

自学自習に配慮したPython プログラミング教科書

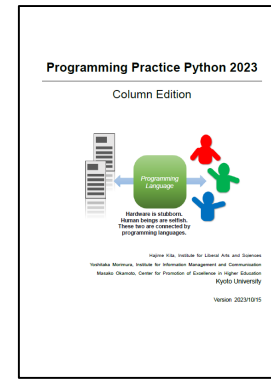
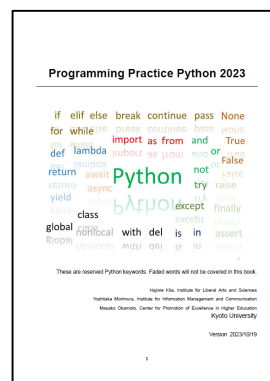
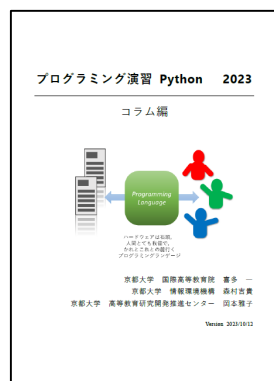
京都大学 国際高等教育院 喜多 一

報告の概要

- 教科書: プログラミング演習 Python
- 教科書作成の背景
- 教科書開発と編集のポイント
- 電子版での配布と公開
- 今後の課題

教科書: プログラミング演習 Python

- 教科「プログラミング演習 (Python)」用に2019年に開発, 以後, 毎年改訂
- 公開: 2019年版, 2021年版(和英), 2023年版(和英)



2023年版は京都大学学術情報リポジトリで公開
和文版 <http://hdl.handle.net/2433/285599>,
英文版 <http://hdl.handle.net/2433/285600>

教科書作成の背景

科目開設と担当の背景

- 2016年に京都大学の教養・共通教育の科目再編
 - 情報系の演習科目として「情報基礎演習」を開設。
内容は学部・学科に応じて調整
 - 文科系を中心に履修する科目用に2017年度に教科書を作成。プログラミングの単元にPythonを採用。
- 2018年度にPythonの授業を開設
 - Java, Lisp, Rubyなどのプログラミング科目をすでに開講、計算機科学系の教員はこれらの担当で多忙。
 - 科目の設計と担当が喜多に要請された。

プログラミング演習 (Python) : 科目の概要

- 京都大学の全学共通教育科目, 選択科目
 - 文科系を含む全学部の学生を対象に開講
 - 後期開講, 週1回2時間, 2単位の演習科目
 - 2019 年度から教科書を開発して使用
- 科目の到達目標
 - Python によるプログラムの実行についての基本操作ができるようになる。
 - Python プログラムを構成する基本的要素の機能と書式について説明し、例題を用いて実行例を構成できるようになる。
 - 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。

開講状況

年度	クラス	備考
2018	喜多	スライド教材
2019	喜多	教科書 2019
2020	喜多・森村	教科書 2020 オンライン実施
2021	喜多・森村／岡本・喜多	教科書 2021, 2クラス化 単元構成見直し
2022	喜多・森村／喜多	教科書 2022
2023	喜多・森村／岡本・喜多	教科書 2023

大学の授業での自己学習

- 日本の大学での2単位科目:

- 学修内容：90 時間.

- 授業（通常）:

- 2 時間/週 × 15 週/半期 =
30 時間

- 60 時間の自己学習を求めている.

