Pythonで始める
プログラミング入門

工学博士　大和田勇人
博士(理学)　金盛　克俊　共著

コロナ社

ま　え　が　き

本書はプログラミング言語 Python の基礎と初歩的な例題に基づくプログラミング入門を目的とした本である。

Python は洗練された文法を持つスクリプト言語で，初心者にも学びやすく，これからプログラミングを始めようとする人にも最適な言語である。ほかの言語でプログラミング習得を断念した人にとっても，Python のシンプルな文法のサポートにより十分に再挑戦できると思う。ほかの言語と比較すると，Python はコード量が少なく，開発効率も高いと言われている。

また，Python ではライブラリと呼ばれる機能群が世界中で盛んに開発されており，そのほとんどが無料で公開されている。数値計算やデータ処理，機械学習やインタフェース機能を担うものなど，豊富なライブラリが存在しているのも Python を学習する大きなメリットである。

これまでは海外を中心に注目されてきた言語であったが，こうした点を考えると今後は国内の利用者も大いに増えることが予想される。実際，最新の制御プログラムが Python で記述され，技術者の求人にも Python のプログラミング能力が求められることも珍しくなくなってきた。

本書は第 I 部の基礎編と第 II 部の展開編により構成されている。第 I 部ではプログラミング言語 Python の基礎的な内容を，第 II 部では Python を用いた数値計算などの展開的な内容を収録している。

まず基礎編を始める前に，1 章で Python そのものの概要と Python の導入方法について解説する。

基礎編は 2 章から始まり，2 章ではプログラムの構成要素である文と式，またそれらを構成する変数や型，演算子といった基本的な概念について解説する。3 章では条件分岐文や繰り返し文のための制御構文について解説し，4 章では関

数の概念とその利用の仕方，定義の仕方を解説する。5 章ではオブジェクト指向プログラミングについてその概要と Python での実現方法を解説する。

展開編は 6 章から始まる。6 章では，7 章以降で必要となるファイルの操作（読み書き）について解説する。7 章では素数判定や素因数分解，最大公約数を求める問題といった基礎的な整数問題をプログラムで解く方法について解説する。8 章では方程式の解や定積分値を近似的に求めるプログラムについて解説する。9 章ではベクトルと行列を扱う方法とその演算について解説する。10 章ではプログラムで最適化問題を解くための最も基礎的な方法について解説する。

本書はプログラミング入門の教科書という位置づけで書かれたもので，半期の授業で学べるような内容になっている。例題はプログラミング入門としてふさわしいものに厳選したので，実際にソースコードをパソコンに打ち込んでみて，どのような実行結果になるかを確認しながら，ソースコードの解説を読んでみてほしい。また，演習問題は読者の理解を定着させる上で非常に重要なので，自力で問題に挑戦してほしい。なお，付録には演習問題の完全な解答を示したので，自分が作ったプログラムと照らし合わせて確認されると本物の実力がつくものと思われる。

最後に，本書を執筆するにあたり，東京理科大学大学院理工学研究科の大屋優氏には Python のインストール方法やプログラムの動作確認において，多大な協力をいただいた。ここに，感謝の意を表したい。

2015 年 8 月

大和田勇人・金盛克俊

目　　次

1. 導　　入

1.1 Python とプログラム …… 1
1.1.1 プログラムとは …… 1
1.1.2 Python と は …… 2
1.2 Python のインストール …… 3
1.3 プログラムの作成と実行 …… 4
1.4 ラ イ ブ ラ リ …… 6
1.5 開発環境について …… 6

第 I 部　基礎編

2. 変 数 と 計 算

2.1 文　と　式 …… 8
2.2 変数とオブジェクト …… 9
◆ 型 と 演 算 子 …… 11
節 末 問 題 …… 14
2.3 文字列の演算 …… 14
節 末 問 題 …… 17
2.4 リ　ス　ト …… 17
◆ 要素の追加・削除と検索 …… 19
2.5 辞　　書 …… 21
2.6 その他のデータ型 …… 24
2.6.1 タ　プ　ル …… 24

2.6.2 集 合 …… 25
Coffee Break（Python という名前） …… 27

3. 制 御 構 文

3.1 条件分岐 if 文 …… 29
◆ if 文の入れ子構造 …… 33
節 末 問 題 …… 35
3.2 繰り返し処理 while 文 …… 35
節 末 問 題 …… 38
3.3 繰り返し処理 for 文 …… 38
◆ リストの内包表記 …… 40
3.4 繰り返しの入れ子構造 …… 42
3.5 繰り返し処理の中の条件分岐 …… 43
3.6 繰り返し処理の中止と継続 …… 44
3.6.1 break 文 …… 44
3.6.2 continue 文 …… 45
章 末 問 題 …… 46

