

目次

まえがき	3
はじめに	4
謝辞	6

第1章 学習継続力の壁 学習が苦しくならないために 12

1.1 初学者が抱きがちな4つの「間違った考え方」とは	12
1.2 本に書かれていることを全部暗記しなくて良い理由	13
1.3 「間違っはいけない」と思わなくて良い理由	13
1.4 他人と学習スピードを比べなくて良い理由	14
1.5 高過ぎる目標を設定しなくて良い理由	14
1.6 プログラミング学習は「学校の勉強」とは違う	15
1.7 プログラミングという登山の第一歩を踏み出そう!	15

第2章 PCスキルの壁 プログラミングに必要なPCの知識を身につけよう 16

2.1 本書の読者に期待するPCの基本スキル	16
2.2 ファイルについて理解を深めよう	16
練習問題	25
2.3 ターミナルとコマンド	25
column 練習のため、ターミナルの横にエクスプローラー (Finder) を表示しよう	35
column ターミナルでは必ず「コマンド」「対象のパス」の順で指定する	38
2.4 ファイルとディレクトリ構造、便利なコマンドをおさらいしよう	42
練習問題	43

第3章 環境構築の壁 プログラミング環境を正しく作ろう 44

3.1 プログラミング環境が学習のつまずきになる理由を知ろう	44
3.2 PC環境をシンプルにすることでトラブルを防ごう	45

3.3	必要最小限のソフトウェアをインストールしよう	46
3.4	Pythonをインストールしよう	47
	column プログラミング言語の種類と入門にPythonを選ぶ理由	47
	column 画面の表示が本と違うときはどうしたらいい?	48
	column Pythonのインストールをやりなおす	53
	発展 Install Certificates.commandも実行しておくが良い	60
	発展 macOSではpythonではなくpython3を使う	62
	column macOS使用者の心得	63
3.5	Visual Studio Codeをインストールしよう	64
3.6	Visual Studio Codeの基本的な使い方	73
	column ショートカットキーを覚えよう	76
	発展 Visual Studio Codeでも「戻せる」	79
	column メニューが英語でも怖くない	82
	column オープンソースソフトウェアの利用と貢献とは	84
3.7	プログラミング環境が正しく作れていることを確認しよう	84
	発展 プログラミング言語のバージョンを意識しよう	85

第4章 変数の壁 文法を学んで変数を正しく理解しよう 87

4.1	なぜ「変数」が正しく理解できないのか	87
4.2	文と式を理解しよう	87
	発展 英語の用語を覚える価値	89
	column スペースは見やすさのため	90
	発展 「1」と書いただけでも式	91
4.3	値と型を理解しよう	92
	発展 整数値の範囲	93
	発展 浮動小数点数は「有効数字」	94
	発展 できるだけダブルクォーテーションに統一するわけ	95
4.4	さまざまな演算を学ぼう	96
	発展 2項演算以外の演算もある	96
4.5	変数を理解しよう	101
	column 変数は「箱」ではない	102
	発展 「変数」「代入」という言葉にとらわれてはいけなわけ	104

▶ 発展 ▲	変数名はできれば英語で書こう	105
▶ 発展 ▲	Pythonではもっと簡単な書き方がある	109
4.6	関数を理解しよう	109
▶ 発展 ▲	関数は自分で作ることもできる	114
4.7	「変数」を構成する文法をおさらいしよう	114
▶ 発展 ▲	本章の説明が当てはまらないプログラミング言語も存在する	115
練習問題		115
▶ 発展 ▲	type関数を使えばさまざまな型を見つけることができる	117

第5章 組み立て方の壁 プログラムの要素と構造を押さえよう 119

5.1	プログラムの3大要素を知ろう	119
5.2	プログラムの3大制御構造を知ろう	123
5.3	順次構造を作ろう	123
5.4	分岐構造を作ろう	126
column	>=や<=を書き間違えないコツ	127
5.5	反復構造を作ろう	132
column	プログラマーはなぜ0から数を数えるのか?	135
▶ 発展 ▲	range関数は何を返すのか	136
5.6	プログラムの構造を組み立てよう	139
5.7	プログラムの組み立て方がわかった次のステップ	143
練習問題		143

