

## 頻出の文法を厳選して紹介

### ● 自己学習の土台作り

入門者向けの情報がコンパクトにまとめられているPython公式サイトにチュートリアル<sup>(1)</sup>があります。コンパクトと言っても、項目だけ見てもかなりの量があります。

この中から全ての解説は行いませんが、Pythonの組み込み関数の一覧を表1(pp.26-27)に示します。組み込み関数とは、Built-in Functionsのことで、ライブラリをimportしなくても、さまざまな処理を行います。

本記事では、さらに必要最小限と思われる内容に絞り込み、以下の内容を扱います。

#### ▶ 標準ライブラリ

import文, int関数, float関数, bool関数, str関数, len関数, range関数, list関数, sort関数, sorted関数

#### ▶ 式

演算子の優先順位, トークン, デリミタ, 代数演算子, 比較演算子, ブール演算子, 文字列, リスト, タプル, 辞書

#### ▶ 複合文

if文, for文, try文, while文, break文, def文, return文

他の言語で学習経験がある方には少なすぎると感じられるかもしれません。さまざまな種類のプログラムを開発するときには、これだけでは不足することがあると思います。しかし、最も利用頻度の高い内容に厳選しておりますので、最初の一步を踏み出して、自己

注1: コードの意味, データの形式や構造などを含むソースコード中で利用されている変数や文が正しく動作するかを判断する基準。

学習できる知識の土台を作るには足りると考えました。

### ● 詳しく知りたくなったら…

Pythonは入門者にとって学習しやすいコンピュータ言語と言われますが、極めようとするPythonの細部の理解が必要になり、なかなか難しいです。

さらに詳しく知りたい方のために、Python公式サイトにあるドキュメントを紹介し、入門からステップアップする際に参考になります。

#### ▶ Python 言語リファレンス<sup>(2)</sup>

Pythonの仕組みや機能を理解する上で参考にできます。文句解釈, データ・モデル, 実行モデル, インポート・システム, 式, 単純文, 複合文などについて書かれています。

各項目の厳密な構文とセマンティクス<sup>注1</sup>について説明されています。これを読むとPythonでプログラムを実行したときに、どのような動作になるか理解できます。

#### ▶ Python 標準ライブラリ<sup>(3)</sup>

必須でない組み込みオブジェクト型や組み込み関数, 組み込みモジュールに関するセマンティクスなどの記載もあります。これを参照する場面は、Pythonの異なるバージョンで関数の動作の違いや引数の仕様を確認するときなどでしょう。Pythonのソースコードを読んで実装を確認するよりも簡単なはず。

#### ◆参考文献◆

- (1) Python公式サイトのチュートリアル。  
<https://docs.python.org/ja/3.7/tutorial/index.html#tutorial-index>
- (2) Python言語リファレンス。  
<https://docs.python.org/ja/3.7/reference/index.html>
- (3) Python標準ライブラリ。  
<https://docs.python.org/ja/3.7/library/index.html#library-index>

## 1-1 単純な変数

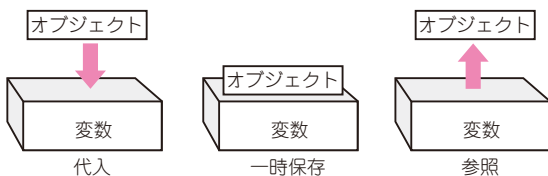


図1 変数はオブジェクトを入れておく入れ物

### ● 変数の利用目的

Pythonは事前に変数を宣言せず、変数で扱うオブジェクト型を指定しなくても利用できます。これはPythonの学習しやすい特徴の1つです。変数に値を

#### リスト1 Pythonの予約語や標準ライブラリに含まれる文字列

```
False, await, else, import, pass, None, break, except,
in, raise, True, class, finally, is, return, and,
continue, for, lambda, try, as, def, from, nonlocal,
while, assert, del, global, not, with, async, elif, if,
or, yieldなど
```

代入する際に、データの型は自動的に定義されます。しかし、オブジェクト型を意識せずに利用できるため関数の引数として利用するとき、変数に格納されているオブジェクトの型が適切でなくエラーになってしまうことがあります。変数の利用イメージを図1に示します。