Programming Languages: Types and Features

A programming language issues a series of commands that help computers, mobile phones, tablets, and other electronic devices function as intended and perform various tasks. There are many types of programming languages, and the correct one must be chosen based on the philosophy and objectives of a particular device or program. Some of the key features of programming languages include:

1. **Syntax**: The specific rules and structure used to write code in a programming language.
2. **Data Types**: The type of values that can be stored in a program, such as numbers, strings, and booleans.
3. **Variables**: Named memory locations that can store values.
4. **Operators**: Symbols used to perform operations on values, such as addition, subtraction, and comparison.
5. **Control Structures**: Statements used to control the flow of a program, such as if-else statements, loops, and function calls.
6. **Libraries** and Frameworks: Collections of pre-written code that can be used to perform common tasks and speed up development.
7. **Paradigms**: The programming style or philosophy used in the language, such as procedural, object-oriented, or functional.

What is a programming language?

A **programming language** is a set of grammatical rules (both syntactic and semantic) that instruct a computer or a device to behave in a certain way. Each programming language has a vocabulary—a unique set of keywords that follows a special syntax to form and organise computer instructions.

Examples of popular programming languages include Python, Java, C++, JavaScript, and Ruby. Each language has its own strengths and weaknesses and is suited for different types of projects.
A programming language is a formal language that specifies a set of instructions for a computer to perform specific tasks. It's used to write software programs and applications, and to control and manipulate computer systems. There are many different programming languages, each with its own syntax, structure, and set of commands. Some of the most commonly used programming languages include Java, Python, C++, JavaScript, and C#. The choice of programming language depends on the specific requirements of a project, including the platform being used, the intended audience, and the desired outcome. Programming languages continue to evolve and change over time, with new languages being developed and older ones being updated to meet changing needs. Now before we understand what programming is, you must know what is a computer. A computer is a device that can accept human instruction, processes it, and responds to it or a computer is a computational device that is used to process the data under the control of a computer program. Program is a sequence of instruction along with data.

The basic components of a computer are:

1. Input unit
2. Central Processing Unit(CPU)
3. Output unit

The CPU is further divided into three parts-

- Memory unit
- Control unit
- Arithmetic Logic unit
-

Most of us have heard that CPU is called the brain of our computer because it accepts data, provides temporary memory space to it until it is stored(saved) on the hard disk, performs logical operations on it and hence processes(here also means converts) data into information. We all know that a computer consists of hardware and software. Software is a set of programs that performs multiple tasks together. An operating system is also software (system software) that helps humans to interact with the computer system.

A program is a set of instructions given to a computer to perform a specific operation. or computer is a computational device that is used to process the data under the control of a computer program. While executing the program, raw data is processed into the desired output format. These computer programs are written in a programming language which are high-level languages. High level languages are nearly human languages that are more complex than the computer understandable language which are called machine language, or low level language. So after knowing the basics, we are ready to create a very simple and basic program. Like we have different languages to communicate with each other, likewise, we have different languages like C, C++, C#, Java, python, etc to communicate with the computers. The computer only understands binary language (the language of 0's and 1's) also called machine-understandable language or low-level language but the programs we are going to write are in a high-level language which is almost similar to human language.

The piece of code given below performs a basic task of printing "hello world! I am learning programming" on the console screen. We must know that keyboard, scanner, mouse, microphone, etc are various examples of input devices, and monitor(console screen), printer, speaker, etc are examples of output devices.

```
main()
{
clrscr();
printf("hello world! I am learning to program");
getch();
}
```

Characteristics of a programming Language –

- A programming language must be simple, easy to learn and use, have good readability, and be human recognizable.
- Abstraction is a must-have Characteristics for a programming language in which the ability to define the complex structure and then its degree of usability comes.
- A portable programming language is always preferred.
- Programming language's efficiency must be high so that it can be easily converted into a machine code and its execution consumes little space in memory.
- A programming language should be well structured and documented so that it is suitable for application development.
- Necessary tools for the development, debugging, testing, maintenance of a program must be provided by a programming language.
- A programming language should provide a single environment known as Integrated Development Environment(IDE).
- A programming language must be consistent in terms of syntax and semantics.

