

## 第2回 プログラミング言語について

プログラミング言語の役割について学ぶ。

### 【ハードウェアとソフトウェア】

パソコン、スマホ、テレビ、冷蔵庫、洗濯機、インターフォン、ウォシュレット、その他あらゆる電子機器には、ハードウェアとソフトウェアがある。

ハードウェア (hardware) とは、物理的な構成要素である。たとえば、モニタ、キーボード、マウス、CPU (中央処理装置)、メインメモリ (主記憶装置)、外部記憶装置 (ハードディスク、SSD、USB メモリ)、ケース、メインボード (マザーボード)、などである。

ソフトウェア (software) とは、ハードウェアとしてのコンピュータを動作させるための、プログラム、データなどの総称<sup>1</sup>であり、ハードウェアのような物理的実体はない。コンピュータはソフトウェアの指示に従って、その目的とする処理を行う。ソフトウェアを取り替えることで、コンピュータはさまざまな処理を行うことができる。「コンピュータ、ソフトがなければ ただの箱」と言われるように、コンピュータにとってソフトは必要不可欠なものである。

### 【プログラム】

プログラミング言語とは、コンピュータに理解できるよう、動作手順を表現してコンピュータに伝える人工的な言語であり、動作手順をプログラミング言語の言葉として表現したものがプログラムである<sup>2</sup>。

### 【基本ソフトウェア】

基本ソフトウェア (OS: Operating System) とは、コンピュータを利用するための基本的な制御やプロセス管理などを行うソフトウェアであり、基本ソフトウェアの上で応用ソフトウェアが動く。基本ソフトウェアは、ハードウェアを制御する基本的なソフトウェアを用意して、応用ソフトウェアとハードウェアとの仲立ちをする<sup>3</sup>。

### 【パソコンの主な OS】

- MS-DOS (Microsoft Disk Operating System)
  - Microsoft 社が 1981 年に開発した IBM PC 用の OS。
  - その後、バージョンアップを重ね、IBM PC/AT 互換機パソコン用 OS の業界標準となった。
- macOS
  - Apple 社のパソコン Macintosh 用の OS。OS の名称は Mac OS, Mac OS X, OS X などと変更され、2016 年に開発されたバージョン 10.12 から macOS となった。
  - 早くから優れた GUI を備え、Windows などに大きな影響を与えた。
- Windows
  - Microsoft 社が MS-DOS の後継として開発した OS。
  - 1986 年に最初のバージョン 1.0 が発表された後、3.1, 95, 98, XP, Vista, 7, 8, 10,

---

<sup>1</sup> JIS の定義では、ソフトウェアは「データ処理システムを機能させるための、プログラム、手順、規則、関連文書などを含む知的な創作」とされており、この場合、規則や関連文書も「ソフトウェア」に含まれる。

<sup>2</sup> ソフトウェアは、コンピュータに関わるすべての知的情報で、プログラムは、コンピュータの動作手順を表したものに限定されるため、ソフトウェアの方がプログラムよりも広い概念で、プログラムは、ソフトウェアに含まれる。

<sup>3</sup> JIS の定義では、「プログラムの実行を制御するソフトウェアであって、資源割振り、スケジューリング、入出力制御、データ管理などのサービスを提供するもの」とされている。

11 などバージョンアップを重ねている。

- UNIX
  - AT&T のベル研究所で 1969 年に開発されたワークステーション用 OS。
  - 大学や研究機関を中心に広く普及。
- Linux (リナックス)
  - フィンランドの大学生リーナス・トーバルズが、UNIX を真似て原型を開発。
  - オープンソースソフトウェアとして公開され、改変が自由に行われている。

#### 【スマホの主な OS】

- Android
  - Google が開発している OS で、スマホのメーカーがカスタマイズしている。
- iOS
  - Apple が iPhone や iPad などの自社製品のために開発している OS。

#### 【応用ソフトウェア】

応用ソフトウェア (アプリケーションソフトウェア application software; アプリ app) とは、一般のユーザーが利用する目的で作られたプログラムである。プログラムの開発元が作成したアプリをユーザーがパソコンあるいはスマホの OS にインストールし (あるいは、製品購入時にすでにインストールされている)、実行する。たとえば次のようなものがある。

