

PHP

(Personal Home Page Tools あるいは PHP: Hypertext Preprocessor の再帰的頭字語 recursive acronym) (参考: 再帰的頭字語の例 Linux → Linux is not Unix ; 開発者 Linus Torvalds)

スクリプト言語 script language, scripting language

スクリプト言語とは、プログラミング言語のうち、プログラムの記述や実行を比較的簡易に行うことができる言語の総称である。

非常に多くの場合、スクリプト言語はインタプリタ型言語であり、コンパイラ型言語に比べて実行までの処理の手間がかからないという特徴を持っている。また、他のプログラミング言語に比べると、英語のような自然言語に近い記述を用いてプログラミングすることが可能であり、習得が比較的容易であるとされる。

スクリプト言語の中には、デスクトップアプリケーションで処理を自動化するために用いられるマクロ言語なども含まれるが、通常、スクリプト言語と言った場合は、Web ブラウザ上で動作する簡易プログラムを記述するための言語を指す場合がほとんどである。代表的なスクリプト言語としては、JavaScript、Perl、Python、PHP、Rubyなどを挙げることができる。

インタプリタ interpreter

プログラミング言語で記述されたソースコードをオブジェクトコードに変換しながらそのプログラムを実行するソフトウェアのこと。インタプリタ型の言語は、プログラムの実行時に変換を行うため、コンパイラ言語に比べて動作が遅い。

PHP はソースコードをコンパイルしてから実行するため純粋なインタプリタとは言えないが、動きはインタプリタに似ている。

コンパイラ compiler

何らかのプログラミング言語で記述されたプログラムをコンピュータが実行可能なコードに変換し、その後実行する言語システムを一般にコンパイラ言語と呼ぶ。コンパイラは、その中でコード変換を行う中核部分もしくは、その技術の名称である。

一般に人間が理解しやすい表現形式を持つ言語を「高級言語」と呼ぶが、この高級言語をコンピュータが実行するには、大きくコンパイラ方式とインタプリタ方式の2つがある。前者は、ソースプログラムを読みこみ、実行可能な機械語からなるバイナリプログラムを出力し、その後実行を行う。インタプリタ方式では、ソースプログラムを読みこみながら、解釈を行い、直接実行を行う。一般にコンパイラ言語のほうが実行効率がよいとされているが、中間言語方式など、両者の方法を取り入れたものなどあれば、言語仕様によってはインタプリタでも高速な実行が可能な場合もある。

アセンブリ言語 assembly language

コンピュータ、マイクロコントローラ、その他のプログラム可能な機器を動作させるための機械語を人間にわかりやすい形で記述する、代表的な低水準言語である。なお、英語の assembly とは組立を意味し日本語の意味では組立言語となる。

アルゴリズム algorithm

アルゴリズムとは、ある特定の問題を解く手順を、単純な計算や操作の組み合わせとして明確に定義したもの。数学の解法や計算手順なども含まれるが、IT の分野ではコンピュータにプログラムの形で与えて実行させることができるよう定式化された、処理手順の集合のことを指すことが多い。四則演算や比較、条件分岐、繰り返しなど、曖昧さのない単純で明確な手順の組み合わせとして記述され、必ず有限回の操作で終了する。数値の列を大きい順または小さい順に並べ替える(整列アルゴリズム)、たくさんのデータの中から目的のものを探し出す(探索アルゴリズム)、といった基本的なものから、画像の中に含まれる人間の顔を検出する、といった複雑なものまで様々な種類のアルゴリズムがある。

同じ問題を解くアルゴリズムが複数存在することもあり、必要な計算量や記憶領域の大きさ、手順のシンプルさ、答えの精度などがそれぞれに異なり、目的に応じて使い分けられる。例えば、原理が単純で簡単にプログラムを記述できるが性能は低いアルゴリズム、計算手順が少なく高速に実行できるが膨大な記憶領域を必要とするアルゴリズム、厳密な解を求めるものより何桁倍も高速に近似解を求めることができるアルゴリズムなどがある。

変数 variable

変数とは、プログラムのソースコード中で、データを一時的に記憶しておくための領域に固有の名前を付けたもの。変数につけた名前を変数名と呼び、記憶されているデータをその変数の値という。データの入れ物のような存在で、プログラム中で複数のデータを扱いたいときや、同じデータを何度も参照したり計算によって変化させたい場合に利用する。

変数をプログラム中で利用するには、自分がこれからどんな変数を利用するかを宣言し、値を代入する必要がある(宣言が不要な言語や処理系もある)。変数に格納された値を利用したいときは、変数名を使って参照する。多くの言語では、整数、実数、文字列など、変数に利用できるデータの種類(データ型)が決まっており、宣言や代入の際にその変数がどのデータ型かを指定する必要がある。

定数 constant

定数とは、プログラミングにおいて、特定の値に名前をつけたもの。変数と異なり、宣言時に決めた値をコードの中で後から変更することができない。多くの言語では“constant”を縮めた“const”というキーワードで宣言する。

プレースホルダ placeholder

プレースホルダとは、実際の内容を後から挿入するために、とりあえず仮に確保した場所のこと。また、そのことを示す標識などのこと。

プログラミングでは、式や命令文などの一部にユーザの入力値など実行時に外部から与えられる要素を反映させるような場合に、それが入る予定の場所(や、そのことを示す特殊な記号や識別子など)のことをプレースホルダという。