Basic Terminologies in Programming Languages:

- **Algorithm**: A step-by-step procedure for solving a problem or performing a task.
- **Variable**: A named storage location in memory that holds a value or data.
- **Data Type**: A classification that specifies what type of data a variable can hold, such as integer, string, or boolean.
- **Function**: A self-contained block of code that performs a specific task and can be called from other parts of the program.
- **Control Flow**: The order in which statements are executed in a program, including loops and conditional statements.
- **Syntax**: The set of rules that govern the structure and format of a programming language.
- **Comment**: A piece of text in a program that is ignored by the compiler or interpreter, used to add notes or explanations to the code.
- **Debugging**: The process of finding and fixing errors or bugs in a program.
- **IDE**: Integrated Development Environment, a software application that provides a comprehensive development environment for coding, debugging, and testing.
- **Operator**: A symbol or keyword that represents an action or operation to be performed on one or more values or variables, such as + (addition), – (subtraction), * (multiplication), and / (division).
- **Statement**: A single line or instruction in a program that performs a specific action or operation.

Advantages of programming languages:

1. **Increased Productivity:** Programming languages provide a set of abstractions that allow developers to write code more quickly and efficiently.
2. **Portability:** Programs written in a high-level programming language can run on many different operating systems and platforms.
3. **Readability**: Well-designed programming languages can make code more readable and easier to understand for both the original author and other developers.
4. **Large Community:** Many programming languages have large communities of users and developers, which can provide support, libraries, and tools.
5. Disadvantages of programming languages:
6. **Complexity**: Some programming languages can be complex and difficult to learn, especially for beginners.
7. **Performance**: Programs written in high-level programming languages can run slower than programs written in lower-level languages.
8. **Limited Functionality**: Some programming languages may not have built-in support for certain types of tasks or may require additional libraries to perform certain functions.
9. **Fragmentation:** There are many different programming languages, which can lead to fragmentation and make it difficult to share code and collaborate with other developers.

There have been many programming languages some of them are listed below:

Python - Artificial Intelligence & Machine Learning

Developed by Guido van Rossum in the 1990s, the multi-purpose high-level Python has grown extremely fast over the years to become one of the most popular programming languages today. And the number one reason for Python's popularity is its beginner-friendliness, which allows anyone, even individuals with no programming background, to pick up Python and start creating simple programs.

But that's not all. It also offers an exceptionally vast collection of packages and libraries that can play a key role in reducing the ETA for your projects, along with a strong community of like-minded developers that is eager to help.

What this language is used for—

Although Python can be used to build pretty much anything, it really shines when it comes to working on technologies like Artificial Intelligence, Machine Learning, Data Analytics. Python also proves to be useful for web development, creating enterprise applications, and GUIs for applications.

JavaScript—Rich Interactive Web Development

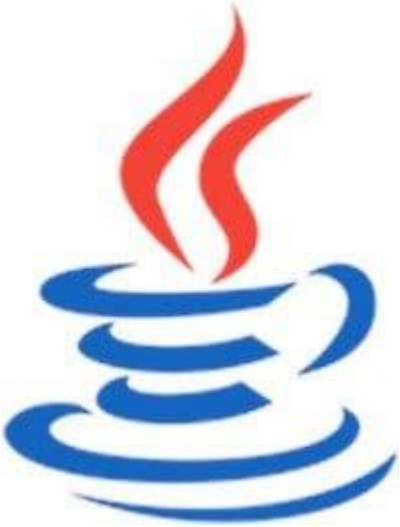
JavaScript was one of the key programming languages alongside HTML and CSS that helped build the internet. JavaScript was created in 1995 by Netscape, the company that released the famous Netscape Navigator browser, to eliminate the crudeness of static web pages and add a pinch of dynamic behavior to them.

Today, JavaScript has become a high-level multi-paradigm programming language that serves as the world's top frontend programming language for the web, handling all the interactions offered by the webpages, such as pop-ups, alerts, events, and many more like them.

What this language is used for—

JavaScript is the perfect option if you want your app to run across a range of devices, such as smartphones, cloud, containers, micro-controllers, and on hundreds of browsers. For the server-side workloads, there's Node.js, a proven JavaScript runtime that is being used by thousands of companies today.

Java—Enterprise Application Development



Java has remained the de-facto programming language for building enterprise-grade applications for more than 20 years now.