4. 関 数

4.1 組 込 み 関 数 …… 47
4.1.1 数値を扱う組込み関数 …… 48
4.1.2 イテラブルなオブジェクトを扱う組込み関数 …… 49
4.1.3 その他の組込み関数 …… 51
4.2 モジュールのインポート …… 52
4.3 関 数 の 定 義 …… 55
4.3.1 関 数 の 定 義 …… 55
4.3.2 ローカル変数とスコープ …… 58
4.3.3 参照渡しと値渡し …… 59

4.3.4　再帰呼び出し …… 60
Coffee Break（数当てゲームを作ってみよう） …… 63
章　末　問　題 …… 65

5. クラスとオブジェクト指向

5.1　メソッドの利用 …… 66
5.1.1　list メソッド …… 67
5.1.2　辞書型のメソッド …… 68
5.1.3　文字列のメソッド …… 70
5.2　クラスとインスタンス …… 71
章　末　問　題 …… 76

第 II 部　展開編

6. ファイル操作

6.1　ファイルの読込み …… 77
6.2　ファイルの書込み …… 79
6.3　CSV ファイルの操作 …… 81
6.3.1　CSV ファイルの読込み …… 81
6.3.2　CSV ファイルの書込み …… 83

7. 整数を扱う計算

7.1　約数と素数 …… 85
◆　エラトステネスのふるい …… 87
節　末　問　題 …… 89
Coffee Break（巨大な素数） …… 89
7.2　素因数分解 …… 90

節 末 問 題 …… 91
7.3 最 大 公 約 数 …… 91
◆ ユークリッドの互除法 …… 93
節 末 問 題 …… 94

8. 数 値 計 算

8.1 方 程 式 の 解 …… 95
8.1.1 二 分 法 …… 95
8.1.2 ニュートン法 …… 99
節 末 問 題 …… 103
8.2 微 分 …… 103
8.3 定 積 分 …… 107
8.3.1 区 分 求 積 法 …… 107
8.3.2 台形公式を用いた解法 …… 109
8.3.3 モンテカルロ法 …… 111
節 末 問 題 …… 114

9. ベクトルと行列

9.1 ベクトルの演算 …… 115
9.1.1 ベクトルの和 …… 115
9.1.2 ベクトルの内積 …… 116
節 末 問 題 …… 117
9.2 行 列 の 演 算 …… 117
9.2.1 行 列 の 和 …… 118
9.2.2 行 列 の 積 …… 119
章 末 問 題 …… 120

10. 最適化問題

10.1　ナップサック問題 …… 121
章末問題 …… 128

付　録 …… 129
A.1　Python のインストール …… 129
A.1.1　Python3.4.3 のインストール …… 129
A.1.2　Anaconda のインストール …… 132
A.2　ライブラリ …… 133
A.2.1　ライブラリのインストール …… 133
A.2.2　NumPy …… 134
A.2.3　SciPy …… 135
A.2.4　Matplotlib …… 135
節末問題解答例 …… 137
章末問題解答例 …… 144
索　引 …… 147

1 導　　入

Python というプログラミング言語を用いてプログラム開発を始めるにあたり，必要な諸概念と，コンピュータ環境の準備について解説する。読者のコンピュータ環境（OS など）はさまざまであると思われるため，環境の準備については，本章で一般的な概念と内容を解説することとし，巻末の付録に Windows 環境における詳細な準備方法を掲載している。

1.1　Python とプログラム

1.1.1　プログラムとは

Python は**プログラミング言語**（programming language）の一つである。プログラミング言語とは**プログラム**（program）を記述するための言語である。プログラムとはコンピュータが計算をするための手順を示す文書である。

プログラムは，コンピュータにどのような計算をさせるかを表す命令文書である。文書なので，なんらかの言語で記述する必要がある。日本語や英語などの人間が日常扱う言語では，あいまいな部分が多くコンピュータには理解が難しい。そのため，コンピュータが厳密にその意味を理解できるように定められた形式的な言語が使われている。この言語こそがプログラミング言語である。プログラミング言語によって記述されたプログラムそのものを**ソースコード**（source code）と呼ぶこともある。

