

# Concept: Programming language

## Headline

A [software language](#) in which [programs](#) are written

## Illustration

Have a look at [Hello world programs](#) in different programming languages.

## Metadata

- [http://en.wikipedia.org/wiki/Programming\\_language](http://en.wikipedia.org/wiki/Programming_language)
  - [Vocabulary:Programming](#)
  - [Vocabulary:Software engineering](#)
  - [Software language](#)
-

# Concept: Software language

## Headline

A software language

## Quote

The following quote, which was extracted on 1 March 2011 from the website <http://www.sleconf.org/2011/> of "The International Conference on Software Language Engineering (SLE 2011)", serves as the approximation of a description: *"The term 'software language' comprises all sorts of artificial languages used in software development including general-purpose programming languages, domain-specific languages, modeling and meta-modeling languages, data models, and ontologies. Used in its broadest sense, examples include modeling languages such as UML-based and domain-specific modeling languages, business process modeling languages, and web application modeling languages. The term 'software language' also comprises APIs and collections of design patterns that are implicitly defined languages."*

## Metadata

- <http://www.sleconf.org/2011/>
  - [Vocabulary:Software language engineering](#)
-

# Concept: Program

## Headline

An executable [software artifact](#) that solves a certain problem

## Description

According to a classic definition, a program is ([Document:Principles of information systems](#)) "a sequence of instructions written to perform a specified task with a computer". This style of definition is possibly too much focused on an imperative view of programming.

More intuitively, more inclusively, and shorter: **a [program](#) is an executable [software artifact](#) that solves a certain problem (that is amenable to automation on a computer). For instance, a program may solve an [algorithmic problem](#).**

A [program](#) may count as a "small" [software system](#) or an (executable) [software component](#). A "proper" [software system](#) or [component](#) typically comprises of multiple [software artifacts](#) that may be elements of different [software languages](#), may or may not be elements of [programming languages](#), may reside at different levels of abstraction, and may interact in various ways.

Strictly speaking, a [program](#), as far as this term is used in practice, may very well also break down into multiple [software artifacts](#) because of, for example, [modular programming](#). Thus, the line between [programs](#) and [software systems](#) or (executable) [software components](#) is somewhat blurred.

## Illustration

See the [Hello world program](#) for a very simple program.

## Metadata

- [http://en.wikipedia.org/wiki/Computer\\_program](http://en.wikipedia.org/wiki/Computer_program)
  - [Software artifact](#)
  - [Vocabulary:Programming](#)
  - [Vocabulary:Software engineering](#)
-