Created by Sun Microsystems' James Gosling in 1995, the object-oriented programming language Java has been serving as a secure, reliable, and scalable tool for developers ever since.
Some of the features offered by Java that make it more preferable than several other programming languages are its garbage collection capabilities, backward compatibility, platform independence via JVM, portability, and high performance.

Java's popularity can be seen clearly among the Fortune 500 members as 90% of them use Java to manage their business efficiently.

What this language is used for—

Apart from being used to develop robust business applications, Java has also been used extensively in Android, making it a prerequisite for Android developers. Java also allows developers to create apps for a range of industries, such as banking, electronic trading, e-commerce, as well as apps for distributed computing.

R—Data Analysis

If you do any sort of data analysis or work on Machine Learning projects, the chances are that you may have heard about R. The R programming language was first released to the public in 1993 by its creators Ross Ihaka and Robert Gentleman as an implementation of the S programming language with a special focus on statistical computing and graphical modeling.

Over the years, R became one of the best programming languages for projects requiring extensive data analysis, graphical data modeling, spatial and time-series analysis.

R also provides great extensibility via its functions and extensions that offer a ton of specialized techniques and capabilities to developers. The language also works remarkably well with code from other programming languages, such as C, C++, Python, Java, and .NET.

What this language is used for—

Apart from some of the uses mentioned above, R can be used for behavior analysis, data science, and machine learning projects that involve classification, clustering, and more.

C/C++—Operating Systems and System Tools

Believe it or not, the programming languages C/C++ were all the rage in the very late 20th century. Why?

It's because C and C++ are both very low-level programming languages, offering blazing fast performance, which is why they were and are still being used to develop operating systems, file systems, and other system-level applications.

While C was released in the 70s by Dennis Ritchie, C++, an extension to C with classes and many other additions, such as object-oriented features, was released later by Bjarne Stroustrup in the mid-80s.
Even after close to 50 years, both the programming languages are still being used to create rock-steady and some of the fastest applications of all times.

What this language is used for—
As C & C++ both offer full access to the underlying hardware, they have been used to create a wide variety of applications and platforms, such as system applications, real-time systems, IoT, embedded systems, games, cloud, containers, and more.

C#—Application & Web Development Using .NET

C# was Microsoft's approach to developing a programming language similar to the object-oriented C as part of its .NET initiative. The **general-purpose multi-paradigm programming language** was unveiled in 2000 by Anders Hejlsberg and **has a syntax similar to C, C++, and Java.**
This was a huge plus point for developers who were familiar with either of these languages. It also offered relatively faster compilation and execution along with seamless scalability.

C# was designed keeping in mind the .NET ecosystem, which allows developers to access a range of libraries and frameworks offered by Microsoft. And with the integration with Windows, C# becomes extremely easy to use, even perfect for developing Windows-based apps.

What this language is used for—
Developers can use C# for a range of projects, including game development, server-side programming, web development, creating web forms, mobile applications, and more. C# has also been used to develop apps for the Windows platform, specifically Windows 8 and 10.

PHP—Web Development

Just like Guido van Rossum's Python, PHP also came to fruition as a side project by Rasmus Lerdorf, with the initial development dating back to the year 1994.

Rasmus's version of PHP was originally intended to help him maintain his personal homepage, but over the years, the project evolved to support web forms and databases.

Today, PHP has become a general-purpose scripting language that's being used around the globe, primarily for server-side web development. It is fast, simple, and is platform-independent, along with a large open-source software community.

What this language is used for—
A large number of companies are using PHP today to create tools like CMS (Content Management Systems), eCommerce platforms, and web applications. PHP also makes it extremely easy to create web pages in an instant.

SQL—Data Management

SQL, short for Structured Query Language, is probably one of the most crucial programming languages on this list.

Designed by Donald D. Chamberlin and Raymond F. Boyce in 1974, the special-purpose programming language has played a key role in enabling developers to create and manage tables and databases for storing relational data over hundreds of thousands of data fields.

Without SQL, organizations would have to rely on older and possibly slower methods of storing and accessing vast amounts of data. With SQL, much of these tasks can be done within seconds. Over the years, SQL has helped spawn a large number of RDBMS (Relational Database Management Systems) that offer much more than just the creation of tables and databases.

What this language is used for—
Pretty much every other project or industry that needs to deal with large amounts of data stored in tables or databases uses SQL through an RDBMS.