プログラミング言語はコンパイラ言語とスクリプト言語に大別される。**コンパイラ言語**（compiler language）は人間によって記述されたプログラムを一度コンピュータが直接理解できる機械語という言語に翻訳してから実行される。

機械語はコンピュータ専用の言語なので，機械語に翻訳されたプログラムは人間に解読することはほとんど不可能だが，コンピュータが高速に効率よく実行できるようになっている。

一方，**スクリプト言語**（script language）では記述されたプログラムがほとんどそのまま読まれて実行される。スクリプト言語では記述されたソースコード（スクリプト（script）とも呼ぶ）はコンピュータが順番に読みながら実行するため，プログラムが効率よく実行されるかどうかはプログラムの記述に直接依存する。以上の理由から，スクリプト言語は人間の理解を損わないこととコンピュータによる実行効率との両立が重要となる。

1.1.2 Python とは

それでは，Python はどのようなプログラミング言語なのだろうか。Python はスクリプト言語である。一般にスクリプト言語はコンパイラ言語に比べて記述量が少なく読みやすいものが多いが，Python はその中でも特に記述量の少なさと洗練された文法を持つと言ってもよい。

さらに，Python には**ライブラリ**（library）と呼ばれる機能群が充実しているという大きな長所がある。ライブラリは世界中の技術者たちによって作成され，そのほとんどを無料でダウンロードして利用することができる。また多くのものはそのソースコードを改良し，新たに配布することも自由である。このような発想で公開されたソースコードあるいは公開することそのものを**オープンソース**（open source）と呼ぶことがある。

現在，公開されているライブラリの種類は多岐に渡り，本書で取り扱うような数値計算のための基本的なライブラリだけでなく，インタフェースやグラフィックを扱うライブラリや，最先端の機械学習の機能を持つライブラリ，ほかのプログラミング言語の機能を扱うためのライブラリなど，さまざまなものを利用することができる。そのため，Python はさまざまな目的に応じて効率よくプログラミングをすることのできるプログラミング言語ということができよう。

最後に，Python のバージョンについて触れておこう。Python には現在よく

利用されているものとして，Python2.X 系と Python3.X 系の 2 種類がある。本書で取り扱うのは Python3.X 系である。3.X 系はより新しいものだが，2.X 系とは文法が異なる点がいくつかあるため，いまでも 2.X 系を使う人がいる。3.X 系がリリースされた当初は多くのライブラリがまだ 3.X 系に対応しておらず，3.X 系に移行しづらい状況があった。しかし今日ではライブラリの対応も進み，今後は Python3.X 系が主流になることが予想される。

1.2 Python のインストール

Python を自分のコンピュータ環境で利用できるようにするには，Python インタプリタをインストールしなくてはならない。**インタプリタ**（interpreter）はスクリプトを読みながら実行するプログラムのことである。この意味で Python は**インタプリタ言語**（interpreter language）とも言われる。

Python のインストール方法には，利用する OS や環境に応じてさまざまな選択肢がある。

Python インタプリタ本体だけをインストールするには，まずは Python Software Foundation のサイト（http://www.python.org/[†]）にアクセスして，自分の環境に合ったインストーラをダウンロードして，指示に従いインストールすればよい。

また，関連する拡張機能（ライブラリ）やソフト（開発環境）などをまとめてインストールできる**ディストリビューション**（distribution）と呼ばれるインストーラも便利である。以下に主なディストリビューションについて説明する。

- **Anaconda**

  数値計算や科学計算，特に大きなデータを扱う解析のために作成された無料のディストリビューションである。Windows や Linux，MacOS などさまざまな OS で利用可能なマルチプラットフォームで，数値計算に便利なライブラリや開発環境がまとめてインストール可能である。Anaconda

[†] 本書での URL は 2015 年 8 月現在で確認済。

はつぎの Web サイトからダウンロードできる。

http://continuum.io/

- **WinPython**

  Windows で利用可能な数値計算のための無料のディストリビューションで，Anaconda と同様にいくつかのライブラリと開発環境がまとめてインストールできる。WinPython はつぎの Web サイトからダウンロードできる。

  http://winpython.sourceforge.net/

本書で扱う内容はこれらのどのインストール方法を用いても問題はないので，目的に応じて適切なインストール方法を選ばれたい。

巻末の付録に Windows 環境における詳細なインストールの方法を解説してあるので，Windows 環境での開発を試みる方は参照していただきたい。

1.3　プログラムの作成と実行

Python の実行方法は 2 通りある。一つはコンソールに直接プログラムを 1 行ずつ打ち込んで順次実行していく対話型の実行方法で，もう一つはプログラムファイルを作成して Python インタプリタに一気に実行させる実行方法である。