- プログラミングの自己学習を支援するか

- プログラミングの学習は躓きやすい

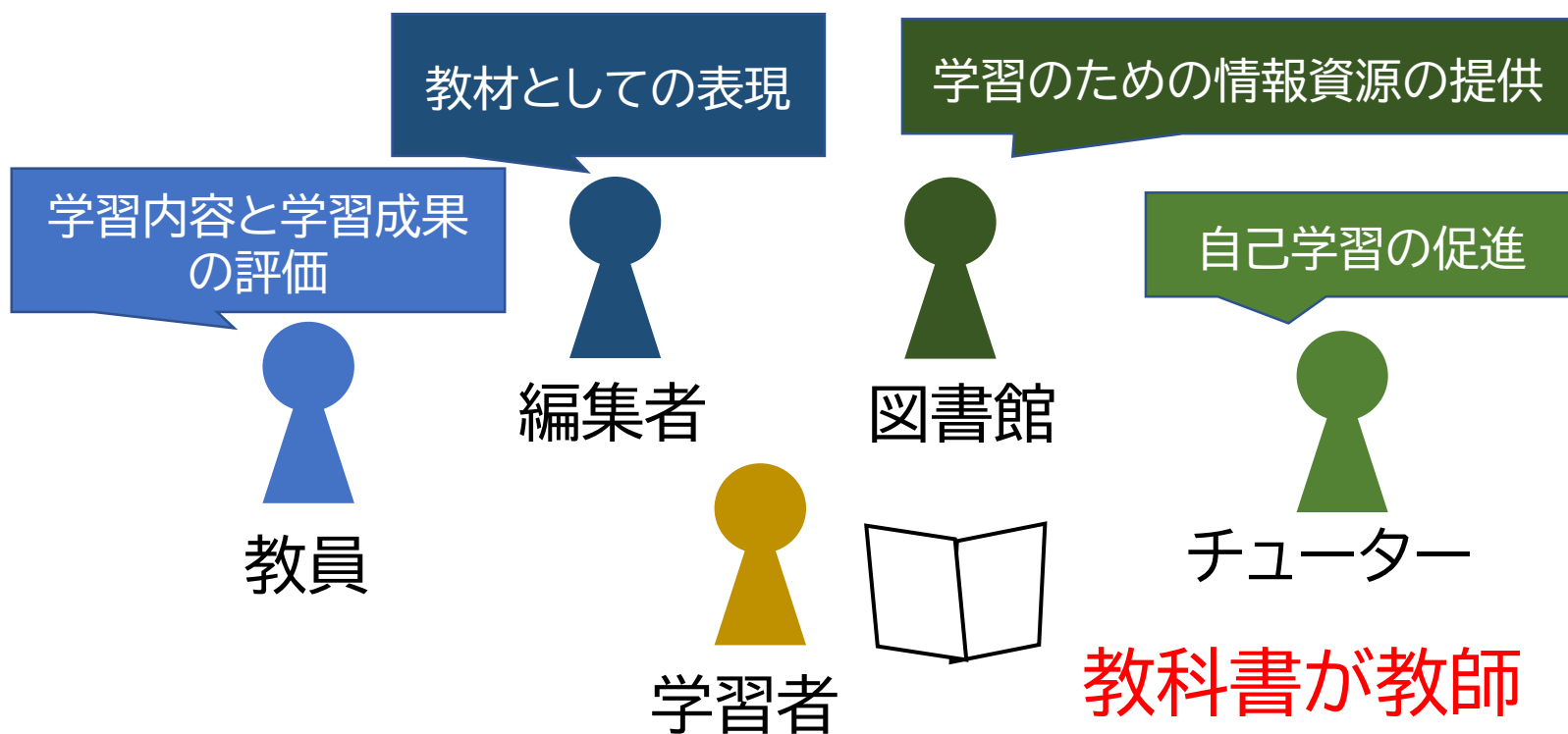
- 授業構成 + 教材で支援



2単位科目

英国 Open University を訪問(2002)

- Dominic Newbould, Visitor and Community Relation Manager, at OUにインタビュー
- 彼の説明, "OU には教員は居るが, 教師はいない"



プログラミング科目の反転授業

- 自己学習で躓かせない
 - 予習:教科書を読んできて, **教科書の例題のコードを作成し実行できることを確認する.** 宿題として課題提出.
- 授業:反転授業 — 予習した例題のコードをベースに **改変等の課題に取り組む**
 - 授業時間中なら教師やTAが個々の学生の支援できる.
 - 学生の多様な躓きを教師が把握できる.
- 提出された課題への素早いフィードバック
 - LMS で提出された課題は次回の授業までに(必要なら)コメントを付けて採点する.

