教育諸学研究　第35巻　4－25　2022

研究論文

小学校教員養成課程の大学生へのコーディング指導とその結果からのプログラミング教育カリキュラムについての考察

穴　田　恭　輔

1．はじめに

2020 年度より小学校でプログラミング教育が必修化され，各教科および総合的な学習の時間においてプログラミングの学習が行われている．しかし小学校でのプログラミング教育では，教師自身にプログラミングの経験がないとその指導についての不安を感じることも少なくない．小学校段階でのプログラミング教育の必修化が議論されていた頃に，黒田・森山 [1] が，2016 年 8 月～ 9 月に行った全国の小学校教員を対象にした調査（有効回答数 522）によると，その 92.0% がプログラミング教育に関する自己の知識・理解の不足に課題を感じ，84.1% が「モデル授業の実践事例集」の入手を希望しており，また，39.8% がプログラミング教育に関する自己の指導力を高める機会がないと回答している．それは小学校教員がプログラミング教育に関する教員研修などの機会を得ることを求めていて，ひいてはこれから小学校教員となる大学生にも，その養成課程のカリキュラムにおいてプログラミング教育が必要ということである．

穴田 [2] が K 女子大学の小学校教員養成課程に在籍する大学 4 年生（75 名）を対象にして行った 2020 年 10 月の調査では，小学校でプログラミングを教えることについてはほぼ全員が不安だと感じており，プログラミングを学びたいと思うかという問いについては，約 9 割が学びたいと回答している．その調査からも小学校でのプログラミング教育は指導者にとって不安を感じる対象となることが伺える．そこで，まずはプログラミング教育としてできることから始めるとすると，アンプラグドプログラミング [3] を行うこと，そしてその指導はプログラミングの 3 つの基本構造である「順次，分岐，反復」を意識しながら「プログラミング的思考」を育むことが肝要であるとした．しかし，その後システム環境も整えた上で児童に関心・意欲があれば，アンプラグドから発展して実際にコンピュータを用いたプログラミングへと展開していくことも十分可能であるので，そのためにも教員にはコーディングスキルが必要となってくることも示唆している．

そこで本研究では，小学校教員養成課程に在籍する数名の大学生に，プログラミング言語の指導を行い，その結果をもとに小学校教員養成課程におけるコーディングを含めたプログラミング教育カリキュラムの検討を行う．

2．本研究の目的とその方法

2.1．問題の設定

先に述べたK女子大学4年生（75名）を対象にした調査では，4割以上にプログラミング経験がまったくなく，経験がないことがプログラミング教育に対する不安の原因の一つになっていると考えられる．そこでこれからは小学校教員養成課程においては，プログラミングスキル習得のためのカリキュラムが必要であると考える．

そしてそのカリキュラムを作成するためには，小学校教員にとってどこまでのプログラミングスキルが必要なのかを明らかにしなければならない．一概にプログラミングスキルといっても，そこにコーディングスキルは含めるべきなのか，そして，もしコーディングスキルまでを必要と考えるならば，プログラミング言語のタイプ（種類）の検討も必要になってくる．

たとえば教育用のプログラミング環境として開発されたScratchなどのようなビジュアルプログラミング言語とCやPythonなどのようにプログラムをテキストで記述するテキスト型プログラミング言語がある．さらにビジュアルプログラミング言語にもブロックタイプやフロータイプがある．また，実行方法によって分類すると，Cのようなコンパイラ方式言語とPythonのようなインタプリタ方式言語に分かれる．このようにやや複雑に分類されることも，初心者に不安を抱かせ尻込みさせてしまう要因となっていると考えられるが，小学校教員養成課程の学生にとっては，どのような種類，タイプのプログラミング言語を習得するのがよいか，またどこまで習得するのがよいかを検討する必要がある．

2.2. 本研究の目的

小学校教員養成課程のためのプログラミング教育に関するカリキュラムを検討し作成する．

2.3. 本研究の方法

本研究では，一般的な私立女子大学の小学校教員養成課程の大学生を被験者としてプログラミング言語の習得を目指す．被験者はプログラミングの初学者で，その学習過程の記録をもとに小学校教員養成課程で学ぶべきプログラミングスキルの考察をする．

