* **[What is the difference between statically typed and dynamically typed languages?](https://stackoverflow.com/questions/1517582/what-is-the-difference-between-statically-typed-and-dynamically-typed-languages)**

|  |  |  |
| --- | --- | --- |
| **S.**  **NO** | **STATICALLY TYPED LANGUAGE** | **DYNAMICALLY TYPED LANGUAGE** |
| **1.** | Type checking is completed at compile time. | Type checking is completed during runtime. |
| **2.** | Explicit type declarations are usually required. | Explicit type declarations are not required. |
| **3.** | Errors are detected earlier. | Type errors are detected later during execution. |
| **4.** | Variable assignments are static and cannot be changed. | Variable assignments are dynamic and can be altered. |
| **5.** | Produce more optimize code. | Produce less optimize code, runtime errors are possible. |
| **6.** | Example language: Java, Haskell, C, C++, C#, Scala, Koltin, Fortran, Go, Pascal and Swift. | Example language: Javascript, Lisp, PHP, Ruby, Perl, Lua and Tcl. |

* [**Scripting Language vs Programming Language**](https://stackoverflow.com/questions/17253545/scripting-language-vs-programming-language)

|  |  |  |
| --- | --- | --- |
| **PARAMETERS** | **SCRIPTING**  **LANGUAGE** | **PROGRAMMING LANGUAGE** |
| **Language Type** | The scripting languages are interpreter-based languages. | The programming languages are compiler-based languages. |
| **Use** | The scripting languages help in combining the existing components of an application. | The programming languages help in developing anything from scratch. |
| **Running of Language** | A user needs to run scripting languages inside an existing program. Thus, it’s program-dependent. | Programming languages are program-independent. |
| **Conversion** | Scripting languages convert high-level instructions into machine language. | Programming languages help in converting the full program into the machine language (at once). |
| **Compilation** | You don’t need to compile these languages. | These languages first need a compilation. |
| **Design** | These make the coding process simple and fast. | These provide full usage of the languages. |
| **File Type** | Scripting languages don’t create any file types. | Programming languages create .exe files. |
| **Complexity** | These are very easy to use and easy to write. | These are pretty complex in terms of writing and usage. |
| **Type of Coding** | Scripting languages help write a small piece of an entire code. | Programming languages help write the full code concerning a program. |
| **Developing Time** | These take less time because they involve lesser code. | These take more time because a programmer must write the entire code. |
| **Interpretation** | We usually interpret a scripting language in another program. | The compile results of a programming language are stand-alone. No other program needs to interpret it. |
| **Requirement of Host** | Scripting languages require hosts for execution. | Programming languages are self-executable. They don’t require any host. |
| **Length of Codes** | These involve very few and short coding lines. | These require numerous lines of coding for a single function. |
| **Support** | These provide limited support to data types, user interface design, and graphic design. | These provide rich support for graphic design, data types, and user interface design. |
| **Maintenance** | These involve very low maintenance. | These involve high maintenance. |
| **Cost** | It is easier and cheaper to maintain a scripting language. | Maintaining a programming language is comparatively more expensive. |
| **Example** | VB Script, Perl, Ruby, PHP, JavaScript, etc. | C, C++, COBOL, Basic, VB, C#, Pascal, Java, etc. |

**Programming paradigm:-**

A Programming paradigm is the classification style or way of programming. It is an approach to solve problems by using programming languages.

The programing paradigms are categorized in multiple categories yet most significant are only two:

* **Imperative programming paradigm**
* **Declarative programming paradigm**

**Imperative programming paradigm :-**

The programmer instructs the machine how to change its state.

Some of the programming languages that support the imperative paradigm are: C, C++, Java, PHP, Ruby, scala, pascal

* Procedural programming paradigm,
* Object Oriented Programming (OOP),
* Parallel processing.
* **Procedural programming paradigm**

It is the same as imperative programming but with a procedure call that lets you reuse the code. And this feature was an amazing advancement at that time. Example C, C++, Java, etc.

* **Object Oriented Programming (OOP)**

This is used to work on real-world entities in form of class and objects. Class is the blueprint of the object and you can replicate as many as the object you want. These classes contain some properties and method which all are replicated in objects. Example C++, Python, etc.

* **Parallel processing**

In this type of programming, a program is processed by dividing it into multiple processors. The system contains multiple processors to solve the problem in less time.

**Declarative programming paradigm :-**

Properties of desired result, but not how to compute it.

Some of the programming languages that support the imperative paradigm are: Prolog, Javascript, scala, Lisp, SQL, XQuery, Clojure.

* + **Logic programming paradigm,**
  + **Functional programming paradigm**
  + **Database programming approach**
* **Logic programming paradigm**

Logic programming uses sentences in logical form and creates an expression by using symbols. In machine learning and artificial intelligence, there are many models that use these programs. The programs are executed very much like some mathematical statement. It is mainly based on forming logic.

* **Functional programming paradigm**

Functional programming is the programming in which a program is constructed by creating and using functions. Rather than a series of statements functional programming use function to map and change one value to another value. Functional programming is the key feature of Javascript.

* **Database programming approach**

This programming is based on enquiring data, its modification, its movement, etc. The most famous programming language that supports this is SQL.