, and it has close relationship with the hardware.

Key Features of C:

- Procedural Language: focuses on functions and step-by-step instructions.
- Compiled Language: Needs a compiler to translate code into machine language.
- Portable: C programs can be compiled on many different machines with little or no modification.

Common Uses:

Operating systems (e.g., much of Unix/Linux is written in C)

C++

- An extension of C with added features like object-oriented programming.
- Used for games, software development, and high-performance apps.
- Supports classes, inheritance, and templates for more complex designs.

C# (C-Sharp) is a modern, object-oriented programming language developed by Microsoft.

- It is designed to be easy to use, safe, and powerful.
- It looks like C/C++, but avoids many of their confusing or risky parts.

- What You Can Build With C#:Desktop apps , Mobile apps, Cloud apps

Programming languages - HTML

HTML (HyperText Markup Language) is the standard language used to create web pages.

A markup language uses tags or symbols to tell a computer how to display or organize content — not what the content means or does.

In Simple Terms:

- HTML tells a web browser what to show on a webpage.
- It defines the structure of a webpage using tags.

Key Features:

- Human-readable: You can understand it easily.
- Tag-based: Uses angle brackets (like <title>) to mark up content.
- Describes layout, not logic or behavior.
- Longer texts are organised in headings.
- Graphic elements have form (area where the user can insert menus, checkboxes, etc.

Programming languages - JAVA

- Java is a high-level, object-oriented programming language.

- Java code is compiled into bytecode, which runs on the Java Virtual Machine (JVM).
- This lets Java programs run on any device with a JVM — Windows, Mac, Linux, Android, etc.
- Java Virtual Machine is a type of environment that works with any platform. It changes bytecode into machine language and executes it.

- The Java API is a massive library with tools that ready to use and can help you to write Java programs more easily and efficiently

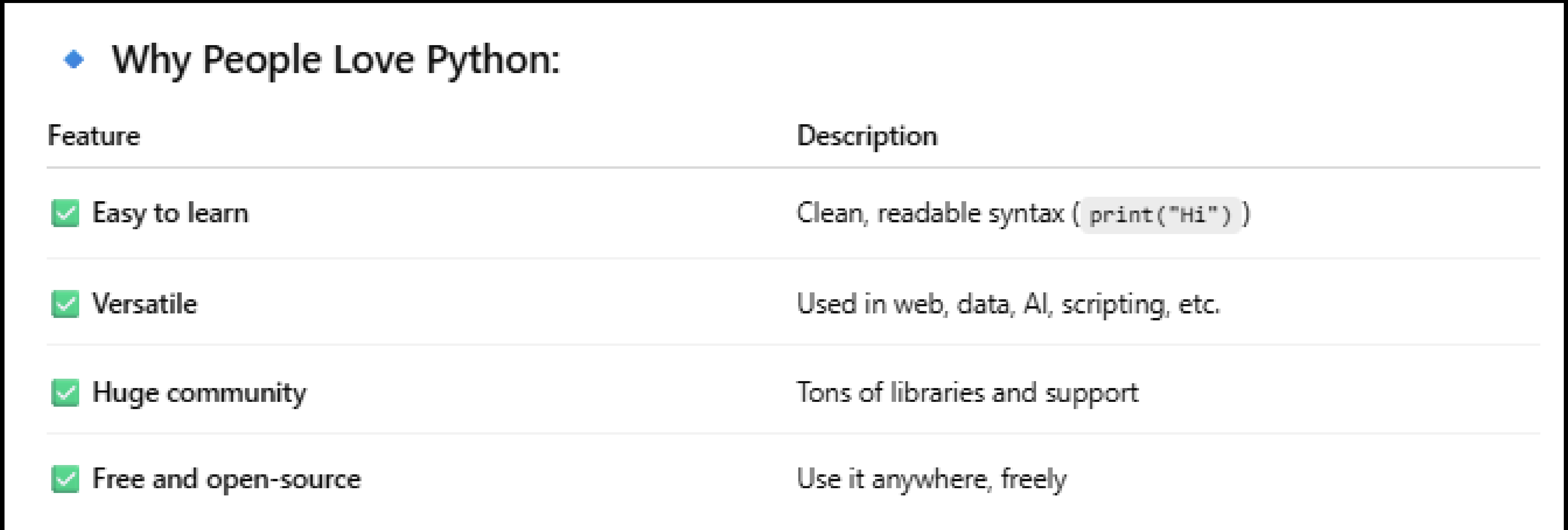
- API is Java Application Programming Interface

Programming languages - PYTHON

Python is a high-level, easy-to-read programming language that's great for beginners and powerful enough for professionals.

