Python版

つくって学ぶ Processingプログラミング入門

博士(工学) 長名 優子
博士(工学) 石畑 宏明 共著
博士(工学) 菊池 眞之

コロナ社

ま　え　が　き

本書は，プログラミング初学者を対象として，「物事を論理的に考えて課題を解決する練習」をプログラミングの学習を通して実践するための書籍「つくって学ぶ Processing プログラミング入門」の姉妹版です。Java 言語をベースとした Processing に代わって Python 言語をベースとした Processing を使用するところが主な違いです。Python は，データ処理などで広く使われるようになったプログラミング言語で，豊富なライブラリを持ち，特にデータサイエンスや人工知能の分野では主流のプログラミング言語になっています。今後，これらの分野へ進む学習者に向けて，プログラミング技術がスムーズに繋がるよう考慮しています。

Processing は，グラフィックス機能・マウスやキーボード入力インターフェースに優れ，プログラムを実行した結果をビジュアルに確認することが簡単にできるプログラミング言語です。本書では，Python モードの Processing を使用します†。プログラムの内容は，姉妹版と同等で，理系の大学初年度程度の知識を持つ，初めてプログラミングに取り組む学生を対象としていますが，高校生でもある程度理解できる内容になっています。「このように記述すれば，このような結果が得られる」という論理的な筋道を簡単に表現し，その動作の確認が行えるので，論理的思考力をトレーニングする教材として最適と考えています。

学習者には，プログラムに興味を持ちプログラミングの面白さを知ってもらいたいと考えています。そこで，プログラミングの課題を，学生が興味が持てる内容であり，かつ，これまで中学・高校を通して学んできた知識を活用する機会を与えるようなものにしました。例えば三角関数などこれまでなんの役に立つのだと思いながら勉強してきた数学の知識が，図形を描画するという課題に直面したときに，実際に有用なことをプログラムの作成を通して実感できます。また，英語で表示されるエラーメッセージも，慣れればどうということもないことに気がつきます。思い通り動かないプログラムと悪戦苦闘しながらも，完成したときの達成感は大きいものです。

1 章から 7 章までで，基本的なプログラミングの技術の最低限の要素を学びます。初学者を対象としていますので，はじめはステップバイステップで丁寧にプログラムの書き方を説明していきます。なお，新しい要素や概念は，必要に応じてその都度説明します。2 章では，

† Python にはバージョン 2 と 3 があり，Processing の Python はバージョン 2 である。本書で扱う範囲では，ほとんどバージョンによる違いの影響はないが，影響のあるところについてはその都度注をつけてある。

Processing を使用して簡単な図形を描くプログラムを作成します。3 章では，変数と繰り返し文の使用方法，4 章では条件文の書き方を学びます。5 章では，マウス・キーボードからの入力によってプログラムの振る舞いを変える方法を学びます。これによって，ゲームなどの対話的な処理ができるようになります。6 章では，関数の作成方法と使い方，7 章では，リストと呼ばれるデータの集合の作り方と使い方を学びます。ここまでで，多くのプログラミング言語で共通に現れる，プログラミングの基本的な技術を学びます。例題に沿って，実際にプログラムを入力して動作確認をしていってください。

8 章以降は，それぞれがプロジェクトになっています。プロジェクトで作成するプログラムは，行数は少ないけれどもそれなりの複雑さを持ったプログラムです。プログラムは穴埋め形式になっており，処理の流れを考えながらプログラムを入力するという作業で進めます。基本的な機能を実装・動作させた後は，各自自分のアイディアを追加機能として組み込んでください。8 章では時計を，9 章ではストップウォッチを作成します。10 章では，音楽ファイルを読み込んでそれを映像として表現するサウンドビジュアライザを作成します。11 章では，アクションゲームの作成に挑戦します。最後の 12 章では，迷路ゲームを作成します。乱数を使用して迷路を生成し，その上でゲームを行うプログラムを作ります。さらに，コンピュータにその迷路を解かせるプログラムを作成します。最後には，それを 3 次元的な表示が行えるように拡張します。学習者の皆さんには，それぞれのプログラミングの課題の実現を通して，論理的に考える習慣をつけ，タイピングに慣れ，英語や数学の知識を活用できるようになることを期待します。なお，本書の執筆分担は 1〜3 章（石畑），4，5 章（菊池），6〜9 章（長名），10 章（長名，石畑），11 章（菊池，長名），12 章（石畑，長名）となっています。また，図面がカラーのものについては，コロナ社の Web ページ（p.20 参照）からダウンロードできます。