具体的には，自習と指導を受けることによってコーディングスキルを習得していく過程を被験者の情意面も含めて6か月間にわたり観察し，その成果から一般的な私立女子大学の小学校教員養成課程では，実際にどこまでが可能かを考察する．

とくに被験者には学習の記録をとってもらうことにした（学習日誌）．後日，その日誌を回収し，プログラミングの学習内容と日誌とを照らして，学習の進捗を評価する．

今回習得する言語はプログラミング言語としては初習言語となるので，今後いくつか他のプログラミング言語も扱うようになる可能性も考えればその基礎となる．プログラムのソースをテキストで記述するテキスト型プログラミング言語は，一般にソースが見やすく慣れればプログラミング構造をとらえやすくむしろ大学生には扱いやすいと考え，テキスト型であるPythonを選んだ．さらにPythonはインタプリタ方式言語なので，作成したソースプログラムをただちに実行できる．プログラミング作業にかかる負荷はコンパイラ方式言語と比較しても小さく，初学者にとっては学びやすく，それは

メリットだと考えられる.

3. プログラミングスキルの習得過程とその指導

3.1. 被験者および指導内容とその進め方等

【被験者】K 女子大学小学校教員養成課程の大学生（3 年生 X，Y，4 年生 Z）の 3 名
これらの学生は算数教育のゼミに所属している.
・3 年生の X と Y の 2 人は，プログラミングスキルの習得を主体的に希望している.
・4 年生の Z は，前年の 1 年間は算数教育に関する講読や演習を行っており，4 年生なので教員採用試験のための準備と卒業論文をメインに取り組む必要があり，それらと並行して以下に示すゼミには③から⑮まで参加したが, プログラミングスキルの習得については, リスト型あたりで学習を終えている.

【学習期間】2021 年 4 月～ 9 月

【レディネス】
被験者は 3 人とも共通して,
・PC の使用はレポート作成のために Office アプリを使用する程度で，プログラミングの経験はない.
・テキスト型プログラミング言語を習得していく過程で，必要なコマンドライン操作やエディタの使用方法，クリップボード，バッチ処理など，PC を扱うときに必要な操作が出てきたときはその都度教えていく必要があり，そしてそれらはいつも初めての体験だった.

これらは，K 女子大学の小学校教員養成課程に在籍する学生のプログラミングに関する一般的なレディネスであり，本研究でいう一般的な私立女子大学の小学校教員養成課程の大学生のレディネスもほぼ同じであると仮定する.

【指導内容とその進め方】
・プログラミング言語：Python 3.9
・テキストとして，邦題『退屈なことは Python にやらせよう』[4] を使用した．本書は 18 章で構成されているが，6 章を終えたところで，Python プログラミングの基本的概念はすべてカバーできており，その後もプログラムを書くための知識としては十分であるとの記述がある．本書の章立てにしたがい, 第 1 章から第 6 章までを輪読することにして, 週 1 回～ 2 回のゼミでは, 輪読の担当者がコードを実行しながら参加者に発表や説明する方式を取った．テキストの指導内容を表 1 に示す.
・指導者である筆者は，輪読担当者の発表を補足，修正しながらの指導を行う.
・プログラミングスキルの習得には自習が重要であることから，学習者は学習の記録（学習日誌）を付ける.