動機付けへの配慮: ARCS モデル

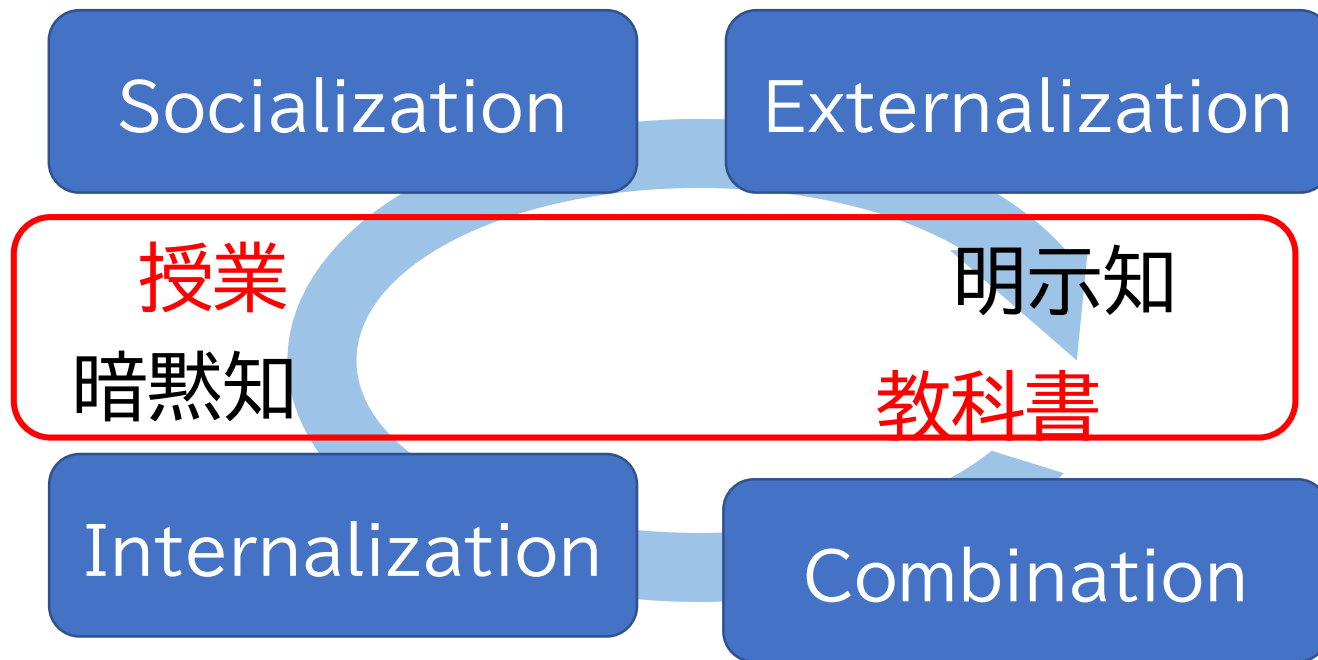
John M. Keller: Motivational design for learning and performance Model of learners motivation, Springer (2010)

- 鈴木訳: 学習意欲をデザインする: ARCS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)

項目	我々のアプローチ
ATTENTION	導入科目, 公開の教科書
RELEVANCE	学習者の背景知識への接続 グラフィカルな応用
CONFIDENCE	毎週の課題と個別コメント付きの採点の フィードバック
SATISFUCTION	自由度のある3回の力試し課題とその採点

SECIモデルと教室・教科書

SECI モデル: 野中, 竹内の知識創造モデル



教室での質疑や課題の採点を暗黙知のままにせず, 教科書として明示知化し授業を改善する

授業計画（2021 に一部組み替え）

Week	~2020	2021~	Project
1	導入, プログラミング環境	←	
2	変数, 代入, 順次実行, 制御構造	変数, 代入, 順次実行, リスト	
3	タートルグラフィクスと関数	制御構造	
4	タートルグラフィクスと関数	力試し	演習
5	GUI とクラス	タートルグラフィクスと関数	タートル
6	GUI とクラス	GUI とクラス	
7	リスト	GUI とクラス	
8	ファイル	←	
9, 10	設計, 実装, テスト	←	
11	学術利用	←	
12	振り返り	←	自由課題
13, 14	自由課題	←	
15	まとめ	←	

