

自学自習に配慮した Python プログラミング教科書の開発

喜多 一¹⁾, 森村吉貴²⁾, 岡本雅子¹⁾

1) 京都大学 国際高等教育院

2) 京都大学 情報環境機構

kita.hajime.7a@kyoto-u.ac.jp

Development of Python Programming Textbook Considering Self-Learning

Hajime Kita¹⁾, Yoshitaka Morimura²⁾, Masako Okamoto¹⁾

1) Institute for Liberal Arts and Sciences, Kyoto University

2) Institute for Information Management and Communication, Kyoto University

概要

京都大学では 2018 年度より全学共通科目としてプログラミング演習 (Python) を開講しており, 2019 年度からはこの科目での利用を目的に教科書を開発し, 以後, 毎年改訂している. この教科書は授業時間以上に授業時間外学習が求められる大学の授業とプログラミングの初学者の履修を前提に自学自習に配慮して開発されている. ここでは一方で, 初学者が陥りがちな躓きに配慮し, 他方で簡単なプログラムを自ら設計, 実装, テストできるようになることを目標としている. 本稿では教科書開発の背景, 編集上の考慮事項, 電子化と公開, 授業実践を通じた改善, 残されている課題などを論じる.

1 はじめに

京都大学の教養・共通教育/全学共通科目では 2016 年度に科目の編成等を見直し [1], 情報系の基礎科目として「情報基礎演習」を開講した. 同科目は理系学部については学部・学科ごとに内容を調整して実施しているが, 文系学生などを対象として [全学向] としても開講されている. この全学向科目では科目の評価基準 (ルーブリック) を作成し, それに沿った教科書を 2017 年度から開発・使用している [2]. その中で一部, プログラミング言語として Python を取り上げ, 導入的な内容も扱っている.

他方, 全学共通科目としてプログラミングをより本格的に扱う科目もいくつか開講されていたが, Python を扱う科目がなかったことから, 情報基礎演習を学んだ学生など Python を本格的に学びたい学生に提供する科目としてプログラミング演習 (Python) を 2018 年度から開講し著者のうち喜多が担当した*1. 2019 年度には同科目用に教科書を開発し学生に電子版を無償で提供している. その後, この教科書は授業実践を

踏まえて毎年度, 改訂しており, 2021 年度版では留学生に配慮して英訳も行った*2. 2019 年度版 [3], 2021 年度版 [4, 5](図 1) については京都大学学術情報リポジトリ (KURENAI)[6] でも公開している.

この科目はプログラミング経験のない初学者を対象としているが, 科目の目標としては「簡単なプログラムを自ら設計, 実装, テストできるようになること」も掲げている*3. 同科目の実践についてはその概要を [7] で報告しているが, 本稿では同科目用に開発している教科書について, 開発の背景, 編集上の考慮事項, 授業実践を通じた改善, 残されている課題などを論じる.



図 1 教科書 プログラミング演習 Python 2021 年版 (和英)

*1 2020 年度から森村が喜多と共同で, 2021 年度からは 1 科目増やして岡本・喜多で担当している.

*2 現在では別に外国人教員による英語での科目 Programming Practice (Python) -E2 が開講されている.

*3 2023 年度の授業のシラバスは [8] 参照

2 大学での授業とプログラミングの学習

大学での授業については大学設置基準で1単位は45時間の学修を必要とする内容とされ、そのうち15～45時間の範囲で大学が定める時間の授業で1単位と計算するとされている。広く実施されている週1コマ(2時間)、半期15回で2単位を発給する科目では90時間の学修内容のうち、30時間を授業として行い、残り60時間分の学修内容は授業時間外の学習を充てることになる。

プログラミングの学習についてはプログラミングを演習する環境が必要となる。各大学においてノートPCの必携化が進み、またプログラミング言語の処理系も無償で利用できるものが広がっており授業時間外の学習に関して講義科目などで一般的に採用されている単位数での実施は可能である。