本書は，Windows10 をプラットフォームとして使用し，この上で動作確認を行っています。使用している Procsessing のバージョンは 3.5.3 です。Processing や Windows は，今後バージョンアップの可能性があります。内容について万全を期して制作しましたが，万一誤りや不備がありましたら出版元までご連絡ください。なお，本書の内容の運用による結果の影響について，一切の責任を負いかねます。最後に，執筆にあたり種々のサポートをいただいたコロナ社に深謝いたします。

2019 年 12 月

長名 優子，石畑 宏明，菊池 眞之

本書を執筆するに当り，下記の書籍を参考にいたしました。
長名，石畑，菊池，伊藤：つくって学ぶ Processing プログラミング入門，コロナ社（2017）

目　　次

1. Processing を始めるための準備

1.1 Processing のインストール …… 1
1.2 動 作 確 認 …… 2

2. 初めての Processing

2.1 画 像 デ ー タ …… 5
2.2 プログラミング …… 6
2.2.1 最初のプログラム …… 6
2.2.2 図形の塗りつぶし …… 7
2.2.3 線の色の指定 …… 9
2.2.4 背　　景 …… 9
章 末 問 題 …… 10

3. 変数と繰り返し文

3.1 変　　数 …… 11
3.1.1 変 数 の 宣 言 …… 11
3.1.2 変数を使用した演算 …… 12
3.2 処理を繰り返す …… 14
3.2.1 繰り返し文（for 文） …… 14
3.2.2 for 文のネスト（入れ子） …… 17
3.3 シ ス テ ム 変 数 …… 19
章 末 問 題 …… 20

4. 条件分岐とマウスカーソルの座標に応じた処理

4.1 条　 件　 文 …… 21
4.1.1 条　 件　 式 …… 21

4.1.2 if 文，if～else 文，if～elif～else 文 …… 22
4.2 スタティックモードとアクティブモード …… 25
4.3 マウスカーソルの座標に応じた処理 …… 28
章 末 問 題 …… 30

5. マウス・キーボードによる操作

5.1 マウスの動きに反応するプログラム …… 31
5.1.1 変数 mousePressed …… 31
5.1.2 システム変数 mouseButton …… 32
5.1.3 関数 mousePressed() …… 33
5.2 キーボード入力に反応するプログラム …… 35
5.2.1 システム変数 keyPressed と key …… 35
5.2.2 関数 keyPressed() …… 37
5.2.3 押されたキーの判定 …… 38
章 末 問 題 …… 40

6. 関 数

6.1 関 数 と は …… 41
6.2 引数も戻り値もない関数 …… 41
6.3 引数のある関数 …… 45
6.4 戻り値のある関数 …… 46
章 末 問 題 …… 48

7. リ ス ト

7.1 リ ス ト …… 49
7.2 多次元（2 次元）リスト …… 55
7.3 リストを利用したプログラムの例 …… 57
章 末 問 題 …… 61

8. つくってみよう：時計

8.1 時刻情報の取得 …… 67

8.2　アナログ時計 …… 68
8.2.1　ウィンドウの作成と時計の外側の円の描画 …… 68
8.2.2　目盛の描画 …… 69
8.2.3　秒針の描画 …… 70
8.2.4　分針の描画 …… 71
8.2.5　時針の描画 …… 72
8.3　ディジタル時計 …… 73
8.3.1　六角形を描画する関数 draw_hex() の作成 …… 73
8.3.2　七つの六角形で数字を描画する関数 seven_segment() の作成（1） …… 74
8.3.3　七つの六角形で数字を描画する関数 seven_segment() の作成（2） …… 76
8.3.4　時刻の描画 …… 77

9.　つくってみよう：ストップウォッチ

9.1　実行開始からの経過時間の取得 …… 78
9.2　一時停止が可能なストップウォッチ …… 83
9.2.1　ストップボタンでの一時停止 …… 83
9.2.2　リセットボタンの追加 …… 84
9.3　ラップタイムの取得が可能なストップウォッチ …… 86

10.　つくってみよう：サウンドビジュアライザ

10.1　音 …… 91
10.1.1　音とは …… 91
10.1.2　PCM …… 92
10.2　minim ライブラリ …… 93
10.3　音声信号の波形の表示（1） …… 93
10.4　音声信号の波形の表示（2） …… 98
10.5　音声信号の線の長さによる表現 …… 101
10.6　音声信号の色による表現 …… 103

