# **COMPUTER PROGRAMMING**

**Computer programming** is the process of designing and building an executable program to accomplish a specific computing result or to perform a specific task.

## **Types of programming languages**

1. Procedural Programming Language.
2. Functional Programming Language.
3. Scripting Programming Language.
4. Logic Programming Language.
5. Object-Oriented Programming Language.

### **Procedural programming language**

### A **procedural language** is a computer [programming language](https://www.computerhope.com/jargon/p/programming-language.htm) that follows, in order, a set of commands. Examples of computer procedural languages are [BASIC](https://www.computerhope.com/jargon/b/basic.htm), [C](https://www.computerhope.com/jargon/c/c.htm), [FORTRAN](https://www.computerhope.com/jargon/f/fortran.htm), [Java](https://www.computerhope.com/jargon/j/java.htm), and [Pascal](https://www.computerhope.com/jargon/p/pascal.htm).

### Procedural languages are some of the common types of programming languages used by script and software programmers. They make use of [functions](https://www.computerhope.com/jargon/f/function.htm), [conditional statements](https://www.computerhope.com/jargon/c/contstat.htm), and [variables](https://www.computerhope.com/jargon/v/variable.htm) to create programs that allow a computer to calculate and display a desired output.

|  |  |  |
| --- | --- | --- |
| **Procedural language** | **Year it was founded** | **Founder** |
| Basic | 1964 | John George Kemeny and Thomas E. Kurtz |
| C | 1972 | Dennis Ritchie and Brian Kernighan |
| Fortran | 1959 | John Backus |
| Java | 1991 | James gosling |
| Pascal | 1970 | Niklaus Wirth |

### **Functional programming language**

Functional programming languages are specially designed to handle symbolic computation and list processing applications. Functional programming is based on mathematical functions. Some of the popular functional programming languages include**:** LISP, Python, Erlang, Haskell, Clojure, etc.

* **Pure Functional Languages** − These types of functional languages support only the functional paradigms. For example − Haskell.
* **Impure Functional Languages** − These types of functional languages support the functional paradigms and imperative style programming. For example – LISP

.

|  |  |  |
| --- | --- | --- |
| **Functional language** | **Year it was founded** | **Founder** |
| Lisp | 1960 | John McCarthy |
| Python | 1991 | Guido van Rossum |
| Erlang | 1986 | Agner Krarup Erlang |
| Haskell | 1990 | Philip Wadler and Stephen Blott (they proposed) |
| Clojure | 2005 (designed)  2007 (released) | Rich Hickey |

### **Scripting programming language**

A **scripting language** or **script language** is a [programming language](https://en.wikipedia.org/wiki/Programming_language) for a [runtime system](https://en.wikipedia.org/wiki/Runtime_system) that automates the [execution](https://en.wikipedia.org/wiki/Execution_(computing)) of tasks that would otherwise be performed individually by a human operator.[[1]](https://en.wikipedia.org/wiki/Scripting_language#cite_note-ecma262-1) Scripting languages are usually [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)) at [runtime](https://en.wikipedia.org/wiki/Runtime_(program_lifecycle_phase)) rather than [compiled](https://en.wikipedia.org/wiki/Compiler). Scripting languages are a popular family of programming languages that allow frequent tasks to be performed quickly. Early scripting languages were generally used for niche applications – and as “glue languages” for combining existing systems. With the rise of the World Wide Web, a range of scripting languages emerged for use on web servers. Since scripting languages simplify the processing of text, they are ideally suited to the **dynamic generation of HTML pages**. Examples include PHP, ruby, groovy, perl, LUA, etc.

|  |  |  |
| --- | --- | --- |
| **Scripting language** | **Year it was founded** | **Founder** |
| PHP | 1994 | Rasmus Lerdorf |
| ruby | 1995 | Yukihiro Matsumoto |
| groovy | 2003 | James Strachan |
| perl | 1987 | Larry Wall |
| LUA | 1993 | Roberto Lerusalimschy  Waldemar Celes  Luiz Henrique de Figueiredo |

### **Logic Programming language**

**Logic programming** is a programming paradigm which is largely based on formal logic. Any program written in a logic programming language is a set of sentences in logical form, expressing facts and rules about some problem domain.It can also be described as a [computer programming](https://www.computerhope.com/jargon/p/programming.htm) paradigm where [program](https://www.computerhope.com/jargon/p/program.htm) [statements](https://www.computerhope.com/jargon/s/statemen.htm) express facts and rules about problems within a system of formal logic.

Some logic [programming languages](https://www.computerhope.com/jargon/p/programming-language.htm), such as [Datalog](https://www.computerhope.com/jargon/d/datalog.htm) and ASP (Answer Set Programming), are purely declarative. They allow for statements about what the program should accomplish, with no explicit step-by-step instructions on how to do so. Others, such as [Prolog](https://www.computerhope.com/jargon/p/prolog.htm), are a combination of declarative and imperative.

|  |  |  |
| --- | --- | --- |
| **Logic language** | **Year it was founded** | **Founder** |
| Datalog | 1977 | Hervé Gallaire , Jack Minker |
| ASP | 1998 | David Frost |
| Prolog | 1972 | Alain Colmerauer |

### **Object-Oriented Programming Language.**

**Object Oriented programming** (**OOP**) is a **programming** paradigm that relies on the concept of classes and **objects**. ... These functions are defined within the class and perform some action helpful to that specific type of **object**. For **example**, our Car class may have a method repaint that changes the color attribute of our car. In **object**-**oriented programming** data structures, or **objects** are **defined**, each with its own properties or attributes. Each **object** can also contain its own procedures or methods. Software is designed by using **objects** that interact with one another.