他方、プログラミングについては、初学者には学習においてさまざまな躓きがある[10]。授業時間外の学習での躓きは学習進度を低下を招き、また学習動機の喪失にもつながる。このため授業時間外の学習の支援がプログラミング授業実践上のポイントであり、本授業では以下の3点を考慮して運営している。

- 予習、復習とも受講者自身で学習を進めやすい教材を準備する。
- 教員が即応できる授業時間内と即応が難しい授業時間外とで反転授業の考え方も取り入れ学習内容を調整する[9]。
- 授業中の課題、宿題などはLMSで提出させ、極力、週内に採点し個別にフィードバックする。

3 科目設計と授業計画

対象科目は京都大学の全学共通科目の選択科目として開講している半期週1コマ、2単位科目であり、前期に先の情報基礎演習を履修した学生の受講を想定して後期に配当している。

文系学部を含む全学部の学生が受講できる科目であり、またプログラミング経験を持たない学生の受講も想定しているが、他方で半期にわたってプログラミングを学ぶ科目の目標として、Pythonを用いて「簡単なプログラムを自ら設計、実装、テストできるようになること」も設定している[8]。

学部の専門科目と異なり、全学共通科目では履修者に学部1年生が多く、その知識と興味関心に適合すること、特定の応用領域を想定しにくいこと、必修性が

ないため履修を途中で諦める学生が生じやすいことへの配慮が必要である。

教科書を用いた2019年度の授業計画を表1に示す。第2回の授業で扱う内容が多いように見えるが、後の授業で繰り返し利用することを通じて学ぶことを想定している。授業は教科書に沿って単元ごとに進めるが、途中でタートルグラフィクスで自由な作品を作成する課題に取り組みせるとともに、最後の数週を各自が設定した目標に応じてプログラムを作成するプロジェクトに充てる運営となっている。

4 教科書開発の背景

主要なプログラミング言語については、すでに多くの教科書が市販されているが、その中で敢えて独自の教科書を開発したことについては以下の理由による。

- 教員はプログラミングの学習について学習者のさまざまな躓きに出会い、それを授業内で解決している。この知見を暗黙知とせず教科書に形式知として記述することで授業の改善を行いたい。
- プログラミングの授業はややもするとプログラミング言語、さらにはその文法の紹介に終始しがちである。他方で初学者は100行程度のプログラムであっても、自分自身でプログラムを書くことに困難を感じる。そこで、数百行程度のまとまった量のプログラムを例示し、設計から実装、テストへと進むプログラム作成のプロセスを体験的に知ってもらいたい。
- プログラミングの授業ではCUIベースの入出力を扱うことが多い。しかしながらWindowsなどGUIベースの環境からコンピュータに触れ始めた初学者にとってはCUI環境でのOSの操作などは馴染みが少なく、応用も意識しづらい。他方でGUIはイベント駆動の考え方でのプログラミングが求められ、フレームワークの相違などを学ぶ必要がある。
- 授業のため市販の教科書を購入させるのではなく無償で配布したい。

5 教科書開発における初学者への配慮

初学者を対象としたプログラミングについて、筆者らの研究成果[10, 11]を踏まえ、またPythonというプログラミング言語や標準的に提供されているライブラリなどを考慮して教科書の開発に際して以下のような配慮を行った。

表1 2019年度の授業計画と教科書

回	授業計画	2019年度版教科書
1	ガイダンス, 実行環境	1, 2 章
2	式と変数, 制御構造	3 章 (式と変数) 4 章 (制御構造)
3, 4	タートルグラフィクス	5 章 (関数) 6 章 (タートル)
5, 6	Tkinter による GUI	7, 8 章 (GUI) 9 章 (クラス)
7	リストの利用	10 章 (リスト)
8	ファイル入出力	11 章 (ファイル)
9, 10	プログラム開発	12 章
11	Python の学術利用	13 章 (numpy, matplotlib, pandas)
12	振り返り	14 章
13, 14	各自のプロジェクト	—
15	試験	—
16	試験の講評	—