11.　つくってみよう：アクションゲーム

11.1　ゲームの作成手順 …… 106

11.2 自キャラの基本動作の処理 …… 107
11.3 床に関する処理 …… 114
11.4 敵キャラに関する処理 …… 125

12. つくってみよう：迷路

12.1 迷路の設定・表示 …… 131
12.1.1 盤 の 作 成 …… 131
12.1.2 盤 の 初 期 化 …… 132
12.1.3 盤 の 表 示 …… 133
12.2 簡単な迷路の生成 …… 135
12.3 キーボード入力でコマを操作するゲームの実現 …… 137
12.3.1 グローバル変数の宣言とゲームの初期化 …… 137
12.3.2 キーボード入力に対する処理 …… 138
12.3.3 コマやプレイ情報の表示およびゲームの終了 …… 140
12.4 乱数を用いた迷路の自動生成 …… 142
12.5 マウスでコマを操作するゲームの実現 …… 145
12.6 迷路の自動的な探索 …… 150
12.7 迷 路 の 3D 化 …… 154
12.7.1 迷路の3D表示 …… 154
12.7.2 キーボードによる操作 …… 160

付録 本書で使用している関数・システム変数一覧 …… 163
索 引 …… 166

1 Processingを始めるための準備

Processingは，MIT Media Lab. のBen FryとCasey Reasが2001年から開発しているソフトウェアであり，スケッチブックに絵を描くように容易にプログラムをつくることのできる環境が提供されている。誰でも公式Webサイトから無料でダウンロードでき，WindowsやMac，Linuxなどの主要なOSの下で動かすことができる。現在では，ソフトウェア教育からビジュアルアートまで幅広く利用されている。

ここでは，Processingプログラムの開発・実行環境をダウンロードし，使用できるようにする。Processingは，Java言語をベースとして初心者にも使いやすく機能拡張したものである。本書で使用するProcessing.py（ProcessingのPython Mode）は，このベース言語をJava言語からPythonに変えたもので，Python言語の文法で記述したプログラムを実行する。

1.1 Processingのインストール

次の手順でProcessingをインストールする。ここでは，Windows 10での方法を中心に説明するが，他のOSでもほぼ同様に行うことができる。

1. Web ブラウザで Processing の Download ページ（https://processing.org/download/?processing/）を開く。
2. 使用しているPCとそのOSに対応したものをダウンロードする。64 bit版のWindowsなら，“Windows 64bit” のリンクをクリックする。
3. ダウンロードした圧縮ファイルを展開する。Windowsの場合は，“ダウンロード” フォルダにあるprocessing-3.5.3-windows64.zipを右クリックして，“すべて展開” を選択する。なお，今後Processingのバージョンが変わる可能性がある。その場合は，これ以後，そのときのバージョンに合わせてバージョン番号の部分（3.5.3）を読み替える。
4. 展開済みのフォルダprocessing-3.5.3を適当な場所（例えば自分のPCの “ドキュメント” フォルダ）に移動する。
5. デスクトップにProcessingのショートカットを作成しておくと便利である。Windowsなら，移動したフォルダ “Processing-3.5.3” をダブルクリックして開き，中にあるファイル “Processing” を右クリックして，“送る” ⇒ “デスクトップ（ショートカットを作成）” を選ぶ。

1.2 動 作 確 認

1. デスクトップ上のショートカット “Processing” をダブルクリックして起動する。起動中は，**図 1.1** ような画面が現れる。起動できると，**図 1.2** のようなウィンドウが開く。

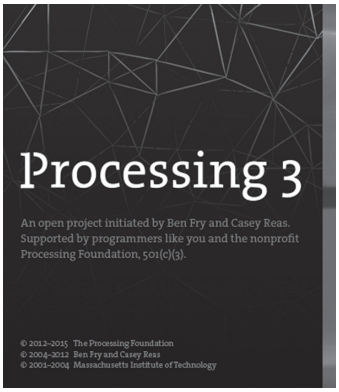

図 1.1　起動中

図 1.2　起動完了

2. インターネットに接続した状態で Python Mode をインストールする。“ツール”⇒“ツールを追加” をクリックして，Contribution Manager を開く。新しく開いたウインドウで，Modes タブを選択し，中程にある “Python Mode for Processing 3 | Write ....” を選択する（**図 1.3**）。Install ボタンを押してインストールする。Update マークがある場合は下の方にある Update ボタンを押す。自動的にインストールまで行われる。インストール終了後，このウィンドウを閉じる。