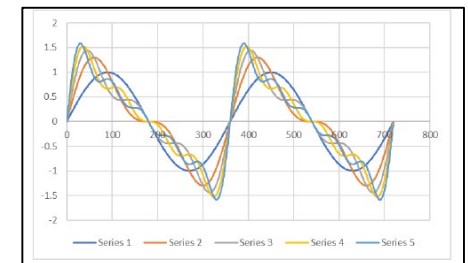
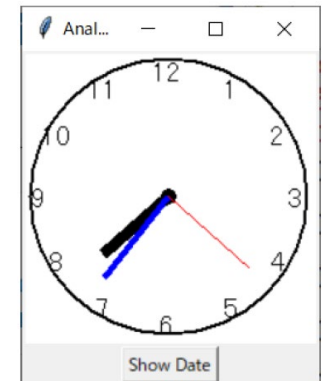
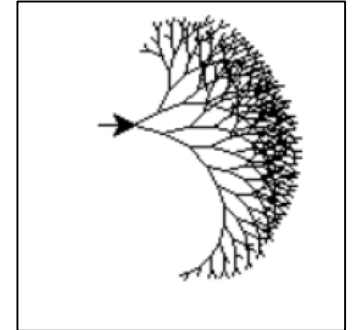
教科書開発と編集のポイント

教科書開発の意図

- 科目目標は Python 言語ではなく**プログラミング**
- 受講者の多くは文科系を含む学部1年生
 - プログラミング経験は乏しい.
 - 特定の専門性・応用を仮定できない.
 - 高校での知識と接続し興味を持てる内容にしたい.
- 大学設置基準の想定する**自学自習を支援**
 - プログラミング学習における躓きの低減.
 - 自由課題に取り組むための参照情報の提供.
- 市販のテキストは条件に合致しないので**独自開発**

初学者への配慮(1)

- Python に限定
 - 他の言語的要素を混在させない
- 授業での対応(暗黙値)の形式知化
 - 数学や英語などの既習の知識との接続
 - シンボルの読み方などコミュニケーションへの配慮
 - エラー対応の明示的経験
- グラフィカルな課題で動機付け, 学習を可視化
 - タートルグラフィクス, GUI アプリケーション, グラフ描画



初学者への配慮(2)

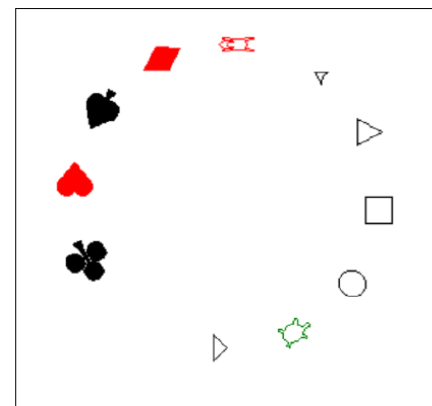
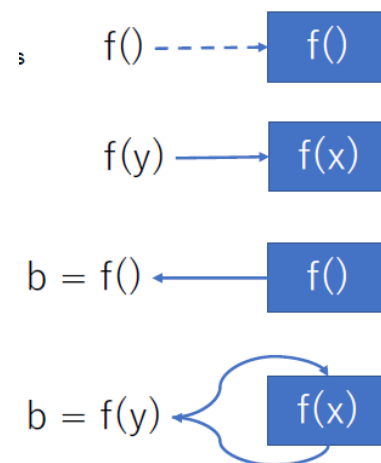
- 2重の for ループは初学者には難物
- 例題に馴染みのある文脈を与える
 - 京都の交差点を作る
 - トランプのデッキを作る

行	ソースコード
1	import random
2	suits = ["♠", "♥", "♦", "♣"]
3	ranks = ["A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"]
4	cards = [] #_Generate_a_card_with_a_suit_and_value #_and_append_to_cards
5	
6	
7	cards.append("Joker")
8	print(cards)
9	#_Shuffle_cards_and_assign_it_to_deck
10	deck =
11	print(deck)
12	

