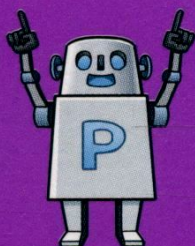


スツキリわかる Python

国本大悟／須藤秋良・著
株式会社 フレアリンク・監修

入門



シリーズ累計60万部突破!※ はじめの一步に最適な新定番書!

サクサク進めてしくみもバッチリ!
プログラミングの「本質」を
身に付け未来への
道を切り拓こう!



業務自動化、データサイエンス、人工知能、
アプリ開発プログラマへの第一歩!

インプレス

スッキリわかる
Python
入門

国本大悟／須藤秋良・著
株式会社 フレアリンク・監修

インプレス

本書の内容については正確な記述につとめました。著者、株式会社インプレスは本書の内容に一切責任を負いかねますので、あらかじめご了承ください。

本書に掲載している会社名や製品名、サービス名は、各社の商標または登録商標です。本文中に、TM および ® は明記していません。

インプレスの書籍ホームページ

書籍の新刊や正誤表など最新情報を随時更新しております。

<https://book.impress.co.jp/>

まえがき

著者の2人は、新入社員からベテランまで多くのエンジニアの学習を、研修を通じてお手伝いしています。従来、プログラミング言語研修といえばJavaが多かったのですが、近年はデータサイエンティストの需要の高まりを受け、Python研修の希望が増えています。研修後に初心者によく相談されるのが「今後ひとりで学習するのに、お勧めの入門書は？」ということです。そこで、数あるPython入門書に目を通してお勧めを探しましたが、初心者がひとりで学習するには難しすぎるか、逆に簡単すぎて今後につながらないものが多く、Pythonエンジニアを本気で目指す人の1冊目として自信を持って勧められる本を見つけられず苦悩しました。ないならば作ってしまおうと生まれたのが本書です。執筆に際しては、特に次の点を意識しています。

1. 今後に活かせる「基礎」を学べる

Pythonの利用分野は幅広く、また文法は多岐にわたるため、すべてを1冊の本にまとめることは非常に困難です。そこで、本書では初心者が利用する機会が少ない文法は思い切って割愛しました。プログラミング未経験者が基礎をしっかりと学べ、機械学習やWebアプリケーションといった専門分野の学習に進めることを目指しています。

2. 初心者でも「楽しく」学べる

Pythonの文法はシンプルでわかりやすいと言われますが、プログラミング初心者にとっては簡単ではありません。本書は、「スッキリわかる」シリーズで好評の親しみやすいイラストと柔らかい文章で仕上げています。初心者がつまずきやすい部分も、楽しくマスターできるでしょう。

3. 「ひとり」でも学べる

筆者はこれまでの研修を通じて、プログラミング言語学習の難しさは文法ではなく、トラブルシューティングにあると感じています。研修ならエラーが発生しても講師に質問して解決することができます。しかし、本での独習ではそうはいきません。そこで本書では、多くの若手エンジニアがよく起こしてしまうエラーやトラブルをできるだけ多く盛り込み、ひとりでも解決できるようにしました。

本書を通じて、読者のみなさまがPython並びにプログラミングの面白さに出会い、ひいてはエンジニアへの第一歩を踏み出すお手伝いできれば、著者としてこれ以上の喜びはありません。

著者

【謝辞】

本書の企画から発売まで多くのアドバイスとご支援をいただいた株式会社フレアリンクの中山清喬様、飯田理恵子様、インプレス編集部、イラストを担当してくださった高田様、私に教え方を教えてくれた教え子のみなさん、応援してくれた家族、その他この本に直接的、間接的に関わったすべてのみなさまに心より感謝申し上げます。

本書の見方

本書には、理解の助けとなるさまざまな用意があります。押さえるべき重要なポイントや覚えておく便利なトピックなどを要所要所に楽しいデザインで盛り込みました。読み進める際にぜひ活用してください。

本文中の色文字:

本文中、重要な用語や特に注意すべき部分に色をつけました。

1.3 データ型

1.3.1 データ型とは

これまで私たちは、数値と文字列の種類の値を使ってきました。数値や文字列といった値の種類のことをデータ型 (data type) または単に型といいます。Pythonでは、この2つの種類以外に、表1-6のようなデータ型を使うことができます。

コード6-5 RPGの勇者を表すクラスの定義と利用

```
class Hero:
    2     name = '松田'
    3     hp = 100
    4     def sleep(self, hours):
    5         print('{}は{}時間寝た!'.format(self.name, hours))
    6         self.hp += hours
    7
    8 # ゲーム開始
    9 print('スッキリファンタジーXII ~金色の理想郷~')
   10 h = Hero()
   11 h.sleep(3)
   12 print('{}のHPは現在{}です'.format(h.name, h.hp))
```

データとしてnameとhp、関数としてsleepを持つオリジナル設計図Heroを定義

HP100の勇者松田がオブジェクトとして誕生

予約語:

予約語 (P049) は色つきで表します。

注目コード:

注目すべきコードには、吹き出しに説明を入れて示しています。

コメント:

グレーの文字の部分はコメント (P026) です。

アイコン:

各アイコンの示す内容については、このページの下 (アイコンの種類) で確認してください。

リストから指定した値を削除

リスト.remove(リストから削除したい値)

削除した要素の後ろの要素は前に詰められる。

コード2-9 リストから要素を削除

```
1 members = ['工藤', '松田', '浅木']
2 members.remove('松田')
3 print(members)
```

リストから要素を削除

実行結果

['工藤', '浅木']



何で僕を削除しちゃうんですか。



松田のいた位置には私が詰めておくから安心してよね。ところで、追加と削除ができるなら、変更もできるんですか？



もちろん!!

リスト内の特定の要素の内容を変更するには、添え字を指定して代入します (コード2-10)。

吹き出し会話:

みなさんと一緒に学ぶ仲間たち (P013) が繰り広げる会話です。学びの場や開発現場でありがちな疑問点やひらめき、さらには重要なヒントが含まれていることも。ぜひ、お見逃しなく!

各章のまとめ:

その章で学んだことをまとめています。内容を正しく理解できているか確認し、達成度を測るチェック表として活用してください。

3.5 第3章のまとめ

この章では、次のことを学びました。

文と制御構造

- 1行に記述された1つの処理が実行単位であり、1つの文である。
- 文の実行順序は制御構造によってコントロールすることができ、主に順次・分岐・繰り返し (ループ) の3つがある。

条件分岐

- if文は、ある条件に当てはまる場合に処理を実行させることができる。
- if文は、条件が成立したらifブロックを、不成立だったらelseブロックを実行する。
- ブロックは、複数の文をひとまとまりとして扱う (文は1つでもよい)。
- ブロックはインデントによって指定する。

4.5 練習問題

練習4-1 次の各コードについて、繰り返しが行われる回数を考えてください。

```
(1) count = 0
while count < 5:
    count += 1

(2) count = 1
while count <= 5:
    count += 1

(3) data = [10, 20, 30, 40, 50]
count = 0
```

各章の練習問題:

各章の章末には練習問題があり、理解度を確認できます。理解できていない場合には、その章を読み返しましょう。

アイコンの種類



構文紹介:

構文の記述ルールと文法上の留意点などを紹介します。



Column:

本書では詳細は取り上げないものの、知っておくと重宝する補足知識やトリビアなどを紹介します。



ポイント紹介:

本文における解説で、特に重要なポイントをまとめています。

CONTENTS

まえがき	003
本書の見方	004

第0章 ようこそ Python の世界へ 011

0.1 ようこそ Python の世界へ	012
0.1.1 Python を使ってできること	012
0.1.2 一緒に Python を学ぶ仲間たち	013
0.2 はじめてのプログラミング	014
0.2.1 はじめてのプログラミング	014
0.2.2 エラーと上手に付き合う	016
0.2.3 セルとノートブック	018
0.3 Python プログラミングの基礎知識	020
0.3.1 開発の流れ	020
0.3.2 統合開発環境	022
0.3.3 プログラムの書き方	024
0.3.4 プログラミング体験を終えて	026

第I部 Python の基礎を学ぼう 031

第1章 変数とデータ型 033

1.1 式と演算	034
1.1.1 数値の演算	034
1.1.2 文字列の演算	036
1.1.3 エスケープシーケンス	039
1.1.4 式と評価	040
1.2 変数	044
1.2.1 変数の利用	044
1.2.2 変数名のルール	048
1.2.3 変数の上書き	050
1.2.4 まとめて代入 (アンパック代入)	052
1.2.5 自分自身への代入	053
1.2.6 複合代入演算子	056
1.2.7 キーボード入力値の代入	057
1.3 データ型	061
1.3.1 データ型とは	061
1.3.2 データ型の変換	065
1.3.3 文字列の中に数値を埋め込む	068

1.4	第1章のまとめ	073
1.5	練習問題	074
1.6	練習問題の解答	075
第2章 コレクション		077
2.1	データの集まり	078
2.1.1	変数が持つ不便さ	078
2.2	リスト	080
2.2.1	リストの特徴	080
2.2.2	リストの作成	081
2.2.3	リストの要素を参照	082
2.2.4	リスト要素の合計と要素数の取得	084
2.2.5	リスト要素の追加・削除・変更	086
2.2.6	高度な要素の指定	089
2.3	ディクショナリ	092
2.3.1	ディクショナリの特徴	092
2.3.2	ディクショナリの作成	093
2.3.3	ディクショナリ要素の参照	094
2.3.4	ディクショナリ要素の追加と変更	095
2.3.5	ディクショナリ要素の削除	096
2.3.6	ディクショナリとリストの比較	097
2.4	タプルとセット	100
2.4.1	タプル	100
2.4.2	セット	104
2.5	コレクションの応用	108
2.5.1	コレクションの相互変換	108
2.5.2	コレクションのネスト	110
2.5.3	集合演算	112
2.6	第2章のまとめ	116
2.7	練習問題	117
2.8	練習問題の解答	118
第3章 条件分岐		119
3.1	プログラムの流れ	120
3.1.1	文と制御構造	120
3.2	条件分岐の基本構造	123
3.2.1	if文	123