In Simple Terms:

- Python is like writing English-like instructions for a computer.
- it's used in websites, AI, data analysis, automation, games, and more.



🔹 Why People Love Python:

Feature	Description
✅ Easy to learn	Clean, readable syntax (`print("Hi")`)
✅ Versatile	Used in web, data, AI, scripting, etc.
✅ Huge community	Tons of libraries and support
✅ Free and open-source	Use it anywhere, freely

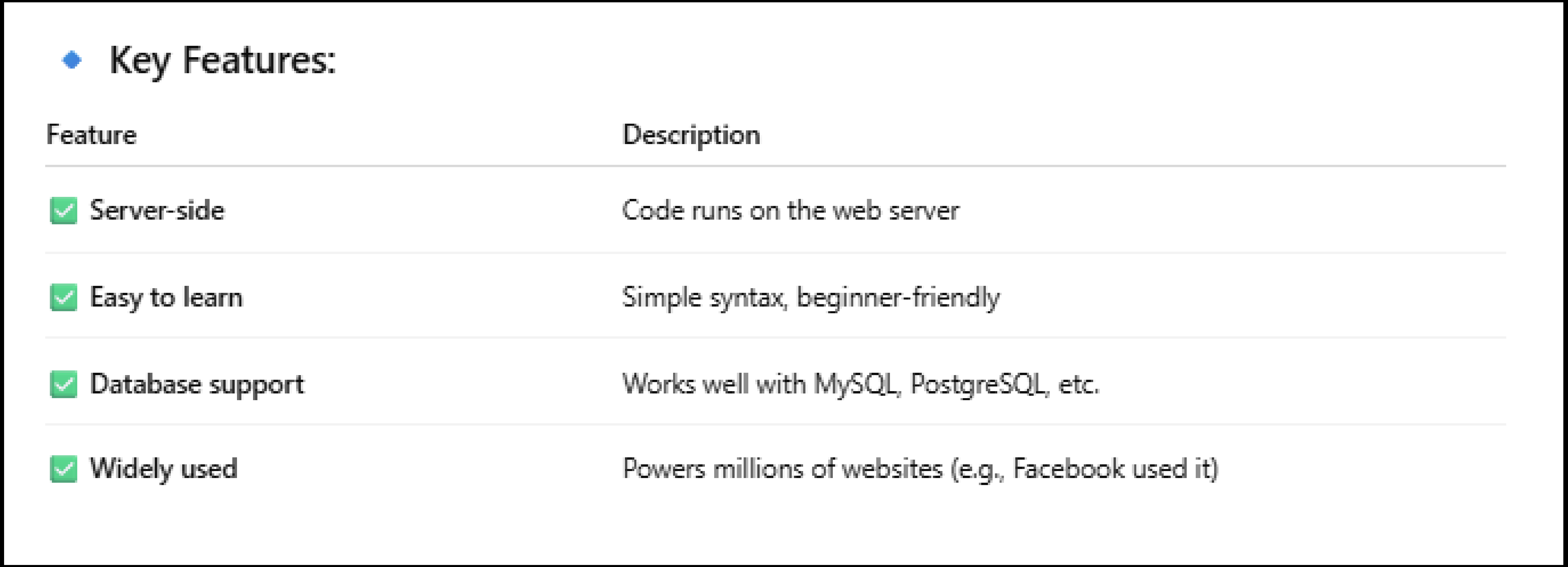
Programming languages - PHP

It’s a server-side scripting language mainly used to build dynamic websites and web applications.

What is a server-side scripting language?
The programming language runs on the server and not in the browser.

What PHP is Used For:

- Websites with user accounts, forms, and content
- E-commerce platforms (like WooCommerce)



🔹 Key Features:

Feature	Description
✅ Server-side	Code runs on the web server
✅ Easy to learn	Simple syntax, beginner-friendly
✅ Database support	Works well with MySQL, PostgreSQL, etc.
✅ Widely used	Powers millions of websites (e.g., Facebook used it)