

プログラミング入門教育における Scratch から Python への接続教材

Connecting Scratch to Python in introductory programming education

山本 百華^{*1}, 中西 通雄^{*2}

追手門学院大学 経営学部 経営学科

^{*1} 20mm419@haruka.otemon.ac.jp, ^{*2} m-nakanishi@haruka.otemon.ac.jp

あらまし：高校でのプログラミング教育が必修化されている。初学者に対してはブロック型のプログラミングから導入すると良いと考えるので、本研究では、Scratch から Python へ接続する際の教育方法に着目して、90 分×1 回分の教材を作成した。プログラミング経験のない文系大学生を対象とし、毎回の学習状況の評価を踏まえて教材に改善を加えながら模擬授業を 3 回実施したので、その成果を報告する。

キーワード：プログラミング入門教育, Scratch, Python

1. はじめに

2022 年度から高校「情報 I」でプログラミング学習が必修化された。Python は最も多くの教科書で取り上げられており⁽¹⁾、Python を中心に学習させる高校が多いと考えられる。「情報 I」の学習指導要領や教科書におけるプログラミングに関する記述の量から、プログラミングに割り当てられる時間は、50 分×8 回程度と考えられる。プログラミングは、中学校の技術科でも学ぶことになっているが、すべての学校で十分に学習できているとは言えない⁽²⁾。従って、現在の高校でのプログラミングは、初学者を想定して教える必要がある。

2. 本研究の方針

本稿では、Python や C 言語のようにキーボードから文字をタイプ入力してプログラムを作成することを CUI 型と呼ぶ。また、Scratch などブロックを組み合わせた、wPEN⁽³⁾などで短冊を並べて、プログラムを作成することを GUI 型と呼ぶことにする。

現在の大学生で、入学前にプログラミングを学んだことのある学生は非常に少ない。著者の所属する学科では、1 年次に必修科目「プログラミング入門」がある。105 分×13 回の最初の 2 回で Scratch、次の 5 回で wPEN、残りの 6 回で Python が使われる。担当教員によると、最初から CUI 型にするとタイプミスをした際、英語でエラーメッセージが表示され、学生は意味が分からず、学習意欲を失ってしまうことから、この順序で教育しているとのことである。2023 年度の授業最終回でのアンケートの結果、過半数の学生が現行の順序を支持していた。

他大学でも、CUI 型の前に GUI 型で教えることで、受講学生の自己評価による理解度が高まったことが報告されている⁽⁴⁾。

GUI 型から始めることは、大学ではじめてプログ



図 1 模擬授業の流れ

ラミングの授業を受け、CUI 型から学習を始めた私の経験からも賛成である。GUI 型では、難しいルールや文法を知らなくても、直感的に操作することが可能である。加えて、プログラムの構造を視覚的に理解することができる。英単語や数字の羅列を相手にする前に、Scratch のブロックあるいは wPEN の短冊を用いたプログラミングを行うことは、高校生の心理的なハードルを下げるという点においても効果的だろう。以上のことから、高校でのプログラミング教育も GUI 型で始めるのがよいと考える。

Scratch や Python そのものの教育教材は数多くあるため、本研究では Scratch から Python への接続部分に注目して教材を作成した。

3. 模擬授業の概要

3.1 授業概要

模擬授業は 3 回実施し、各回 90 分で、図 1 のとおり 4 部構成で進めた。なお、第 2 回と第 3 回の教材は、前回の授業の反省をもとに修正して用いた。3 回の受講者は、全員プログラミング初学者の文系大学生であり、毎回の受講者はすべて異なる。

3.2 第 1 回・第 2 回模擬授業の内容

第 1 回授業の受講者は 3 名で、第 3 部で図 2 のようなプログラムを用いて演習問題を行った。着目すべきところは繰り返し回数のところだけであり、他は無視して良いことを説明したが、画面に表示されるコード量の多さに対して苦手意識がはたらき、敬遠してしまう受講者がいたため、第 2 回授業からはよりシンプルな教材を作成し、使用した。

```

1 """
2 kenryu_2_2
3 """
4 import tkinter
5 import time
6
7 def draw_fish():
8     global x, a
9     x = x + a
10    time.sleep(1)
11    canvas.delete("FISH")
12    canvas.create_image(x,350,image=gazou,tag="FISH")
13    canvas.update()
14
15 def swim_fish():
16     for i in range(10):
17         draw_fish()
18
19 root = tkinter.Tk()
20 root.title("泳ぐクレママス")
21 root.resizable(False, False)
22 canvas = tkinter.Canvas(root, width=1500,
23 height=1000)
24 canvas.pack()
25 gazou = tkinter.PhotoImage(file="fish_k.p")
26 canvas.create_image(200,350,image=gazou,t
27 x = 200
28 a = 100
29 swim_fish()
30 root.mainloop()