前者のように対話型で実行したい場合は，Python インタプリタを引数（ひきすう）なしで起動すると，1 行ずつのプログラムを受け付ける状態で Python インタプリタが起動されるので，1 行ずつプログラムを打ち込んでいけばよい。後者の場合は，Python インタプリタの起動時に引数として実行したいプログラムファイル名を与えればよい。本書では特に断りがない限り，プログラムファイルを作成して実行する方式を前提とする。

また，ここで紹介する方法は特別な開発環境を用いない方法であるので，開発環境を利用してプログラムの作成と実行を行う場合には，別途開発環境の利用方法を調べていただきたい。

Python プログラムを作成するには，なんでもよいのでテキストエディタな

どでプログラムソースを打ち込んでファイルに保存するだけでよい。例えば，**プログラム 1-1** のようなプログラムをテキストエディタで打ち込んでみてほしい。たった 1 行だけのソースコードだが，これも立派なプログラムの一つである。このプログラムは，「Hello World!」という文字列を表示しなさいという意味を表している。打ち込んだら，これにファイル名をつけて保存してほしい。ここでは HelloWorld.py というファイル名で保存することにしよう。ファイル名の末尾の「.py」という拡張子は，そのファイルが Python のプログラムファイルであることをわかりやすくするための識別子なので，実際にはどんな拡張子でも実行することができるが，本書では「.py」で統一する。Python3 では UTF-8 という文字コードを用いて日本語などのマルチバイト文字を扱うことになっているため，文字コードを UTF-8 で指定保存できるエディタを利用するのが望ましい。ファイルが保存できたらプログラム作成の準備は完了である。

プログラム 1-1

```
1  print('Hello World!')
```

ファイルを保存したら，インストールした Python インタプリタを起動してこのプログラムを実行することができる。コマンドラインにつぎのようなコマンドを入力することで，指定した Python スクリプトファイルを実行することができる。

```
python HelloWorld.py
```

実行が成功すると**実行結果 1-1** のような実行結果が表示される。

実行結果 1-1

```
Hello World!
```

このプログラムの`print('Hello World!')`という 1 行は，「`Hello World!`」という文字列を表示しなさいという意味である。「`print`」は直後に続くカッコ内のものを表示しなさいという**関数**であるため，カッコ内の文字列を変更すると，実行結果も変わることが確認できる。カッコ内のシングルクォーテーショ

索　　引

【あ】

値渡し　59
アルゴリズム　91

【い】

イテラブル　38
入れ子構造　33, 42
インスタンス　71
インスタンス変数　66
インタプリタ　3
インタプリタ言語　3

【え】

エラトステネスのふるい　87
演算子　13

【お】

オープンソース　2, 6
オブジェクト　10
オブジェクト指向　66
オブジェクト指向
　プログラミング　66

【か】

改行コード　80
返り値　47
型　11
仮引数　56
関　数　47

【き】

疑似乱数　54
基本型　11

【く】

区分求積法　107
組合せ最適化問題　122
組込み関数　47
クラス　11, 71
クラスインスタンス　71
クラス変数　74
繰り返し処理　35

【こ】

コメント文　9
コンストラクタ　72
コンパイラ言語　1

【さ】

再帰関数　60
再帰呼び出し　60
最適化問題　121
参照渡し　59

【し】

式　8
辞書型　21
実引数　48
集　合　25
出　力　9
条件分岐文　29

【す】

数値解析　95
数値計算　95
スクリプト言語　2
スコープ　58

【せ】

制御構文　28

【そ】

素因数分解　90
ソースコード　1
属　性　66

【た】

代　入　10
代入演算子　10
代入文　10
タプル　24
単項演算子　13

【て】

定　数　53
ディストリビューション　3

【な】

内包表記　18, 40
ナップサック問題　122
名前空間　58

【に】

二項演算子　13
二次元リスト　118
二分法　95, 96
ニュートン法　95, 99
ニュートン・ラフソン法　99

【ね】

ネスト構造　33

【は】

パラメータ　47

【ひ】

比較演算子　11, 29
引　数　47
標準出力　8
標準入力　51

【ふ】

プログラミング言語 1
プログラム 1
文 8

【へ】

変 数 9

【む】

無限ループ 36

【め】

メソッド 66

【も】

モジュール 52
戻り値 47
モンテカルロ法 111

【ゆ】

ユークリッドの互除法 92

【よ】

要 素 17
予約語 29

【ら】

ライブラリ 2, 6

【り】

リスト 17

【ろ】

ローカル変数 58

論理演算子 30

【A】

`abs` 49
algorithm 91
Anaconda 3, 132
`append` 67
argument 48
assignment 10
assignment operator 10
assignment statement 10