- Python 以外のプログラミング言語を混在させない。初学者にとって複数の言語を扱うことは混乱を招くおそれがあり、Python に集中して学んでもらうこととした*4
- 文法事項に関わるプログラムの一部ではなく、動作するソースコード全体を掲載する。文法的な事項を説明する際に、それを確認できるプログラム全体を示している例を図2に示す。これにより、プログラム全体を構成することに困難を抱える初学者でも単元で扱う事項の動作を確認できる。ただし、対話的な使用も可能な Python の特性も踏まえ、多数の行に渡るスクリプト実行と動作確認をするシェルでの短い入力での対話とを併用する記述を採用した。
- キーボードやそこでの記号の読み方など基本的な事項から説明する。これは指導者とのコミュニケーションを円滑化することを通じて初学者への支援を改善するためである*5。
- 関連知識としての数学との差異の説明。プログラミングについては初学者であっても大学生は中学、高校レベルの数学的な知識は持っている。そこで変数、式、関数などの説明に際しては数学の知識との橋渡し（類似性と差異）に配慮した。
- プログラミングに際して遭遇するエラーなどの

*4 このため、HTML などにも言及しなければならない Web 関連の話題は取り上げていない。

*5 文献 [11] から継承し、Python 用に調整している

明示的な体験。初学者はプログラムでエラーが出た時に適切な対応がとれず学習が止まりがちである。そこで典型的なエラーについて明示的に説明したり、意図的に体験し、エラーメッセージを読む演習を設定したりしている。

- 関数、クラス型オブジェクトのプログラミングに先立つ使用経験を通じた理解。タートルグラフィクスや GUI アプリケーションの作成を通じて、関数やクラス型オブジェクトについて、その定義のプログラミングに先行して使用経験を持たせることを通じて理解させるようにした。
- プログラミング言語の説明だけでなく、プログラムの設計やテストについて説明を含める。文献 [11] で取り上げた「三目並べ」の例題について、Python で書き換え、プログラムの設計から、コーディング、テストまでのプロセス全体を体験できるようにした。これは個人プロジェクトの円滑な実施のための「型」や「ヒント」となることも期待している。
- 授業での主たる進行とは別に、理解や興味の促進のための並行する話題をコラムとして記述している。こちらはまとめて別冊のコラム編として編集している。
- 多様な学部 of 1 年生が多数受講することから、学習者の動機付けが重要である。学習の動機付けについての ARCS 理論 [12] などを勘案し、グラフィカルな内容として以下の3種類を取り上げている (図3参照)。

タートルグラフィクス プログラムの動作を可視化するとともに、関数の多様な使い方を理解する。また Python ではクラス指向でのプログラミングとして複数のタートルを扱うことが可能であり、使用を通じてクラスオブジェクトを学ぶ役割も期待している。さらに授業ではタートルグラフィクスを用いた自由作品の制作を課しており、その際のヒントになる例題も取り上げている。

tkinter による GUI アプリケーション Python で標準的に使え、複数の OS に対応した GUI フレームワークである tkinter を用いたアプリケーションのプログラミングを紹介している。これは GUI 型のアプリケーションに慣れた初学者への動機づけとともに、イベント駆動型のプログラミングを理解してもらうこと、時計の例題を通じてグラフィクスの描画

に加え自立的に動くプログラムと GUI のイベント駆動との協調を説明し、個人のプロジェクト課題へのヒントとしている。

さらにファイルの扱いの際に CUI 環境でのファイルパス指定の負荷を軽減するため GUI ベースのファイル選択を紹介している。

matplotlib によるグラフ作成 多様な学術分野でのプログラムの利用に配慮して matplotlib でのグラフ作成を取り上げた。またこれに先立って csv 形式のファイルを通じて数値計算を行う Python プログラムと表計算ソフトによるグラフ作成とを連携させることも取り上げている。