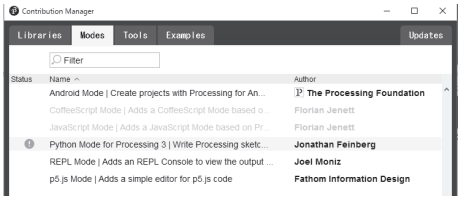

図 1.3　Contribution Manager

3. 起動画面に戻り，右上にある Java ボタンを押して，Python に切り替える。新規に起動画面が現れる。右上部分に Python の表示があることを確認しよう。この状態で，Python 言語で Processing のプログラムを記述・実行ができる（**図 1.4**）。なお，一度 Python Mode にすると，それ以後 Processing を起動すると自動的に Python Mode で起動される。

図 1.4 Python Mode での起動画面

4. プログラム編集用の領域に “rect(10, 10, 60, 60)” と入力して，簡単なプログラム（一辺が 60 の正方形を一つ描くプログラム）を書いてみる。Processing では，プログラムのことをスケッチと呼ぶ（**図 1.5**）。

注意 1：必ず「半角」で英数字を書く。全角のカッコや空白が混じると動かない。
注意 2：大文字と小文字は区別される。
注意 3：プログラム中の文字の色は，プログラムで使用するキーワードなどその種類に対応して勝手につく。プログラムの文法上明らかにおかしい部分は赤い波下線で示される。
注意 4：プログラムのエラーは，コンソールおよびメッセージエリアに表示される。

図 1.5 スケッチ

5. Run ボタン を押して実行する。メニューの “スケッチ” ⇒ “実行”，もしくはキーボードで Ctrl+R でも実行できる。
6. **図 1.6** のように新たに小さなウィンドウができ，その中に正方形が描かれていることを確認する。

図 1.6　実行結果

7. プログラム編集ウィンドウで Stop ボタン を押して終了する。Esc キー，もしくはウィンドウ右上の×ボタンでも終了できる。

メニューの “ファイル” ⇒ “新規” で，新たにプログラムを作成できる。新たに作成したスケッチには，sketch_YYMMDDx（YYMMDD は年月日，x は a, b, c · · ·）のように自動的に名前がつけられる。自分で名前をつけて保存したい場合には，“ファイル” ⇒ “保存” で保存場所を自分で指定して保存できる。すでに保存されたファイルを再度開くには，“ファイル” ⇒ “開く” でファイルを選択する。

ウィンドウの一番下には，コンソールタブがある。コンソールタブを選択すると，プログラムの実行に伴うメッセージがコンソール領域に表示される。Processing のプログラムであるスケッチはどこにおいてもよいが，デフォルトでは，**図 1.7** のようにユーザの書いたスケッチは，“ドキュメント” の下に置かれた “Processing” フォルダの下に配置されている。

Processing	スケッチを置くフォルダ
examples	
libraries	
modes	
sketch_YYMMDDx	スケッチのフォルダ
sketch_YYMMDDx	スケッチのプログラム本体
mysketch	スケッチのフォルダ
mysketch	自分で’mysketch’ と名前をつけて保存した例
tools	
Processing-3.5.3	Processing 本体のフォルダ
Processing	Processing の実行ファイル
...	Processing に必要なその他のフォルダやファイル

図 1.7　Processing のフォルダ構成

2 初めての Processing

ここでは，矩形や円などの簡単な図形を描く Processing のプログラムの作成を通して，プログラミングの基本を学ぶ。背景をカラーにしたり，図形を指定した色で塗りつぶす方法などを習得する。

2.1 画像データ

画像はピクセル（画素）と呼ばれる点の集まりで表現され，各ピクセルは，明るさや色を示す値を持つ。各ピクセルの位置は座標で指定する。Processing における座標系は，**図 2.1** に示すように，左上を原点とし，右方向が x 軸，下方向が y 軸となる。図では，30 × 20 の大きさの画像（キャンバス）で（17, 2），（7, 7），（22, 16）のピクセルに色が付いている状態を示している。