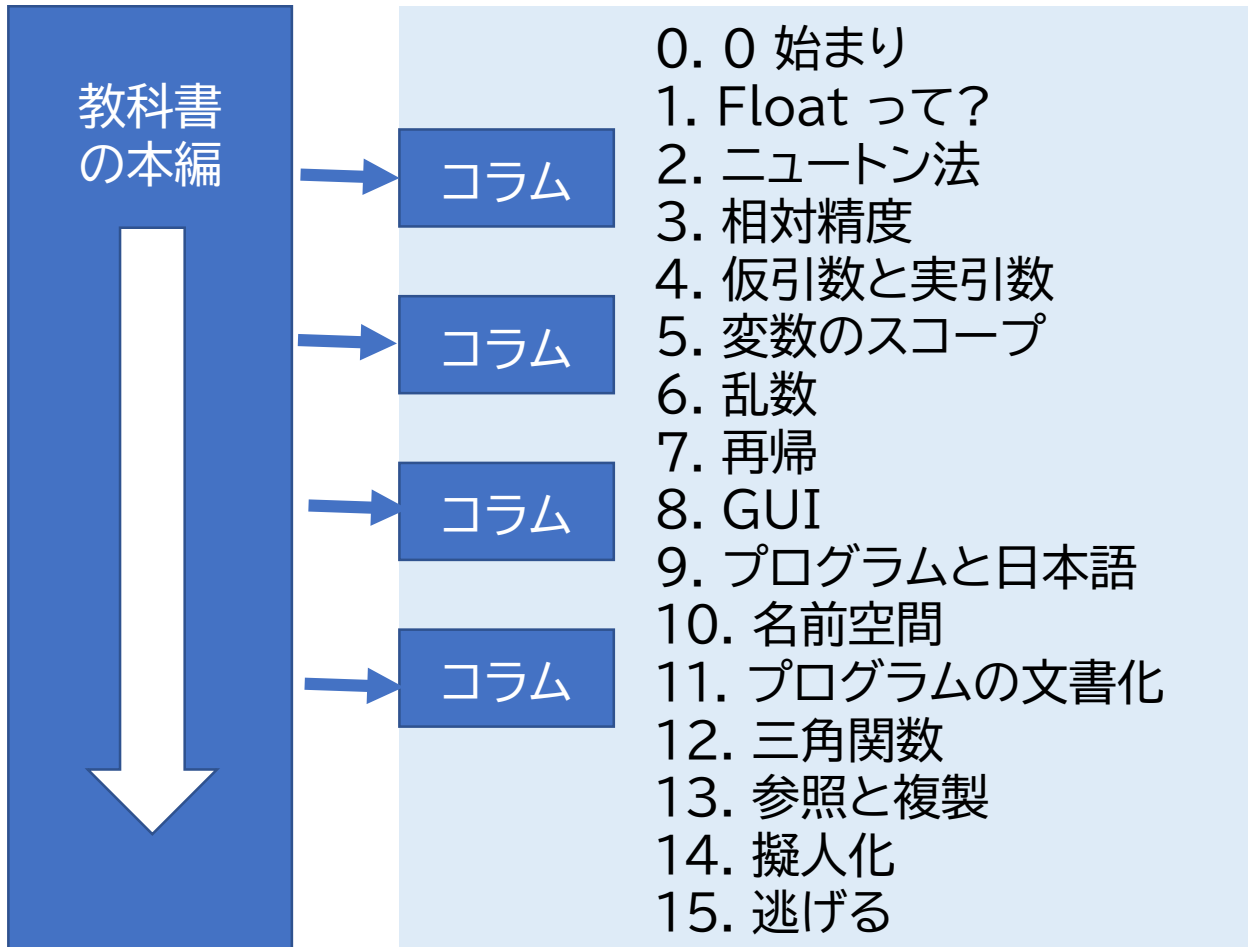
Row	Source code
1	tozai = ["Sanjyo", "Shijyo", "Gojyo"]
2	nanboku = ["Horikawa", "Karasuma", "Kawaramachi"]
3	for i in tozai:
4	for j in nanboku:
5	cross = i+j
6	print(cross)