- このほか、コンピュータやプログラミング言語についての背景知識と学習法の説明を含めるとともに、初学者が学びやすい Python の実行環境としての IDLE の採用なども行っている。

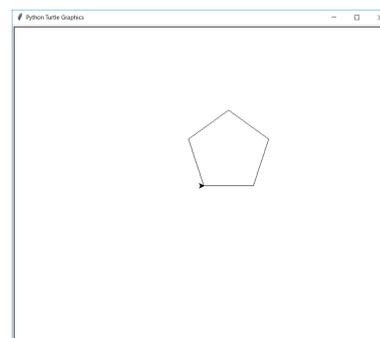
行	ソースコード	説明
1	<code>#!/x_の平方根を求める</code>	
2	<code>x_ = 2</code>	
3	<code>#</code>	
4	<code>rnew_ = x</code>	
5	<code>#</code>	
6	<code>diff_ = rnew_ - x/rnew</code>	2つの近似値(xと1(=x/x))の差をとる
7	<code>if diff_ < 0:</code>	負なら符号を反転。
8	<code>.....diff_ = -diff</code>	
9	<code>while (diff_ > 1.0E-6):</code>	差が 10 ⁻⁶ より大きければ繰り返す
10	<code>.....r1_ = rnew</code>	
11	<code>.....r2_ = x/r1</code>	
12	<code>.....rnew_ = (r1_ + r2_)/2</code>	
13	<code>.....print(r1_, rnew_, r2_)</code>	
14	<code>.....diff_ = r1_ - r2</code>	差の再計算
15	<code>.....if diff_ < 0:</code>	
16	<code>.....diff_ = -diff</code>	

図2 K2PFE フォントを用いた教科書でのソースコードの記載

6 組版とフォント

教科書の書式・組版方法にはいくつかあるが、本教科書については多様な環境で利用に配慮して、通常の文書の形式をとった。組版は情報基礎演習の教科書[2]用に開発したテンプレートにスタイルを追加する形で Microsoft Word で行い、履修者への配布は PDF 形式で行っている。

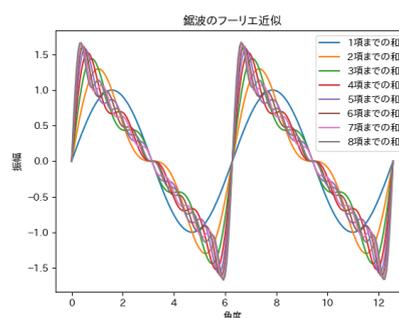
ただし、PDF では複数文字の空白の列について、コピー&ペースト可能なその文字数の情報を持たない。このため、ブロックの記述上、字下げの深さが重要な意味を持つ Python ではソースコードの適切な記述が難しい。そこで 2021 年度版から、この教材と実際の授業での利用の両方を狙って IPA のゴシックフォントに改変を加える形でフォント (K2PFE) を開発し使用している。開発の要点は以下である。



(a) タートルグラフィックスの例



(b) tkinter による GUI アプリケーションの例



(c) matplotlib によるグラフ描画の例

図3 教科書でのグラフィカルな素材の採用

- 空白の代わりに空白記号 () をソースコードの記述に用いることとし、文字幅を半角文字に揃えた。組版の際には4文字づつ色を変えている(図2参照)。これによりソースコードの字下げの深さを可視化するとともに、コピー&ペーストでのソースコードの利用に配慮した。
- 初学者の躓きの一つにソースコードのタイピングにおける全角、半角の文字間違いがある。そこで全角の空白を可視化するとともに半角文字と同じ意味をもつ英数字や記号については下線を付与し区別を容易にした。
- カンマとピリオド、コロんとセミコロンなど差異の少ない文字について差異を強調し、視認性を高

めた。

- Windows では UTF-8 の利用においても標準フォントではバックslashコードに円記号の字形が用いられている。授業では Mac ユーザも居り、混乱を招くため、バックslashについて本来の字形としている。

