

Programming Paradigms

Programming paradigms are not languages or tools. They are sets of ideas & guidelines. A language is not tied to a specific paradigm. There are multi-paradigm languages.

Classification

Imperative

- specifies how computation is done
- There is a notion of program state and statements can change that state
- procedural
 - organized by procedures that call each other. i.e. Fortran, C, BASIC, Rust
- object-oriented
 - organized as objects that data & associated behaviors
 - C++, Java, Python

Declarative

- specify what computation is to be done, declaring properties of the desired effect.
- functional
 - a result is declared via a series of function evaluations, Avoids state & mutable data i.e. Lisp
 - based on lambda calculus (Alonzo Church)
- logic
 - a result is declared as an answer to a system of rules. (Predicate logic) i.e. Prolog

1940's - first electronic computers (Assembly)

→ 1949 - first high level language implemented on a computer

- Short Code - 50x slower than assembly

1950's

- 1953 - Fortran (Formula translation)
- 1958 - Lisp (list processor)
- 1959 - Algol
- 1959 - Cobol • business use

1960's

- BASIC (Beginners all-purpose symbolic instruction code)
 - Installed on nearly all PCs in the 70's & 80's.
- 1962 - Simula • first object-oriented language

1970's

- Pascal, derived from Algol. Was the intro CS language for many undergraduates in the 80's.
- 1972 - Prolog • one of the first logic languages
- 1972 - C, influenced by Algol General-purpose
- 1974 - SQL (structured query lang.) - databases
 - lex/yacc

1980's

- 1980 - C++ (C with classes), influenced by Simula
- 1984 - Matlab (matrix laboratory) used by physics & engineering students
- 1987 - Perl • sys admin
- 1989 - Bash
 - Unix shell

1990's

- 1990 - Haskell • purely functional
- 1991 - Python
- 1991 - Visual Basic • Drag & Drop
- 1995 - Ruby, Java, PHP, Javascript

2000's → 2002 - Scratch

2010's → 2010 - Rust

- critical systems

LISP

$(+ 1 (- 4 2))$

C

$$y = 4 - 2$$
$$x = 1 + y$$