

Google Colaboratory でプログラミング力と
 計算論的思考力を身につけられる！

計算論的思考を育む Python プログラミング入門

著者：綾 皓二郎

仕様：B5 判・並製・印刷版モノクロ / 電子版一部カラー・
 346 頁

印刷版・電子版価格：3,500 円（税抜）

ISBN：978-4-7649-6050-3 C3004

発行：近代科学社 Digital

発売：近代科学社

内容紹介

本書では、手続き型からオブジェクト指向へ向かうプログラミングの進化の道筋を辿りながらアルゴリズム的思考とプログラミング的思考を培い、Google Colaboratory を使ったプログラミング力と計算論的思考力を身につけることができます。

第 1 章はプログラミング全般に関わる説明、第 2 章はプログラミングの準備、第 3 章～第 5 章までに小さなプログラムの作成を学びます。第 6 章と第 7 章では関数やモジュール、ライブラリの使い方、第 8 章ではファイルの取り扱い、第 9 章では手続き型プログラミングの総まとめを解説。第 10 章はオブジェクト指向プログラミング（OOP）の基本的な考え方、およびクラスやオブジェクトの作り方、第 11 章は Anaconda を用い GUI アプリの作成について扱います。プログラミング言語の文法を網羅的に説明している入門書とは一線を画する内容であり、初心者だけでなく Python に既に触れている中級者にも最適な参考書です。

著者紹介

綾 皓二郎（あや こうじろう）

石巻専修大学名誉教授

1968 年東北大学工学部卒業、1970 年東北大学大学院工学研究科修士課程修了、(株)東芝を経て、1975 年東北大学医学部助手、生体情報科学・神経科学を専攻、1986 年医学博士（東北大学）、1989 年石巻専修大学理工学部助教授、1993 年石巻専修大学理工学部教授、「情報システム概論」「情報理学」などを担当、大学入試センター試験『情報関係基礎』作題委員（2002，2003 年度）、2015 年石巻専修大学退職（石巻専修大学名誉教授）



全国の書店・ネット書店にてお求めいただけます。お取り扱い店は以下のウェブページをご覧ください。

https://www.kindaikagaku.co.jp/book_list/detail/9784764960503/



近代科学社 Digital

<https://www.kindaikagaku.co.jp/kdd/>

近代科学社 Digital は、株式会社近代科学社が推進する 21 世紀型の理工系出版レーベルです。デジタルパワーを積極活用することで、オンデマンド型のスピーディで持続可能な出版モデルを提案します。

お問い合わせ先

株式会社近代科学社

〒101-0051 東京都千代田区神田神保町 1-105
 神保町三井ビルディング

電子メール：contact@kindaikagaku.co.jp

目次

第1章 プログラミングと計算論的思考

- 1.1 いくつかの用語の説明
- 1.2 アルゴリズムの構築とプログラミング
- 1.3 プログラミングという学び
- 1.4 計算論的思考とは何か
- 1.5 プログラミング言語の基礎知識
- 1.6 プログラミングに関連する予備知識

第2章 プログラミングの準備

- 2.1 統合開発・学習環境 Google Colaboratory
- 2.2 Google Colaboratory(Colab) の立ち上げ
- 2.3 翻訳サービスとオンライン辞書, 計算サービス
- 2.4 初めてのプログラミング
- 2.5 プログラミングに関する注意事項など

第3章 順次構造と処理

- 3.1 構造化プログラミング
- 3.2 変数とデータ型, 演算子, 算術演算
- 3.3 リストと関数, メソッド
- 3.4 文字列と関数, メソッド
- 3.5 ハノイの塔のパズルを解く
- 3.6 エラーとエラーメッセージ, バグ, 例外

第4章 条件分岐構造と処理

- 4.1 ブール型と論理演算子
- 4.2 比較演算子
- 4.3 オブジェクトと識別番号 id
- 4.4 条件分岐による処理の流れの制御
- 4.5 条件分岐処理① if 節
- 4.6 条件分岐処理② if 節, else 節
- 4.7 条件分岐処理③ if 節, elif 節, else 節

第5章 繰り返し構造と処理

- 5.1 コンテナ/コレクション
- 5.2 繰り返し処理の必要性和威力
- 5.3 for 節とイテラブル, イテレータ
- 5.4 range, for 文, 内包表記
- 5.5 for 節を用いる繰り返し処理
- 5.6 while 節を用いる繰り返し処理
- 5.7 繰り返しの中断

第6章 関数の定義と自作

- 6.1 関数 (function) とは何か
- 6.2 高階関数 (higher-order function)
- 6.3 無名関数, ラムダ式
- 6.4 ジェネレータ関数とジェネレータ式
- 6.5 ユーザー定義の関数とジェネレータ関数の作成
- 6.6 再帰関数と再帰呼び出し, 分割統治

第7章 モジュールとライブラリの利用

- 7.1 モジュール (module), ライブラリ (library) とは
- 7.2 モジュールの自作とインポート
- 7.3 ライブラリの種類
- 7.4 標準・外部ライブラリを使う
- 7.5 Linux OS と IPython のコマンド

第8章 ファイルの入出力と例外処理

- 8.1 テキストファイルへの出力, 文字列データの書き込み
- 8.2 テキストファイルからの入力, 文字列データの読み込み
- 8.3 データの入出力にファイルを用いる処理プログラム
- 8.4 CSV ファイルの書き込みと読み込み
- 8.5 画像ファイルの読み込みと書き込み
- 8.6 例外と例外処理

第9章 アルゴリズムと計算論的思考

- 9.1 暗号 (cipher, cryptogram)
- 9.2 コンピューターによる数値計算
- 9.3 スタックを用いた四則演算
- 9.4 ソート (sort, 整列)
- 9.5 探索 (search, サーチ)

第10章 オブジェクト指向プログラミング

- 10.1 手続き型言語からオブジェクト指向言語への拡張
- 10.2 オブジェクト指向プログラミングとは何か
- 10.3 クラスベースのオブジェクト指向プログラミング
- 10.4 標準ライブラリのモジュールにあるクラスの利用
- 10.5 オブジェクト指向プログラミングの三大要素

第11章 Anaconda によるプログラミング

- 11.1 Anaconda のプログラミング環境
 - 11.2 タートルグラフィックス
 - 11.3 OOP による GUI アプリの作成
-