1. ワープロソフトウェア
2. 表計算ソフトウェア
3. データベースソフトウェア
4. プレゼンテーションソフトウェア
5. ウェブブラウザ
6. グラフィックスソフトウェア
7. ユーティリティソフトウェア
8. 業務ソフトウェア
9. ゲームソフトウェア

誰かが作ったプログラムをインストールして実行するだけでなく、自ら応用ソフトウェアのプログラミングをして実行することもできる。この授業では自らが作成したプログラムを実行するという経験をする。

#### 【プログラミング言語の種類】

1. 機械語 (マシン語) : CPU が解釈し実行できる唯一の言語。機械語のプログラムは、CPU の 1 単位の動作を表現する命令を並べたもの。
2. アセンブリ言語 : 機械語の命令を人間に分かりやすいような表現で記述したもの。アセンブラにより、機械語に変換してから実行される。
3. 高水準言語 : 自然言語に近く、人間の考えを表現しやすいプログラミング言語。英語に近い記述ができる。

高水準言語によって書かれたプログラムの実行方法には、コンパイラによって機械語のプログラム (実行ファイル) に変換してから実行する方法と、インタプリタを起動して、インタプリタにプログラム (ソースファイル) を読み込ませて実行する方法がある<sup>1</sup>。

<sup>1</sup> 最初はインタプリタとして動作して、コンパイルが完了すると高速に実行する動的コンパイラなどの手法もある。

## 【主な高水準言語】

特に有名なものを選んで簡単に紹介する。

1. FORTRAN：1954年にIBMが開発。コンピュータにおいて広く使われた最古の高水準言語であり、特に科学技術計算の分野では今でも利用されている言語。
2. LISP：1958年に計算機科学者のジョン・マッカーシーが考案。現在広く使われているプログラミング言語の中では、FORTRANに次いで古い。S式というリスト構造の表記法を使うことが特徴であり、そのために大量の括弧を使うところが印象的である。
3. COBOL：1960年にCODASYLが開発。汎用コンピュータでの事務処理用として現在も広く利用されている言語。
4. BASIC：1970年代以降にパソコン用の言語として広く使われた言語。Visual Basicなどへ発展し、Microsoft OfficeにはVBA (Visual Basic for Applications)として搭載されている。
5. Pascal：1970年に構造化プログラミングとして設計され、特に1980年代に人気を博した言語。Turbo PascalやDelphiなどの統合開発環境が発売された。
6. C：1972年に開発されてUNIXの記述に用いられるようになり、現在も幅広く利用されている言語。ポインタによってメモリを直接指定してアクセスするといったような、高水準言語の中では低水準な処理ができた。その後、様々なC系言語が派生した。
7. Smalltalk：オブジェクト指向言語として最初に普及した言語。1972年に開発が開始され、1980年に公開された。
8. C++：C言語をベースにして、オブジェクト指向などの様々な機能を取り込んだ言語。C言語の上位互換とされるが、厳密には異なる。1983年に公開された。
9. Objective-C：1983年にBrad Coxらが開発。その後スティーブ・ジョブズのNeXTコンピュータの主力言語となり、権利を買い取る。さらにアップルがNeXT社を買収し、Mac OS XのCocoaフレームワークのコア言語として採用した。
10. Perl：ラリー・ウォールが1987年に開発したスクリプト言語。ウェブアプリケーション、システム管理、テキスト処理などのプログラムを書くために現在も広く用いられている。非常に高いレベルで後方互換性が維持されることで定評がある。
11. Haskell：純粋関数型プログラミング言語。習得の難易度は高いがバグが発生しにくいとされている。1987年にアメリカのポートランドで開かれた関数型プログラミング言語に関する学術会議で委員会が結成され、1990年に作成された。
12. Python：1991年に公開されたスクリプト言語。読みやすく簡潔にコードが書けるように設計されている。この授業で採用するプログラミング言語であるため、この後にさらに詳しく紹介する。
13. Ruby：1995年にまつもとゆきひろが開発したスクリプト言語。日本で開発されたプログラミング言語としてははじめて国際電気標準会議で国際規格に認証された。ウェブアプリケーションフレームワークのRuby on Railsが広く使われている。
14. Java：1995年にSun Microsystemsが開発し、Oracleが開発を引き継いでいる。SmalltalkとC++をベースとした、機種非依存のオブジェクト指向言語。企業の業務システム、PCやスマホのアプリ、家電製品の組み込み、など幅広い用途に利用される。環境を選ばない汎用性の高さから需要が高く、システム開発では最も使われている言語の一つである。
15. JavaScript：1995年にNetscape Communicationsが開発し、Netscape Navigatorというウェブブラウザで実装されたスクリプト言語。主要なウェブブラウザ上で動作するため、動的なウェブサイト構築やウェブサイト上でのアプリケーション開発に用いられる。Javaと混同されがちであるがまったく異なる。
16. PHP：ラスマス・ラードフが1995年に公開したスクリプト言語。ウェブアプリケーション開発で使われている言語。レンタルサーバーで動作するところが多く、初心者が習得しやすいと思われることから、一定の人気を保っている。