```



図 2 第 1 回授業の第 3 部で使用した教材

第2回授業の受講者は4名で、第2部で時間を使い過ぎてしまい、時間が足りなくなったため、第3回授業では、第1部と第2部の内容を修正した。

3.3 第3回模擬授業の内容詳細

第3回授業は、2024年1月25日に実施した。受講者は5名で、1名は3年生、4名は4年生である。

第2部でScratchを用いてプログラミングを始めた。第1・2回授業では、図3のイメージのように第2部と第3部の内容をリンクさせるようにし、第3回授業では、繰り返しを用いてネコを動かす演習問題のみ行った。「60歩進ませる」ところから始め、「60歩進ませるといふ動きを3つ並べて180歩の位置まで進ませる」、「ネコを三角形の形で進ませる」という順序で逐次処理の学習を進めた。

第3部では、冒頭に変数とprint文の説明をし、右のようなプログラムを完成させる演習問題を行った。次に、for文の説明を行い、「0~9の数字を表示させる」といった簡単なものから、turtleモジュールを使った「カメを三角形の形で進ませる」という図4のような演習問題を行った。

```
tate = 10
yoko = 20
menseki = ???
print(menseki)
```

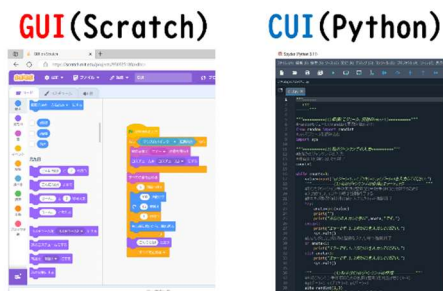


図3 GUI型とCUI型

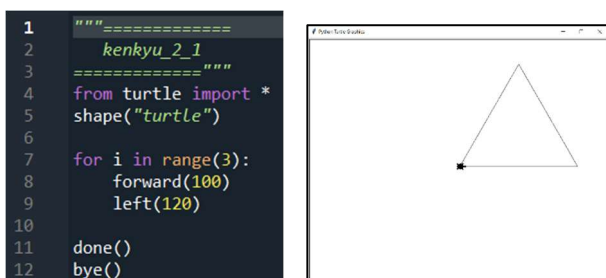


図4 turtleモジュールを用いた演習問題

4. 第3回模擬授業の結果と考察

第4部で実施した確認テストは全3問で、受講者5名全員が全問正解であったため、第3回の授業内容を全員が十分理解できたものと判断できる。確認テストの内容は次の通りである。

問1	Scratchで次のようにネコを動かしてください。【20歩ずつ後ろに下がるを5回繰り返す】
問2	Pythonでコンソール画面に0~10の数字が表示されるようなプログラムを組んで下さい。

問3	Pythonでカメが四角形の形に動くようなプログラムを組んで下さい。
----	------------------------------------

アンケートでは「Scratchを用いたプログラミングを経験しなくても、Pythonで行ったプログラミングの内容を理解することができたと思いますか?」と尋ね、「理解することができたと思う」「理解することができなかったと思う」のどちらかを選択してもらった。結果、受講者全員が「理解することができなかったと思う」と答えた。また、選択した理由についても尋ね、回答は以下の通りである。

- Pythonは文字・数字がほとんどで、所見では難しい印象を受けたが、Scratchはイラストや図形が使われていて親しみやすいデザインであったため。
- 図形でイメージがついて、そこから文字に起こしたため、頭で考えやすかった。

以上の結果から、Pythonでプログラミングを行う前にScratchを経験することで、プログラミングの感覚を掴むことができ、Pythonにスムーズに入っていることが分かった。しかし、今回は少人数での授業だが、高校では40人程度の生徒を対象とするので、高校の授業は生徒数が違うことから、同じような効果が得られるかは未知数である。

5. まとめ

高校でのプログラミング入門教育の分かりやすい教材作成を目的に研究を行った。ScratchからPythonへ接続する際の教育方法に着目し、改善を重ねながら、3回の模擬授業を実施した。第3回の授業結果から、Scratchを経験することでPythonの内容を理解してもらいやすくなるということが分かった。

今回行った模擬授業は、90分で完結させるために、Pythonの基本中の基本である変数やprint文から初学者にとっては難易度の高い反復処理までを詰め込む形で実施した。今後の課題として、接続部分の教材に条件分岐を入れること、受講者数を増やすこと、授業を複数回に分けて行うことが挙げられる。

[謝辞] 本研究の一部は、JSPS 科研費 JP19K12281, 21K02791 の助成を受けた。

参考文献

- 井手広康: “情報Iの教科書におけるプログラミング分野の比較と考察”, 情報処理学会論文誌, 教育とコンピュータ, Vol.8, No.3, pp.8-18 (2022)
- 大橋礼: “中学校のプログラミング授業の内容は?【中学生のプログラミング事情を徹底レポート】 | 教育トピック”, コエテコ, <https://coeteco.jp/articles/12141>, (2020.8.28), (2024.2.8 参照)
- 中西通雄ほか: “短冊形のコードによるプログラミング入門教育”, JSiSE 全国大会論文集, pp.48-49 (2020)
- 恐神正博ほか: “プログラミング学習におけるGUIベースからCUIベースへのスムーズな移行教育”, FIT 論文集, pp.479-480 (2023)