表 1　学習したテキストの構成内容

章	項目	主な内容
1 Python 入門	インタラクティブシェル	プログラミング環境づくり，数学演算子
	整数，浮動小数点数，文字列型	主なデータ型
	文字列の連結と複製	
	変数	代入文
	最初のプログラム	コメント，print()，input()，len()，str()，int()，float()
	ソースファイル，保存，実行	
2 フロー制御	ブール型	True，False，比較演算子，ブール演算子(and,or,not)
	フロー制御の構成要素	フローチャート，条件式，ブロック
	フロー制御文	if 文，else 文，elif 文，while ループ文，for ループ文，break 文，continue 文，range()
	モジュールのインポート	import 文，random.randint()
3 関数	関数の定義	def 文
	戻り値	return 文，None 値
	引数	
	ローカルスコープ，グローバルスコープ	ローカル変数，グローバル変数
		例外処理（try 節，except 節）
4 リスト	リスト型	インデックス，スライス，len()，del 文
	メソッド	index()，append()，remove()，sort()
	文字列とタプル	ミュータブル，イミュータブル，タプル型，list()，tuple()
	参照	参照，copy()，deepcopy()
5 辞書とデータ構造	辞書型	keys()，values()，items() メソッド
		get()，setdefault() メソッド
		整形表示
		データ構造
6 文字列操作	文字列	ダブルクォート，三連クォート，エスケープ文字，raw 文字列，in，not in 演算子，
	文字列のメソッド	
	変換	upper()，lower()，isupper()，islower()
	識別	isalpha()，isalnum()，isdecimal()，isspace()，istitle()
		startswith()，endswith()
	連結，分割	join()，split()
	揃える	rjust()，ljust()，center()
	空白文字を除去	strip()，rstrip()，lstrip()
	コピー & ペースト	pyperclip モジュール

3.2. 学習内容と情意の記録

2021 年 4 月から 9 月の 6 か月間の学習過程（日数）を整理すると，表 2 のようになる．これは，9 月中旬に提出された被験者 Y の学習日誌をベースにして作成している．各月をだいたい上旬，中旬，下旬に分けて，学習日誌が付けられた日を○で示し，ゼミで指導した日を丸付番号で示している．この期間中に学習日誌は 50 日分付けられ，ゼミは 20 回行われた．ゼミでの学習内容は概ねⅠからⅥの 6 期に分けられ，それぞれ，Ⅰ．入門期，Ⅱ．フロー制御，Ⅲ．関数，Ⅳ．リスト型，Ⅴ．辞書型，Ⅵ．文字列操作と正規表現となる．

表 2　Y の学習日誌（50 日）とゼミ（20 回）

月	4 月				5 月				6 月				7 月				8 月				9 月		
日	1	11	21	30	1	11	21	31	1	11	21	30	1	11	21	31	1	11	21	31	1	11	
Y の学習日誌		○	○	○	○	○	○		○	○	○		○		○		○	○	○		○		計
					○	○	○		○	○	○				○		○	○	○		○		
					○	○	○		○	○							○	○	○		○		
						○	○		○	○							○		○		○		
						○			○										○		○		
									○										○		○		
																					○		50
ゼミ		①	②	③	④	⑤	⑦		⑧	⑩	⑪		⑫	⑭	⑮		⑯	⑱	⑲				
						⑥			⑨				⑬				⑰		⑳				20
期	Ⅰ．入門		Ⅱ．フロー制御		Ⅲ．関数				Ⅳ．リスト型				Ⅴ．辞書型				Ⅵ．文字列操作と正規表現						

被験者たちは初めて学ぶプログラミング言語として Python を学習していくので，以下に示す記録には Python の仕様に特化した内容も一部含まれるが，大部分は一般にテキスト型でインタプリタ方式のプログラミング言語を学習するときに共通する内容である．そして，学習内容とそのときの Y の振り返りである【Y の日誌から】を併記する．学習日誌からの引用を「」付で示す．そこには，内容について疑問や納得したことが記載され，また情意面の記述も現れる．そして，本稿において筆者の見解を適宜付けている．なお，以下で例えば，(1) 2021/4/8　…① のような表記の意味を示すと，括弧付番号は Y の学習日誌の通し番号でその回を含めた学習回数でもあり，そこに日付を付けている．また，その日にゼミがあったなら，その後に付いた丸付数字はゼミの通し番号で，それはその回を含めたゼミの回数を示している．また，期間中は新型コロナウイルス禍にあり，大学の授業は対面期間と遠隔期間があったので，その日のゼミが対面であったか，遠隔であったかも示している．

I. 入門的な学習の期間（4/8 ～ 4/15）

(1)　2021/4/8　…①（対面）

Python プログラミングのテキストを使って講義を行った．実際にコードを入力して動作確認ができるようにあらかじめ Python のプログラミング環境をインストールしているノート PC を 1 人で 1

台を使えるように準備しておき，テキストの例を適宜，実行できるようにしていた.