初学者への配慮(3)

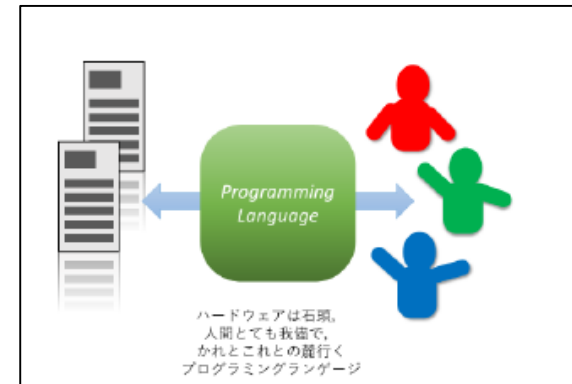
- 関数(+クラス):定義の仕方を教える前に使うことを先行させる.
- プログラム開発のプロセスの提示
 - 三目並べを例に設計, 実装, テスト
- 関連する話題はコラム編にまとめた
- 2021 年からは英訳も作成:留学生への配慮



コラム編としての関連する話題



プログラミングと計算の背景概念



組版とフォント開発

- 書籍形式 (PDF) の採用: 手離れがよい
 - 紙数, カラー利用から出版ではなく電子版の無償配布
- 組版: Word のスタイル, テンプレートを開発
- Python の特性を反映した2種類のコードの記述
 - シェルでの対話,
 - 完全なソースコードの記述
- 専用フォント K2PFE の開発
 - インデントの可視化, 全角半角の区別, PDFからのコピー&ペーストの支援

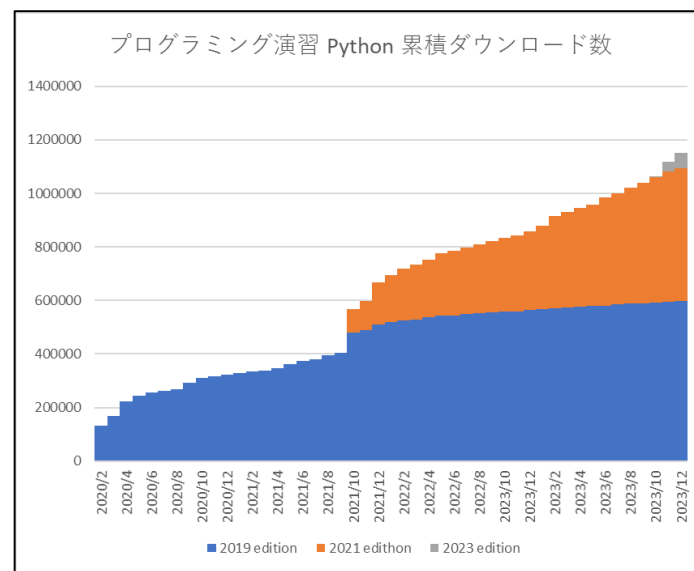
```
single entity. For example, inputting
    a = [5, 1, 3, 4]
and then
    print(a)
prints the entire list.
[5, 1, 3, 4]
```

Row	Source code	Notes
1	tozai=["Sanjyo", "Shijyo", "Gojyo"]	Only the text in red font uses full-width characters
2	nanboku=["Horikawa", "Karasuma", "Kawaramachi"]	
3	for i in tozai:	Concatenate strings using the + operation
4	for j in nanboku:	
5	cross=i+j	
6	print(cross)	

電子版での配布と公開

電子版での提供, ライセンス, 公開

- 教科書は電子版を学生に無償提供
 - 学生の費用負担への配慮
 - 履修できない学生への学習機会の提供
- 公開
 - 京都大学の学術情報リポジトリ KURENAI で公開
 - CC ライセンスの付与 (CC-BY-NC-ND)
著作権への心配を払拭
 - 累積ダウンロード数 110 万件以上 (~ 2024/1)



今後の課題

- 多様な Python の応用への接続
- 「プログラミング」, 「プログラムの動作」という動的なプロセスの学習支援
 - 教科書という静的メディアの限界
 - 授業では動画を併用
- プログラミングとその学習における生成 AI の活用