17. C#: Microsoft が 2000 年に開発し、C、C++、Java、Delphi 他多くの言語の影響を受けている。Microsoft のフレームワーク .NET Framework で動作する。
18. Scratch: 子供に楽しくプログラミングを学習させることを目的に 2006 年に MIT メディアラボが開発した。パレットにブロックを並べていくというインターフェイスを採用しているため、敷居が低くなっている。プログラミングの必修化にともない、日本でも教育業界で普及し、学校の授業でも取り入れられている。
19. Rust: C 言語、C++に代わるシステムプログラミング言語として、性能、メモリ安全性、安全な並行性を目指して設計された。2006 年に Mozilla 従業員の Graydon Hoare が開発を始め、2009 年から Mozilla Research の公式プロジェクトとなった。
20. Go: 2009 年に Google が発表した。golang とも呼ばれる。並行処理機能が優れていることなどからクラウドアプリ開発での人気が高い。
21. 2010 年代前半に開発された Dart, TypeScript, Swift, Kotlin, Julia は、いずれもこれまでの多くのプログラミング言語の良い特徴を取り入れて使いやすくされていることから人気が高く、今後の発展が期待されている。いずれもオープンソースである。
22. クロスプラットフォームアプリ開発フレームワークとして、Dart ベースの Flutter, JavaScript, TypeScript ベースの React Native, Cordova, Ionic, C#ベースの .NET MAUI (旧 Xamarin), Unity などがよく使われている。

#### 【Python について】

Python (パイソン) は、ガイド・ヴァン・ロッサム<sup>1</sup>が 1991 年に公開したプログラミング言語であり、世界中のプログラマーのコミュニティによって開発が続けられ、2001 年には非営利団体 Python ソフトウェア財団 (PSF) が立ち上がり、フリーソフトウェアの PSF ライセンスで配布されている。

Python は多くの環境で動作し、簡潔にコードを書けることが特徴として挙げられる。科学計算でよく使われ、数値計算ライブラリの NumPy や科学計算ライブラリの SciPy、データ解析ライブラリの pandas、グラフ描画ライブラリの Matplotlib などを使うことで、高度なデータ処理や統計解析をすることができる。

特に、近年の AI ブームは Python の人気上昇を加速させている。Google では、社内の公式言語として C/C++、Java、JavaScript、Python、Go が使われている。Google が開発しオープンソースで公開している TensorFlow という機械学習のためのソフトウェアライブラリが、AI のプログラミングに便利である。

このように学術的な分野で人気が高い Python であるが、実務でも Django のような Web アプリケーションフレームワークによって Web サイトの構築ができることから、よく使われている。YouTube、Instagram、Dropbox は Python で開発されたとされている。

この授業では、Scratch のような教育用のプログラミング言語を使わずに、Python を学習する。この授業をきっかけとしてプログラミングを本格的に学び、これからの大学での学習や就職後に使うスキルとして役立てるためには、よく使われている言語で習熟する方が良いと考えるためである<sup>2</sup>。まずは Python で簡単なプログラミングができるようになることで、必要に応じて他の言語も自習により習得できるようになるであろう。

#### 【★課題】

ToyoNet-ACE の小テストを提出期限までに受験すること。提出期限をすぎた場合は理由の如何を問わずに提出を受け付けない。

<sup>1</sup> Guido van Rossum – Personal Home Page <https://gvanrossum.github.io/>

<sup>2</sup> プログラミング言語の人気については <https://redmonk.com/> / <https://www.tiobe.com/tiobe-index/> / <https://spectrum.ieee.org/> / <https://youtu.be/Og847HVwRSI> などを参照