【学習内容】

・インタラクティブシェル

IDLE は Python 用の統合開発環境であり, IDLE を実行するとプログラムを入力するインタラクティブシェルが起動する．>>> というプロンプトが表示され，2+2 のような値と演算子で構成される式を入力するとそれを評価して（式を計算して簡単にする）4 という値を表示する．Python はインタプリタでコードを処理・実行するので，被験者たちは数学演算子（**, *, /, +, - 等）を使って，いくつかの計算をしてみる．また，整数，浮動小数点数，文字列型を知り，文字列の連結と複製を行ってみる．被験者たちにとって初めてとなる Python プログラミング操作を行った.

・SyntaxError（シンタックスエラー）

Python の命令を間違って入力すると，SyntaxError（シンタックスエラー）が表示されるのを確認し，今後このエラーにたびたび見舞われることになるので，文法や構文のエラーであることを知らせておく.

・変数に値を格納する（代入文）

変数に値を保存するには，代入文を用いる．代入文は，変数と等号 ' = '（代入演算子）と保存する値から構成され，変数にはじめて値が入ることを初期化といい，その後別の新しい値が入ると上書きされる．数学で用いる等号とは違って，a = a + 1 のように変数を扱う操作である代入やその参照の結果は，プログラミングのおもしろさを最初に実感するところである.

【Y の日誌から】

「今まで無縁だったプログラミングを初めてした．実際に動かしてみると初めてのことだらけでとても楽しく，継続的に頑張って習得しようと思った．こちらが正しく命令を入力すると必ず正しく動いてくれるため，とても賢いし，便利なものだと感じた.」

この「とても賢い」という記述には，プログラムを初めて動かす体験で素直におもしろさを感じ，今後の学習につながる情意的な側面がよく表れている.

⑵ 2021/4/15 …②（対面）

【学習内容】

・最初のプログラム

ファイルエディタを使ってテキストにあるサンプル（名前と年齢を尋ねるプログラム）のソースコードを入力する．ファイルに保存し，実行する．組み込み関数 print(), input(), len(), int() が使われていて，ステップごとにその処理を確認した．また，コードを入力するさい，入力ミスで，SyntaxError にも悩まされた.

【Y の日誌から】

「初めてソースコードを見よう見まねで打った．最初見たときは暗号にしか見えなかったが，実際に実行してみると何が何を表しているのが理解できた．1文字でも誤って打つとエラーになるので，そこが大変で難しい点だと感じた.」

Ⅱ．フロー制御の学習に入る（4/22 ～ 5/13）

⑶ 2021/4/22 …③（遠隔）

この日からしばらく遠隔授業になって，被験者各自が自分の PC にプログラミング環境をつくって授業（ゼミ）に臨んだ.

【学習内容】

フローチャートとプログラミングの3つの基本構造である「順次，分岐，反復」を知る.

ブール型の2つの値（True，False）を知り，比較演算子（==，!=，<，>，<=，>=）で2つの値を比較してブール値が返されることを知る.

⑷ 2021/5/5

※ Y は，この日から自習をスタートした.

【学習内容】

・ブール演算子（and, or , not）と比較演算子を組み合わせて用いてブール値を確認する.
・フロー制御は，条件式が True，False によって何を実行するかを決定することを知る.
・フロー制御文（if 文，else 文，elif 文）
・while ループ，break 文

【Y の日誌から】

「今日は，授業以外で初めて Python にさわってみた．何度やり直しても，ずっとエラーが出て，画面が真っ赤になっていて，心が折れそうだった．テキストを眺めているだけでは全然コードの意味が分からないので，実際に動かしてみる必要があると強く感じた.」

⑸ 2021/5/6 …④（遠隔）

【学習内容】

・フロー制御文（if 文，else 文，elif 文）を中心に学ぶ.
・ゼミでは，いくつかの分岐があるとき，順次に処理が行われるプログラムにとって，その順番が変わると処理の結果が変わってしまうことを知る.