なお、このフォントの導入により、学生が課題として提出するプログラムでの字下げの質が向上した。また教員側ではソースコードを含む教材の編集効率が改善した。

7 電子版での提供・ライセンス・公開

本教材は作成当初から、学生には電子版を無償提供することを意図していた。ページ数が多く、カラーの図も多用するため出版社からの出版は費用面で不適当なこと、次に述べる授業実践を通じた継続的改訂を考えていたことによる。

また、この授業は履修希望者が教室定員を超えていることから履修できなかった学生への学習機会を提供することや、文献 [11] を知る研究者から、同書の Python 版が臨まれるという意見を得ていたことから教科書を公開した。自由な利用に配慮してクリエイティブコモンズライセンス（表示 - 非営利 - 改変禁止 CC BY-NC-ND）を付与している [13]。

公開のプラットフォームとしては、付与される URL が安定して運用されることから教材の公開への利用の可否を確認したうえで京都大学学術情報リポジトリ KURENAI[6] を選んだ。このサイトでは 2019, 2021 両年度版の合計で 103 万件を超えるダウンロード (2023/9/21 現在) があるなど多数の利用を頂くとともに、優秀教材としても評価された [14]。

8 授業実践を通じた改善

先のフォント開発も授業実践を通じた改善であるが、学生の受講を通じた教材と授業計画の改善も行われている。本授業では自由な課題も課しており、教材は単元の内容紹介だけでなく、自由課題へのヒントの提供も担っている。このため授業での学生の躓きへの配慮に加え、自由課題を通じた学生の興味・関心の対象も可能なものは教科書に反映している。授業実践を通じた教科書の改善サイクルを図 4 に示すが、教材改善はまず毎回の授業で補足資料や追加課題など提供し、それを毎年度まとめて翌年度の教科書の改訂に反映している。

また全学共通で実施される選択科目は、受講を途中で取りやめる学生が少なくない。図 5 に 2020 年度の授業における課題の提出状況を示す*6。毎回 2 つほど課題を課しているがタートルグラフィックスの単元に入る 6 回目以降で提出数の減少が目立つ。大学が実施する授業評価アンケートでもこのあたりからプログラムの行数が多くなることを指摘する意見を頂いた。そこで、2021 年度からはリストの単元を関数やタートルグラフィックスより前に移動することで、プログラムの行数の急増を緩和し、それを踏まえて教科書も見直した。併せて、この単元が終了した段階で複数の課題から選択する形で力試しの課題を追加導入した。対象となる学生が異なるので比較は難しいが 2021 年度以降の受講状況は改善している。なお、力試し課題については 2023 年度版の教科書に反映させている。

このほか、プログラミングの最初の段階 (3 章, 4 章) で例題として四則演算を用いた平方根の近似計算を取り上げた。授業では口頭での説明も行うため特に支障はなかったが、自学自習で使う際に躓きとなるとの指摘を複数頂いた。これについては 2023 年度版の教科書では内容は変えないものの題材の導入時の説明を丁寧にするようにした。

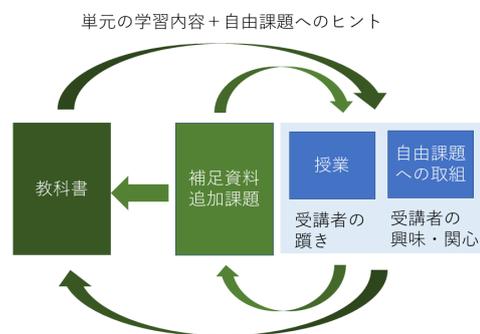


図 4 教材改善ループ

9 おわりに

本稿では静的なメディアである教科書について報告したがプログラミングにおける操作も作成したプログラムの実行も元来、動的なものである。この点について、静的なメディアでは分かりにくい点があり、授業では操作の様子を録画したものなどを動画教材として