第6章 練習から実用の壁 プログラミングでできることを増やそう 146

6.1	モジュールからさまざまな機能の関数を呼び出そう	146
column	必要なライブラリの情報にどのようにたどり着くのか	151
6.2	プログラムで複数のファイルの名前をまとめて変更しよう	152
column	「面倒」はプログラマーにとって大切な気づき	154
6.3	電話伝言メモ作成プログラムを作ろう	161
▶ 発展 ▲	エスケープシーケンスについて	164
▶ 発展 ▲	Visual Studio Codeで文字化けしてしまったら	170
6.4	プログラムを作ってみて見えてくる次のステップ	173

第7章 読みやすいコードの壁 可読性が高いプログラムを書けるようになるろう 174

- 7.1 プログラムが読みづらいと感じる原因は? 174
- 7.2 ソースコードに説明を入れよう 174
 - ▶ 発展 ▲ コメントアウトによって一部の行を一時的に無効にする 178
- 7.3 f-stringで文面の作成を簡潔に記述しよう 179
- 7.4 関数によってプログラムを意味を持った処理の固まりに分けよう 181
 - 練習問題 186
 - column 関数を作るときに気をつけること 188
- 7.5 モジュール化して関数の動作を個別に確認しよう 189
- 7.6 電話伝言メモ作成プログラムを改良しよう 192
- 7.7 完成したプログラムと元のプログラムを比べてみよう 196
 - column プログラムを書くにあたっての7つのアドバイス 198

第8章 アイデア実現の壁 プログラム作りの計画を立てられるようになるろう 201

- 8.1 次に進むためには「きっかけ」が必要 201
- 8.2 プログラミングとは「できること」を組み合わせる 201
- 8.3 自分がプログラミングでできることリストを作ろう 202
- 8.4 UIとロジックを分けて「作れそうなもの」を考えよう 203
 - column タプルとリストの使い分け 207
 - ▶ 発展 ▲ 値の交換や複数の値を返す関数ではタプルを使っていた 207
- 8.5 UIとロジックを組み合わせた完成形をイメージする 208
- 8.6 アイデアをストックし「難易度」「優先度」をつけよう 208
- 8.7 要件定義をしてアイデアをより具体的にしよう 210
 - ▶ 発展 ▲ プログラムは「ソフトウェア」 214
- 8.8 プログラムの設計をしよう 214
 - column ソフトウェア設計の目的は設計図を描くことではない 223
- 8.9 自分で考えたプログラムを作り始めることで見える世界 224

第9章 完成の壁 作り始めたプログラムを完成させられるようになるろう 225

- 9.1 トップダウン・プログラミングによって枠組みを作ろう 225

9.2	ボトムアップ・プログラミングで部品を作ろう	232
	▼ 発展 ▲ importのあとの「*」とは何か	235
	▼ 発展 ▲ shuffle関数はちょっと不思議な関数?	239
9.3	単体テストでバグを見つけよう	242
	column テストも専門スキル	246
9.4	デバッグをしてバグを修正しよう	250
	column 単体テストがプログラミングを楽にしてくれる	253
	▼ 発展 ▲ テストを自動化すること	255
9.5	結合テストで要件を満たすことを確認しよう	256
9.6	プログラムを改良して成長させよう	262
	▼ 発展 ▲ 改良の中でのリファクタリング	263
9.7	まとめとプログラム全文	264
	▼ 発展 ▲ V字モデルを知ろう	268

第10章 プログラマーらしさの壁 プログラマーのライフスタイルを知ろう 270

10.1	プログラマーのライフスタイルとは何か	270
10.2	リファレンス本を手元に置こう	270
10.3	別のプログラミング言語にも興味を持とう	271
10.4	プログラマー向けの雑誌を購読してみよう	272
10.5	資格に挑戦してみよう	273
10.6	イベントに参加して仲間のプログラマーに出会おう	273
	column オンライン化におけるコミュニケーションの課題と展望	281
10.7	日々の学びをアウトプットしよう	282
10.8	プログラマーの世界に一步を踏み出すための道しるべ	283
	column 新たなチャンスに巡り合おう	283

索引	285
著者プロフィール	287