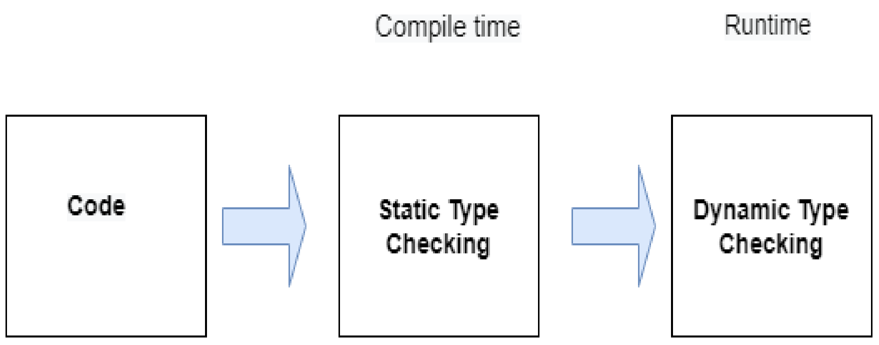
## 1.What is the difference b/w statically typed and dynamically typed language?

A data type in a programming language refers to a characteristic that defines the nature of the value that a data element has. Some common examples include string, integer, character, and Boolean.

Every programming language has a system of checking that values have been assigned their correct types, which is known as type checking. Type checking is essential in programming to minimize errors during the execution of programs. This usually occurs at either runtime or compile time. There are two categories of type checking implemented in most programming languages, static and dynamic:



### Statically Typed Languages

In statically typed programming languages, type checking occurs at compile time. At compile time, source code in a specific programming language is converted to a machine-readable format. This means that before source code is compiled, the type associated with each and every single variable must be known. Some common examples of programming languages that belong to this category are **Java, Haskell, C, C++, C#, Scala, Kotlin, Fortran, Go, Pascal, and Swift.**

### Dynamically Typed Languages

Conversely, **in dynamically typed languages, type checking takes place at runtime or execution time**. This means that variables are checked against types only when the program is executing. Some examples of programming languages that belong to this category are Python, JavaScript, Lisp, PHP, Ruby, Perl, Lua, and Tcl.

| **Statically typed language** | **Dynamically typed language** |
|---|---|
| Type checking is completed at compile time | Type checking is completed during runtime |
| Explicit type declarations are usually required | Explicit type declarations are not required |
| Errors are detected earlier | Type errors are detected later during execution |
| Produces more optimized code. | Produces less optimized code runtime errors are possible. |

## 2.Scripting language vs programming language:

A scripting language is a programming language designed specifically for runtime environments. It automates the execution of tasks. They are used in system administration, web development, games, and creating plugins and extensions. These languages are interpreted languages (An Interpreter executes instructions written in a programming or scripting language directly, without requiring them previously to have been compiled into a machine language program), and they bring new functionalities to the applications. These languages are usually short and snappy and are interpreted from the source or byte code. Mostly, scripting languages are open-sourced languages and are supported by almost every platform, which means that no special kind of software is required to run them, as they are a set of commands run without the use of compiler.

- Server-side scripting language: Server-side scripting languages are used to create dynamic web pages. They are executed at a web server. These languages perform backend operations. Examples- PHP, Python, Node.js, Ruby, and Pearl.

- Client-side scripting language: Client-side scripting language runs off the browsers. These languages are considered front end languages. Examples- HTML, jQuery, CSS and JavaScript.

## Key Features of Scripting Language

- Easy to learn and use- They are often a great starting point for those who want to step their foot in the world of development. They are easy to learn and implement. JavaScript and PHP are some of the easiest scripting languages.
- Open-source and free- Most of the scripting languages are open-sourced, which means there are no limits on who can utilize scripting languages. All they have to do now is study them and integrate them into their existing system. They're all open-source, which means anyone around the world can contribute to their development.
- Powerful and extensible- Scripting languages are powerful (in terms of application) enough so that the necessary tasks can be accomplished with the scripts. Scripting languages are also highly extensible, i.e., you can add certain features if you feel them necessary.

## Programming language:

A programming language is used to interact with computers to develop mostly used to build desktop apps, websites, and mobile apps. It's a sequence of instructions written to accomplish a given goal. C, C++, Java, Python are some of the examples of programming languages. Programming languages usually consist of two components-syntax(form) and semantics(meaning). They are used to implement algorithms and enable computers to perform actions.

- Machine language- Machine language is a type of low-level language that computers can understand easily. They are composed of instructions in binary or hexadecimal and are the elemental language of computers.
- Assembly language- Assembly language (ASM) is yet another low-level programming language created for specific processors. It is just a symbolic and human-understandable representation of the collection of instructions. It translates assembly language to machine language using an assembler.
- High-level language- HLL is a high-level programming language that is used to create user-friendly software and websites. This programming language necessitates the use of a compiler or interpreter to convert the code written by the user to the machine

language (language understood by the computer). A high-level language’s main advantage is that it is simple to read, write, and maintain. Python, Java, JavaScript, PHP, C#, C++, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift are examples of high-level programming languages.