*6 この年度はコロナ禍のため全面的なオンライン授業として実施し、教室を用いないため履修定員の制限をなくしたことから当初 200 人程度が科目に登録した。



図5 2020年度の授業での課題の提出状況

併用している。

また Python の利用は多方面に広がっているが、多様で具体的なプログラミングの応用については学習者自身による継続的な学習のための適切な橋渡しが必要である。

さらに、大規模言語モデルに基づく生成 AI の利用が多方面で注目されているが、プログラミングの支援はその応用として注目されている (例えば [15])。これをどのように活かすかも重要な課題である。

謝辞

教科書で利用している K2PFE フォントは著者のうち喜多と京都市立芸術大学教授、辰巳明久氏、同非常勤講師、楠麻耶氏、京都大学助教 (当時)、元木環氏との共同研究により開発したものです。また、2021 年度版の教科書の英訳については京都大学国際高等教育院附属データ科学イノベーション教育研究センターから経費の補助を得て実施しました。さらに、2021 年度からの力試し課題の追加については、本授業について 2021 年 9 月に開催した京都大学学術情報メディアセンターセミナーでの報告について熊本大学教授 (当時) 鈴木克明氏に頂いたご意見を参考にしました。ここに謝意を表します。本研究は JSPS 科研費 JP 21K02880 の助成を受けたものです。

参考文献

[1] 京都大学 国際高等教育院: 新しくなった教養・共通教育 (2019) <https://www.z.kyoto-u.ac.jp/pdf/link/link0736.pdf?1566378573> (2023/9/21 アクセス)

[2] 喜多, 北村, 日置: 情報基礎演習 2017, <http://hdl.handle.net/2433/224813> (2017)

[3] 喜多: プログラミング演習 Python 2019, <http://hdl.handle.net/2433/245698> (2020)

[4] 喜多, 森村, 岡本: プログラミング演習 Python 2021, <http://hdl.handle.net/2433/265459> (2021)

[5] H. Kita, Y. Morimura and M. Okamoto: Programming Practice Python 2021, <http://hdl.handle.net/2433/265460> (2021)

[6] 京都大学学術情報リポジトリ (KURENAI) <https://repository.kulib.kyoto-u.ac.jp/dspace/> (2023/9/25 アクセス) <https://www.ipsj.or.jp/award/yusyukyoza.html> (2023-9-20 アクセス)

[7] H. Kita, Y. Morimura and M. Okamoto: Design and Practice of an Elective Python Programming Course in General Education, abstract-book, p. 172, WCCE 2022 (2022)

[8] 喜多, 森村: プログラミング演習 (Python) 2023 年度シラバス, https://www.k.kyoto-u.ac.jp/external/open_syllabus/1a_syllabus?lectureNo=52731 (2023/9/25 アクセス)

[9] 喜多, 岡本: プログラミング教育における反転授業の一構成法, AXIES: 大学 ICT 推進協議会 2016 年度 年次大会 (2016)

[10] 岡本: 模倣の重要性に着目した初学者向けプログラミング教育の研究, 京都大学 学位論文 (2014)

[11] 喜多, 岡本, 藤岡, 吉川: 写経型学習による C 言語プログラミングワークブック, 共立出版 (2012)

[12] ジョン・ケラー著, 鈴木克明 (監訳): 学習意欲をデザインする: ARC Sモデルによるインストラクショナルデザイン, 北大路書房 (2010)

[13] Creative Commons — 表示 - 非営利 - 改変禁止 4.0 国際— CC BY-NC-ND 4.0 <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.ja> (2023/9/25 アクセス)

[14] 情報処理学会 2021 年度優秀教材賞, 喜多一, 森村吉貴, 岡本雅子 (京都大学), Python プログラミング学習のための教科書の作成と無料公開,

[15] 倉光 君郎: ChatGPT と Colab の連携: プログラミング教育における ChatGPT の活用事例, 【第 65 回】 大学等におけるオンライン教育とデジタル変革に関するサイバーシンポジウム「教育機関 DX シンポ」, 国立情報学研究所 (2023)