Swift—For Mobile App Development on iOS

Apple's full control over its hardware and software has allowed it to deliver smooth and consistent experiences across its range of devices. And that's where Swift comes in.

Swift is **Apple's own programming language** that was **released in 2014** as a replacement for its Objective-C programming language. It is a multi-paradigm general-purpose programming language that's extremely efficient and designed to improve developer productivity.

Swift is a modern programming language (newest on this list), **fast, powerful, and offers full interoperability with Objective-C**. Over the years, Swift received numerous updates that helped it gain significant popularity among Apple's iOS, macOS, watchOS, and tvOS platforms.

What this language is used for—
Paired with Apple's Cocoa and Cocoa Touch framework, Swift can be used to create apps for virtually every Apple device, such as iPhones, iPads, Mac, Watch, and other devices.

Differences between natural language and programming language

Natural languages are spoken by people, while programming languages are intended for machines. Both languages contain important similarities, such as the differentiation they make between syntax and semantics and the existence of a basic composition. Essentially, the two types were created to communicate ideas, expressions, and instructions.
Whilst there are several other similarities and points in common between them, it is also possible to identify some of their differences.

Natural Languages	Programming Languages
More ambiguous. Human beings have the ability to clarify the meaning of an expression. The built-in redundancy of human languages allows some ambiguity to be resolved using context.	Stricter and less tolerant. Computers are very precise about the instructions they like to receive. Therefore, programming languages have practically no redundancy to prevent ambiguity and issue the correct commands.
Are open and allow combinations without the risk of making mistakes.	Are closed and fixed to avoid confusion and mistakes.

Programming languages: Types and Features

The evolution of computers has led to the creation of hundreds of different programming languages for various types of development. The field of programming is vast, so the use of a particular language will depend on the objectives to be achieved.

Types of programming languages

Programming languages can mainly be classified as low-level and high-level programming languages. Although simple compared to human languages, high-level languages are more complex than low-level languages. At the same time, a high-level language affords more readability in comparison to its low-level counterpart, which needs specialist knowledge in computer architecture to interpret.

- **Low-level languages** include assembly and machine languages.
- An **assembly language** contains a list of basic instructions and is much harder to read than a high-level language. It is just one level above machine code in terms of abstraction, using simple codes that are easily converted to strings of 1s and 0s (binary representation). It cannot be used to structure and manipulate complex information.
- **Machine language** is directly understood by the computer's processing unit. A programmer will first write his code in a high-level language, then compile it into a machine-readable format where instructions are represented in binary.
- **High-level languages**, on the other hand, are designed to be easy to read and understood, allowing programmers to write source code using logical, meaningful words and symbols. They encapsulate everything from early algorithmic languages such as FORTRAN to more widespread, object-oriented languages like C++, C#, and Java.

The following activities can be performed using high-level programming languages:

- Programs and applications development.
- Artificial intelligence development.
- Database development.
- Video game development.
- Development of drivers and hardware interface.
- Internet and web pages development.
- Script development.

Main features of programming languages

The popularity of a programming language depends on the features and utilities it provides to programmers. But, what are the main features of programming languages? Here are the main features that a programming language must possess to stand out from all the rest:

- **Simplicity:** the language must offer clear and simple concepts that are easy to understand, facilitating learning and application. But simplicity can be a difficult balance to strike without compromising the overall capability of the language.
- **Capability:** apart from being easy to use, the language must be well-equipped with a robust set of features to perform a wide range of tasks. If a programming language was designed to be used in a specific area, it must provide the necessary means (operators, structures, and syntax) to achieve ideal results.
- **Abstraction**: it is the language's ability to define and use complicated structures or operations while ignoring certain low level details.
- **Efficiency**: programming languages that can be translated and executed efficiently help avoid the excessive consumption of memory and time.
- **Structuring**: the language allows programmers to write their code according to structured programming concepts to avoid creating errors.
- **Compactness**: a language with this characteristic can express operations concisely without having to write too many details.
- **Principle of Locality**: also known as the **locality of reference**, this phenomenon describes a computer program's preference for continually accessing the same areas of memory over a short span of time. By enabling the usage of loops and subroutines, a programming language can exploit the principle of locality for optimising the overall performance of an application.