

# 今日からはじめる Python

平成 29 年 10 月 5 日

## 1 はじめに

タイトルにある Python は Guido van Rossum 氏によって作られた汎用プログラミング言語で、現在は Python ソフトウェア財団で管理、開発されています。数年前、すでに海外において多くのユーザーを獲得していた Python ですが、日本においては、和文の入門書や解説書の数も少なく、周囲で利用しているユーザーも少ない状態でした。近年、大量に取得されたデータ（ビッグデータ）の統計解析やそれを用いた機械学習といった技術が広まるとともに、日本においても、その解析手法の開発環境として Python のユーザーが増加してきたと思われます。

最近では、書店のパソコン関連の棚にもタイトルに Python が入った書籍を見かけるようになり、入門書も数多く出版されています。また、インターネット経由でアクセスできる情報も充実しており、多くのサンプルコードや日本語による解説を見ることができます。

プラズマ・核融合分野においても、Python を利用している人を見かけるようになりました。そろそろ使ってみようと考えている人もいることでしょう。はじめようと思うけれど、これまで馴染んできた Fortran や C 言語といったコンパイラとは異なる点に不安を覚えているかもしれません。本講座は、はじめてみようかと思っている人の背中を押して、実際にはじめてもらうことを目指しています。Python という言語を短く説明すると「誰でも利用できる書きやすく読みやすい汎用のプログラミング言語、だけど簡単」となります。

### 1.1 Python は誰でも利用できます

Python は無料で利用できるもので、どなたでもインストールしておくことができます。また、Windows, Linux, OS X といった主要なプラットフォームで利用できます。これは、共同研究をする上で非常に重要なことです。ある解析プログラムを渡して実行をお願いしても、その実行環境のライセンスを相手が持っていない場合がしばしばあるためです。Python で記述されたスクリプトは、主要なプラットフォーム上で、誰でも実行できます。

## 1.2 Python は書きやすい言語です

Python は、実行時の値によって変数の型が決められる（動的型付け）言語です。変数の型を指定する必要がないため、処理内容に集中して書くことができます。また、インタラクティブな実行環境を持っているため、処理の記述（プログラム作成）と適用結果の表示を繰り返して解析を進めていくことが容易です。

## 1.3 Python は読みやすい言語です

Python では、このブロック構造をインデントを用いて表現するため、読みやすい構造をもった書式が維持されます。ブロック構造とは、条件付きの処理、繰り返し処理や関数定義などの範囲のことです。このことは自由形式の言語に慣れている人には、書きづらいと感じるかもしれません。インデントのずれによって正常に動かなくなることを不安に感じる人も多いと思います。しかし、読み易さを損なわないためには、どの言語であっても一貫したインデントは必要です。また、インデントのずれについては、TAB キーによって空白が入らないようにエディタを設定するだけで防ぐことができます。そして、Python では技巧的な短い（パズル的で楽しい）記述よりも、分かりやすい記述が選ばれます。このような特徴から、インターネット上の解説やスクリプト、さらには付属のライブラリまで、読んで理解することが比較的容易です。

## 1.4 Python は汎用プログラミング言語です

Python は汎用のプログラミング言語なので、データ解析のみならず、テキストファイルの処理、グラフィカルユーザーインターフェース (GUI) の作成、ネットワークプログラミング、画像処理、ゲーム作成などパソコンで行われるほとんどのことに利用できます。また、Python プログラムは、手続き的に記述することも、オブジェクト指向や関数型と呼ばれるような技法で記述することも可能であるため、そのようなプログラミング技法を理解することの助けにもなるでしょう。そのため、プログラミング入門用の言語としてもおすすめてできます。

## 1.5 だけど簡単

Python が人気を得た理由のひとつとして、多くの有用なモジュールの存在があると思われます。モジュールは、パッケージと呼ばれる機能がつまったモジュールファイルの集まりとして提供され、それを自分の Python 環境にインストールすることによってプログラムを開発しなくても、必要な処理が

行えるようになるものです。Python に標準で備わっているモジュールでも多くのことができますが、外部から提供されるモジュールによってさらに簡単にできるようになります。たいいていの場合、行いたい処理に応じたモジュールを利用することになります。プラズマ・核融合分野においては、数値計算や解析のための優れたモジュール (numpy, scipy, matplotlib, pandas) や、それらのモジュールを便利に利用するためのインタラクティブな環境 (jupyter) が利用されています。

よいことばかりのように書いてしまいましたが、Python は万能ではありません。C 言語や Fortran といったコンパイラ言語を使っている方は、その実行時の速度に不満を持つかもしれません。大規模な数値計算やシミュレーションの主となる部分に Python を利用することは考えるべきではないでしょう。コードに依存しますが、動的な型付けを行うインタプリタである Python は大雑把に言って C 言語と比べて数十倍遅いでしょう。しかしながら、実験データの解析処理のうちインタラクティブに進められるような時間スケールの処理については、記述しやすい Python を用いる方が素早く結果を得られることがあるでしょう。Python から C 言語や Fortran のコードを利用することができますので、時間がかかる処理を C 言語や Fortran に置き換えることも可能です。Python は、それらの処理へのデータ入出力を担当することができます。そうすることで、C 言語や Fortran では記述するのが煩わしい入出力の部分を、ユーザーにとって便利なものにすることが容易になります。

Python を使うと便利な場面を箇条書きしてみます。

- 電卓代りにちょっとした計算をしたい。
- シェルスクリプトで解析処理を制御しているけど、より複雑な分岐制御をしたい。
- 様々なデータファイルから必要なデータを取り出して、解析したい。
- 解析処理プログラムを Fortran で書いたけど、入力データファイルの書式がバラバラだ。書式を合わせたい。
- ネットワーク経由で情報を取得したい。サーバー上の解析プログラムを自動で実行したい。

- 解析結果をグラフで描画しながら処理を進めたい。
- すばやく解析処理を書いて、手法の可能性を確認したい。

Pythonをはじめてみようという気分になってきたでしょうか。Pythonをはじめたばかりの方からは、どのようなときにどのようなモジュールを利用したらよいか判断できない、モジュールの使い方が分からないという声を聞きます。そこで本講座では、プラズマ・核融合分野でよく利用されている環境やモジュールを、みなさんが使い始めることができるように紹介していきたいと思います。

本講座は、計3回の講座で、今回は「Python スタートアップガイド」として、AnacondaというPythonの実行環境のインストール方法を、Windows、Linux、OS X を利用の方に向けて説明します。その後、Python スクリプトの作成方法、実行方法とJupyterというインタラクティブな環境を紹介し、それを用いてPythonの基本的な文法を説明します。次号では、「Pythonによる科学技術計算」ということで、核融合分野に限らず、科学技術計算に利用されるモジュールであるnumpy、scipy、およびその結果をグラフとして描画するのに便利なmatplotlibの使い方が説明されます。また、機械学習を実行するモジュールscikit-learnについても紹介されます。三回目では、「Pythonの活用事例」として、LHD実験やJT-60SA実験においてどのようにPythonが利用されているかを紹介していただきます。