3.2.2	ブロックとインデント	127
3.3	条件式	131
3.3.1	比較演算子	131
3.3.2	in 演算子	132
3.3.3	真偽値	136
3.3.4	論理演算子	138
3.4	分岐構文のバリエーション	143
3.4.1	3種類のif文	143
3.4.2	if-else 構文	143
3.4.3	if のみの構文	144
3.4.4	if-elif 構文	147
3.4.5	if 文のネスト	150
3.5	第3章のまとめ	153
3.6	練習問題	154
3.7	練習問題の解答	156

第4章 繰り返し **159**

4.1	繰り返しの基本構造	160
4.1.1	while 文	160
4.1.2	無限ループ	164
4.1.3	状態による繰り返し	166
4.1.4	繰り返しによるリストの作成	167
4.1.5	繰り返しによるリスト要素の利用	169
4.2	for 文	171
4.2.1	for 文による繰り返し	171
4.2.2	for 文の基本構造	172
4.2.3	for 文による決まった回数の繰り返し	173
4.2.4	while 文と for 文の使い分け	175
4.3	繰り返しの制御	177
4.3.1	繰り返しの強制終了	177
4.3.2	繰り返しのスキップ	179
4.3.3	break 文と continue 文	181
4.4	第4章のまとめ	183
4.5	練習問題	184
4.6	練習問題の解答	187

第Ⅱ部 Python で部品を組み上げよう 191

第5章 関数 193

5.1 オリジナルの関数	194
5.1.1 関数の必要性和メリット	194
5.1.2 関数を使うための2ステップ	199
5.1.3 関数定義と呼び出し	200
5.1.4 ローカル変数と独立性	202
5.2 引数と戻り値	206
5.2.1 引数	206
5.2.2 複数の引数を渡す	208
5.2.3 戻り値	211
5.2.4 関数呼び出しの正体	214
5.2.5 関数の連携	217
5.3 関数の応用テクニック	220
5.3.1 暗黙のタプルによる複数の戻り値	220
5.3.2 デフォルト引数	221
5.3.3 引数のキーワード指定	225
5.3.4 可変長引数	226
5.4 独立性の破れ	230
5.4.1 グローバル変数	230
5.4.2 引数と戻り値の存在価値	233
5.5 第5章のまとめ	236
5.6 練習問題	237
5.7 練習問題の解答	241

第6章 オブジェクト 243

6.1 「値」の正体	244
6.1.1 format 関数の謎	244
6.1.2 オブジェクトの型	247
6.1.3 文字列オブジェクトが持つメソッド	248
6.2 オブジェクトの設計図	251
6.2.1 オブジェクトの姿を決定づける設計図	251
6.2.2 オリジナルの設計図を作る	254
6.3 オブジェクトの落とし穴	257
6.3.1 オブジェクトの identity	257
6.3.2 参照	259

6.3.3 参照による副作用	263
6.3.4 防御的コピー	265
6.3.5 不変オブジェクト	268
6.4 第6章のまとめ	274
6.5 練習問題	275
6.6 練習問題の解答	277

第7章 モジュール 279

7.1 部品を使おう	280
7.1.1 Python で使える部品たち	280
7.2 組み込み関数	281
7.2.1 組み込み関数とは	281
7.2.2 ファイル入出力	282
7.3 モジュールの利用	287
7.3.1 モジュールとは	287
7.3.2 標準ライブラリ	289
7.3.3 モジュールの取り込み	289
7.3.4 特定の変数や関数だけを取り込む	292
7.3.5 ワイルドカードインポート	295
7.3.6 モジュール取り込みのまとめ	297
7.4 パッケージの利用	299
7.4.1 パッケージとは	299
7.4.2 パッケージ内のモジュールを取り込む	300
7.5 外部ライブラリの利用	304
7.5.1 外部ライブラリとは	304
7.5.2 外部ライブラリの準備	305
7.5.3 matplotlib	306
7.5.4 requests	309
7.6 第7章のまとめ	313
7.7 練習問題	314
7.8 練習問題の解答	316

第8章 まだまだ広がる Python の世界 319

8.1 Python の可能性	320
8.1.1 まだまだ広がる Python の世界	320
8.1.2 ルーチンワークの自動化	321
8.1.3 データベースの操作	322
8.1.4 ウィンドウアプリケーションの作成	324