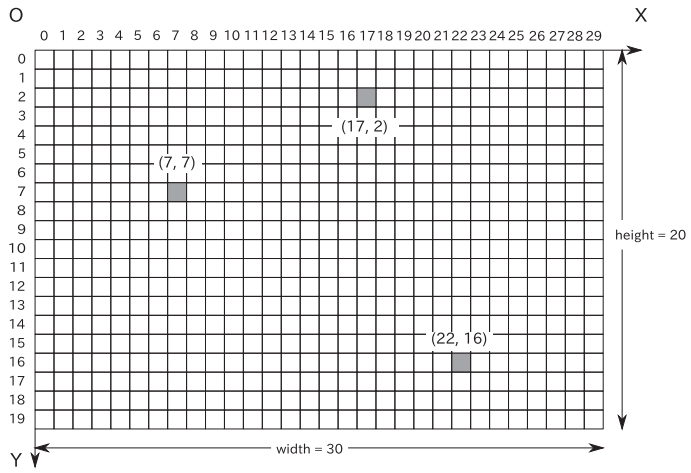

図 2.1 座標系

各ピクセルは，明るさや色を示す値を持つ。グレースケール（白黒）画像の場合は，0〜255 の範囲の値で明るさを表す。0 に近いほど暗く（黒く），255 に近いほど明るい（白い）色となる。カラー画像の場合は，赤（R），緑（G），青（B）の 3 原色のそれぞれについて 0〜255 の範囲の値で明るさを表す。**表 2.1** に代表的な色とそれに対応する R，G，B の値を示す。

索　　引

【あ】
アクティブモード 26

【い】
イテラブル 55
入れ子 17
インデント 15

【え】
遠近法 155

【お】
音　波 91

【か】
型 11
関　数 6, 41

【き】
キャンバス 5

【く】
繰り返し文 14
グレースケール 5
グローバル変数 27

【こ】
コンソール 3

【さ】
彩　度 104
サンプリング 92
サンプリング周波数 92

【し】
色　相 104
システム変数 19, 28
視　点 155
視野角 156

【す】
スケッチ 3
スタティックモード 25
ステレオ 96

【そ】
疎密波 91

【た】
代入演算子 13
多次元リスト 55

【ね】
ネスト 17

【ひ】
引　数 6, 41
ピクセル 5
標本化 92

【ふ】
フレーム 27
フレームレート 28

【へ】
変　数 11

【め】
明　度 104
メッセージエリア 3

【も】
文字列リテラル 33
戻り値 41
モノラル 95

【よ】
予約語 11

【ら】
ライブラリ 93
乱　数 44

【り】
リスト 49
リスト内包表記 55
量子化 92
量子化ビット数 92

【ろ】
ローカル変数 27

【英字】
for 文 14
HSB 104
if 文 22
minim 93
PCM 92
return 46

【数字】
1 次元リスト 55
2 次元リスト 55

—— 著 者 略 歴 ——

長名　優子（おさな　ゆうこ）
1996年　慶應義塾大学理工学部電気工学科卒業
1998年　慶應義塾大学大学院理工学研究科修士課程修了（電気工学専攻）
1998年　日本学術振興会特別研究員（博士課程在学中）
2001年　慶應義塾大学大学院理工学研究科博士課程修了（電気工学専攻），博士（工学）
2001年　東京工科大学助手
2003年　東京工科大学講師
2012年　東京工科大学准教授
現在に至る

菊池　眞之（きくち　まさゆき）
1994年　早稲田大学理工学部電気工学科卒業
1996年　大阪大学大学院基礎工学研究科博士前期課程修了（物理系専攻）
1996年　大阪大学大学院基礎工学研究科博士後期課程中退（物理系専攻）
1996年　大阪大学助手
1997年　大阪大学大学院助手
1999年　博士（工学）（大阪大学）
1999年　筑波大学助手
2003年　東京工科大学講師
現在に至る

石畑　宏明（いしはた　ひろあき）
1980年　早稲田大学理工学部電子通信学科卒業
1980年　株式会社富士通研究所入社
1996年　博士（工学）（早稲田大学）
2007年　東京工科大学教授
現在に至る

Python 版 つくって学ぶ Processing プログラミング入門

Python Version Introduction to Processing Programming —Learning by Making—

2020 年 1 月 10 日　初版第 1 刷発行　★

検印省略

著　者　長名　優子／石畑　宏明／菊池　眞之
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
製本所　有限会社　愛千製本所

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　https://www.coronasha.co.jp

ISBN 978–4–339–02901–7　C3055　Printed in Japan　（大井）