【Y の日誌から】

「elif 文（= else if ）は多数の節のうちからひとつを実行，elif 文は順番が重要！条件式が True に

なると，残りの elif 文はスキップされてしまう」

この記述から，順次，分岐について，そうか，なるほど．と自覚したようすが伺える．

また，「やはり事前に自分で一通りやってみてから授業を受けると理解が容易になるし，わからないことが明確になった．」とあり，事前に自習することが重要であることを自覚している記述がある．

⑹ 2021/5/10

【学習内容】

・if 文, while 文, continue, break を使ったサンプルプログラムを実行し，分岐，反復の制御を理解する．

・渡された値の型を変換する関数の int() の使い方を理解する．

【Y の日誌から】

「 age = input()

age = int(age)

int() 関数で，文字列から数値にすると，elif age < 12 が実行できる．不等号の左右を数値にそろえるためにはどうしたらよいかがなかなか分からなかった．とにかく色々試してみてたくさんエラーを出してみて，やっと int() 関数を使うところにたどり着いた．エラーばかりで画面が真っ赤になってくると嫌気がさすが，うまく実行できた時の達成感のためにもう少しだけ粘ってみようという気持ちが出てきた．」

この記述からは，上記 2 行のコードについて，ああそうかという納得した理解を得たことが読み取れる．つまり，ユーザから入力される年齢は文字列として変数 age に代入されるが，int() によって数値として変数 age に代入されなければ比較演算子 '<' が使えないことに気づき，それまでに何度も失敗を経てやっとそこにたどり着いたことが，この日の学習の最大の成果であったことが伺える．これは条件を評価するときに起こるエラーに対しての Tips の一つとして重要な事柄である．

⑺ 2021/5/12

【学習内容】

・この日で，自習でのフロー制御についての章を終えた．

【Y の日誌から】

「コードの内容を理解するスピードが少し早くなったと思う．この調子で頑張りたい．」

⑻ 2021/5/13 …⑤（遠隔）

【学習内容】

ゼミでは，反復（繰り返し）の制御について，

・while ループ，無限ループ，break 文

・for ループと range() 関数を使って，1 から 100 までの和

を扱い，とくに，ソースコードの処理をことばで説明するようにした．

【Y の日誌から】

「このコードがどんなことを示しているのかを一つ一つ丁寧に説明することはとても難しいだろうなと感じた．来週からしっかりと説明できるように事前に自分で何度も試しておこうと思う．」（亀のイラストつき）

ソースの処理をことばで説明することに難しさを感じているようすであるが，処理を論理的に組み立てるためにも，その理解をあいまいにしないためにも重要なことである．

Ⅲ．関数の学習に入る（5/16 ～ 6/4）

⑼ 2021/5/16

【学習内容】

・これまで print() 関数などの組み込み関数を使ってきたが，自分で関数を書く（def 文）．

・ローカルスコープ，グローバルスコープ，関数に渡すパラメータについて学ぶ．

・「グローバルスコープは一つだけ存在し，プログラムが実行されるとき生成される．プログラムが終了すると，グローバルスコープは消滅し，すべての変数は忘れ去られる．ローカルスコープは関数が呼び出されるたびに生成され，その関数の中で代入される変数はすべてローカルスコープ内に存在し，関数から返ると，ローカルスコープは消滅し，ローカル変数はすべて忘れ去られる．」[5] という初学者にとっては，その必要性の理解も含めて難しい概念である．

【Y の日誌から】

「ローカルスコープ，グローバルスコープ，パラメータなど知らない言葉がたくさん出てきた．説明の文章ですら，難しく理解するのに今まで以上に時間がかかった．」

⑽ 2021/5/19

【学習内容】

・再度，⑼と同じところを自習

・変数はローカル変数かグローバル変数のいずれかであり，ローカル変数はローカルスコープの中の変数で，グローバル変数は，すべてのスコープからアクセスできる変数である．これらの変数を使った例を学びながら，ローカル変数とグローバル変数について理解を深めていく．初学者には，自習だけでは難しい概念である．