【B】

binary operator 13
bisection method 95
`bool` 型 11
`break` 文 44
built-in function 47

【C】

call by reference 59
call by value 59
class 11, 71
`class` 文 71
class variable 74
`close` 78, 80
combinatorial optimization 122
comma-separated values 81
comment statement 9
comparison operator 11, 29
compiler language 1
`complex` 型 11
conditional statement 29
constructor 72
`continue` 文 45
control flow 28
`cos` 53
`count` 67, 70
CSV 形式 77, 81

【D】

`def` 文 55
dict 21
distribution 3

【E】

`e` 54
Eclipse 7
element 17
Euclidean algorithm 93
`exp` 53
expression 8

【F】

`factorial` 53
`False` 11
`find` 70
`float` 型 11
`for` 文 38
formal parameter 56
function 47

【I】

`id` 51
`if` 文 29
`import` 文 52
`in` 演算子 20
`index` 67
infinite loop 36
`input` 51
instance 71
instance variable 66
`int` 51
`int` 型 11
interpreter 3
interpreter language 3
`items` 69
iterable 38

【K】

keys 69
knapsack problem 122

【L】

len 50
library 2
line feed code 80
list 17
list comprehension 18, 40
local variable 58
log 53
logical operator 30
log10 53
log2 53
loops 35

【M】

math モジュール 53
max 49
method 66
min 49
module 52
monadic operator 13
Monte Carlo method 111

【N】

name space 58
nested structure 33
Newton's method 95, 99
Newton-Raphson method 99
None 12
numerical analysis 95
numerical calculation 95

【O】

object 10
object oriented programming 66
open 77
open source 2
operator 13
optimization problem 121
output 9

【P】

parameter 47, 56
pi 53
pop 67, 69
prime factorization 90
print 5
program 1
programming language 1
property 66
pseudorandom numbers 54
PyScripter 7

【Q】

quadrature by parts 107

【R】

randint 54
random 54
random モジュール 54
range 39, 50
read 78
readline 78
readlines 78
recursive call 60
recursive function 60
remove 67
reserved word 29
return 文 56
return value 47
round 49
rstrip 82

【S】

scope 58
script language 2
self 72
set 25
sieve of Eratosthenes 87
sin 53
sort 68
sorted 50
source code 1
Spider 7
split 70
standard input 51
standard output 8
statement 8
str 型 11
sum 50

【T】

tan 53
TextIOWrapper 77
True 11
tuple 24
two dimensional list 118
type 12, 51
type 11

【U】

uniform 54
uniform 関数 112
update 69

【V】

values 69
variable 9

【W】

while 文 35
WinPython 4
write 80
writelines 80

【記号】

__init__ 72

—— 著 者 略 歴 ——

大和田　勇人（おおわだ　はやと）

1983 年　東京理科大学理工学部経営工学科卒業
1988 年　東京理科大学大学院理工学研究科博士課程修了（経営工学専攻）工学博士
1988 年　東京理科大学助手
1999 年　東京理科大学専任講師
2001 年　東京理科大学助教授
2005 年　東京理科大学教授
現在に至る

金盛　克俊（かなもり　かつとし）

2004 年　東京理科大学理工学部情報科学科卒業
2009 年　東京理科大学大学院理工学研究科博士課程修了（情報科学専攻）博士（理学）
2009 年　フリーランスでアプリ開発業務に従事
2012 年　東京理科大学助教
現在に至る

Python で始めるプログラミング入門

Introduction to Python Programming

2015 年 10 月 13 日　初版第 1 刷発行　★

検印省略

著　者　大和田　勇人
　　金盛　克俊
発行者　株式会社　コロナ社
　　代表者　牛来真也
印刷所　三美印刷株式会社

112–0011　東京都文京区千石 4–46–10

発行所　株式会社　コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140–8–14844・電話(03)3941–3131(代)

ホームページ　http://www.coronasha.co.jp

ISBN 978–4–339–02498–2　（中原）　（製本：SBC）

Printed in Japan