Interpreted languages

Interpreted languages are programming languages in which programs may be executed from source code form, by an interpreter. Theoretically, any language can be compiled or interpreted, so the term interpreted language generally refers to languages that are usually interpreted rather than compiled.

Ant

APL

AutoHotkey scripting language

AutoIt scripting language

BASIC (some dialects)

Programming Language for Business (PL/B, formerly DATABUS, later versions added optional compiling)

Eiffel (via Melting Ice Technology in EiffelStudio)

Emacs Lisp

FOCAL

GameMaker Language

Groovy

J

Julia (compiled on the fly to machine code, but a transpiler Julia2C exists)

JavaScript

Lisp (early versions, pre-1962, and some experimental ones; production

Lisp systems are compilers, but many of them still provide an interpreter if needed)

LPC

Lua

MUMPS (an ANSI standard general-purpose language)

Maple

Mathematica

MATLAB

OCaml

Pascal (early implementations)

PCASTL

Perl

PHP

PostScript

PowerShell

PROSE

Python

Rexx

R

REBOL

Ring

Ruby

S-Lang

Speakeasy

Standard ML (SML)

Spin

Tcl

Tea

TorqueScript

thinBasic scripting language

VBScript

Windows PowerShell – .NET-based CLI

Wolfram Language

Some scripting languages – below