【Y の日誌から】

「グローバル変数とローカル変数の区別がややこしく，なかなかコードの意味を理解できず，時間がかかった．」

⑾ 2021/5/20 …⑥（遠隔）

【学習内容】

ゼミでは，

・反復（繰り返し）を中心に，for 文，while 文をこの日の輪読担当 X が説明していく．
・Python の標準ライブラリにある関数を呼び出すためのモジュールをインポート（呼び出す）ための方法を学ぶ．ここでは，乱数関係の関数が含まれる random モジュール扱う．
・1 から 9 の整数乱数を生成して関数に渡し，その戻り値として，ランダムに 9 種類のメッセージを出力するサンプルプログラムのソースコードを解析する．

【Y の日誌から】

「import について自分で進めている時はイマイチ納得のいく理解ができなかったが，今日話を聞いてそういうことかと思うことができた．コードを分かりやすくするために，改行を心掛けようと思った．今までどうやってこのプログラミングを日常に取り入れていくのか，正直疑問に感じていた．しかし，今日 p.62 のコードで，これを使えばランダムに当選者を選ぶことができると知り，初めて実用的な使い方が少し見えた．このまま，もっと勉強して，ちゃんと実用的に使えるようになりたいと改めて感じた．」

これまでプログラムの実用性を実感しないままプログラミングを行っていたところに，はじめて具体的な実用例に出会い，学習のモチベーションを上げることができたようである．

⑿　2021/5/26

【学習内容】

・ローカル変数とグローバル変数のところを改めてもう一度復習した．
・0 で割るわり算で，例外処理の try 節，except 節の書き方を学ぶ．

【Y の日誌から】

「だいぶ慣れてきて，どのような内容のエラーがどのコードに対して出ているのかが分かってきた．そのため，エラーが出ることがあまり苦ではなくなった．解説書（テキスト）を読みながらどんどん進めていくのが楽しいし，わくわくする．全てが新しく知ることなので，いつも新鮮な気持ちで取り組むことができる．」

⒀　2021/5/27　…⑦（遠隔）

【学習内容】

ゼミでは，関数に入り，輪読担当が Y になる．

・None 値（戻り値なしを示す値）
・ローカルスコープ（ローカル変数），グローバルスコープ（グローバル変数）の正しい理解のためのサンプルコードを解析しながら，処理をことばで説明していく．

【Y の日誌から】

「今日は主にローカルスコープ，グローバルスコープについて確認した．自分で進めている時に曖

昧だったので（ゼミで）正しい理解ができてよかった．自分の認識が誤っていたこともあったので，1 人でするのではなく，誰かと一緒にやっていく大切さを改めて感じた．また，自分の言葉でコード等を説明することでその知識をより確かなものにできると思った．徐々に覚えたものが増えてきて，それらを組み合わせて使うことができるようになってきた．新しいアイテムをどんどん手に入れていく感じがしてとても楽しい.」

⑭ 2021/5/28

【学習内容】

・短いプログラム : 数当てゲーム（乱数で発生させた数を for 文による反復と条件分岐を使って 6 回以内に当てるゲーム）のサンプルコードを解析する.

【Y の日誌から】

「短いプログラムを見よう見まねで打って，数当てゲームを作ってみた．今まで暗号のように見えていたプログラムが，何を表しているのか読み取ることができるようになっていて，自分で驚いた．コードを書くのが楽しくなってきた.」

プログラミングの 3 つの基本構造である「順次, 分岐, 反復」を少しずつ使いこなせるようになりコーディングが楽しくなってきた時期である.

⑮ 2021/5/31

【学習内容】

・コラッツ数列を表示するプログラムを自力で作成する演習プロジェクト.

入力した正の整数の偶奇によって処理をした値（任意の正の数 n が偶数のとき n/2，奇数のとき 3n+1）を返す collatz() 関数を作成し，その関数を反復して呼び出し有限数列を出力するプログラムを作成する.

【Y の日誌から】

「エラー三昧で悲しかった. でも, 色々直していくと, エラーなく実行できて, とても嬉しかった. ずっとずっと考え込むより，どんどん変えて実行していく方が成功にはやくたどり着くような気がした.」

プログラミングの 3 つの基本構造である「順次，分岐，反復」に慣れてきたことで，とにかくコードを書いてみることが習得の近道だと感じているようすである.

