Knowledge Leveling

Programming Languages Overview

PROGRAMMING LANGUAGES

What is a Programming Language?

A **programming language** is a formal language that specifies a set of instructions for a computer to produce different kinds of data. Programming languages can be used to create programs that implement specific algorithms that control the physical and logical behavior of a computer.

Most used programming languages:

Python
C
Java
C#
Javascript

Programming languages are languages created by humans to communicate with computers. Thus we could say that a programming language is the set of symbols and words that allow the user of a computer to give instructions and commands for the computer to execute them.

A programming language consists of a set of syntactic and semantic symbols and rules that define its structure and the meaning of its elements and expressions.

The process by which a computer program's source code is written, tested, debugged, compiled (if necessary), and maintained is called **"programming"**.

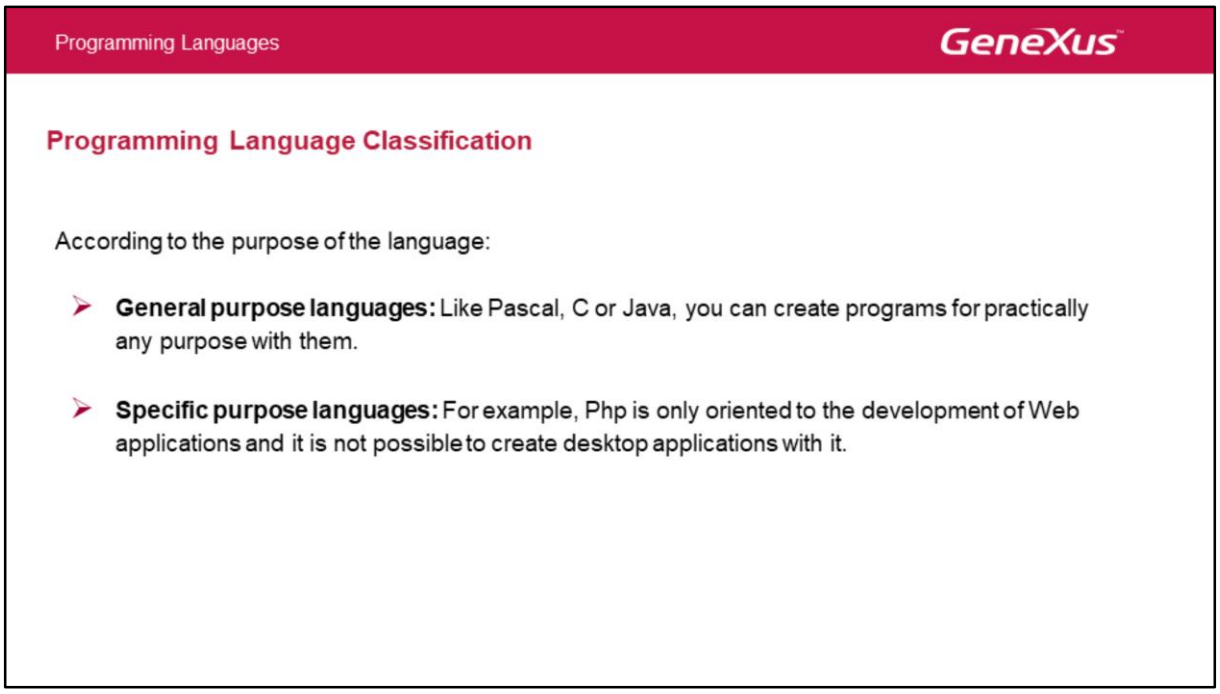
- **PYTHON:** Simple and easy to use. It has a large number of libraries that other programming languages don't have.
- **C:** This programming language is one of the most popular among the developer community. Its efficiency makes it a tool from which you can mix with other languages as an assembler, or directly access the computer hardware.
- **JAVA:** Designed to be a multiplatform language, Java is the language from which applications are made in Android.
- **C#:** This language was developed by Microsoft in early 2000. According to many developers, it is an alternative to C and C+++, positioning itself as the best of this group. It is defined as a set of the best features of C, C++, Java and other languages, and is mainly used for web, desktop and telephony.
- **JAVASCRIPT:** One of the most widely used and preferred programming languages by the worldwide developer community. JavaScript is a web language for objects used to develop on the client side.

Programming Language Classification

Depending on their proximity to machine language:

- **Machine Language:** is the programming language that the machine (computer) understands. This programming language uses the binary alphabet, i.e. 0 and 1.
- **Low-level programming languages:** They are much easier to use than machine language, but they depend a lot on the machine or computer as was the case with machine language.
- **High-level programming languages:** They are very similar to human language, but they require an interpreter or compiler program that translates this high-level programming language into a low-level one such as machine language that the computer can understand.

- **Machine language:** The so-called binary strings are formed with only two digits, also known as bits (combinations of zeros and ones). They are used to write instructions, and through these instructions the computer microprocessor understands our requests. Machine language was the first programming language. This programming language stopped being used because of its great difficulty and how easy it was to make mistakes when writing binary strings.
- **Low-level languages:** Languages of this type can create very fast programs, but they are difficult to learn, they are specific to each processor (of each machine), and if we take the program to another computer it will be necessary to rewrite the program from scratch.
- **High-level languages:** High-level programming languages are easier to learn because they use words or commands from natural language, usually English. This is the case of **BASIC**, the most popular programming language.



There are other criteria for classifying programming languages:

Classification of the programming languages according to their execution method:

- **Compiled languages** translate the source code of the program into machine code or object code. Examples: C, Pascal
- **Interpreted languages** execute the instructions of a program line by line. They require source code to run the program. Examples: Perl, Lisp.

Classification of programming languages according to their Programming Paradigm:

- **Imperative languages** are abstractions from some sequence of instructions that specify in detail the order in which the program is executed. Examples: Fortran, Algol, Ada, Pascal, C, C++.
- **Declarative languages** express what the program should accomplish without prescribing how to do it, in terms of sequences of actions to be taken. Examples: SQL, HTML, RPG.
- **Functional Languages** consist of a set of pre-defined functions. Examples: Lisp, Scheme, Common Lisp, ML, CAML.
- **Logic languages** express tasks using formal mathematical logic. Example: Prolog.
- **Object-Oriented Languages** create a system of classes and objects following the real world scheme to define objects, actions and how they communicate between objects. Examples: C++, Java.

GeneXus™

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certificactions	training.genexus.com/certifications