⑯ 2021/6/2

【学習内容】

・⑮のコラッツ数列のプログラムに，ユーザが入力した文字列が整数でなかった場合の処理を try 〜 except 文で追加する．（入力の妥当性検証の処理）

【Y の日誌から】

「プログラムを書いて最終的にエラー無く実行することができたが，若干あやしい感じもあるので，

明日の授業で確認できたらと思う.」

⑰　2021/6/3　…⑧（遠隔）

【学習内容】

ゼミでは，⑮，⑯のコラッツ数列の動作確認をする．プログラムを完成するために，既習内容も復習しながらのコーディングとなった.

・例外処理（try ～ except 文）

・数当てゲーム

・関数を作ることの利点「複数回呼び出されるコードをまとめることができる．バグを見つけやすい．等」を考える．知る.

・インポートしたモジュール内の関数の呼び出し方は ' モジュール . 関数 ' ：例 spam.bacon()

・while 文，break

【Y の日誌から】

「コラッツ数列のプログラムにはやはり不備があった．1 を入力した場合，1 → 4 → 2 → 1 となる必要があった．コードを変更する過程で，while 文のみ使用すると無限ループになってしまい，止まらなくなった．そこでループを抜け出すように break 文を使うとうまくいった．1 つ何かが抜けると，全然動きが変わってしまうことを改めて実感した．でもそこが楽しいというか，面白いなと感じた.」

⑱　2021/6/4

【学習内容】

・コラッツ数列表示プログラムの入力の妥当性検証の処理を書き直した.

（つまり，try ～ except 文で，整数値以外の文字列が渡されたときのエラー処理を行う.）

【Y の日誌から】

「色々と試してみたが，結局できなかった.」

Ⅳ．リストの学習に入る（6/7 ～ 6/24）

⑲　2021/6/7

【学習内容】

・コラッツ数列表示プログラムの入力の妥当性検証の処理を完成した.

・自習でリスト型に入る.

【Y の日誌から】

「今回はまず前回完成させられなかったコードを完成させた．while 文を 2 回も使うということは頭になかった．なので，なるほど！と思った．4 章（リスト）に入った.」

プログラミングスキルを習得するために「なるほど！」という自覚的な理解はその学習過程におい

て重要である．自覚的に理解する対象はプログラムの構造であり，それこそがプログラミング教育のねらいとされている「プログラミング的思考」であろう．

⑳ 2021/6/9

【学習内容】

・リストを使う．for ループとリストを組み合わせてリストへの格納とリストからの取り出しをする．

【Y の日誌から】

「今，何に何が代入されているのか追うのが大変である．既習事項が容赦なくどんどん出てくるので，何だったっけとなるが，数をこなしてだいぶん何も調べずに分かるようになってきた．」

㉑ 2021/6/10 …⑨（遠隔）

【学習内容】

ゼミでも，リスト型に入り，リストに対しての操作を学んだ．

・リストの操作，インデックス，スライス（部分リストの取得），len()（長さの取得），del()，連結 等

・累算代入演算子 spam += 1 （spam = spam +1 と同じ）

【Y の日誌から】

「今日は，リストに授業で入った．覚えるというよりは何度も使っていくうちに身に付く感じだなと思った．今日は予想していたよりも進むスピードが速かったので，空いている時間にもっと頑張って進めようと思った．徐々にできることが増えてきて楽しい．タイピングを速くできるようになりたいなと思った．」

㉒ 2021/6/12

【学習内容】

・リスト型に備わっているメソッド index()，append()，insert()，remove()，sort()

【Y の日誌から】

「リスト名として使われている英単語の意味をヒントに考えればある程度理解することができると思った．」（※リスト名と記述されているのはメソッド名の間違いであろう．）

㉓ 2021/6/13

【学習内容】

・リスト型，タプル型，ミュータブル，イミュータブル，リストの参照，リストのコピー

リストを変数に代入することは，実際にはリストの参照を変数に代入していることになる．この参照については，自習では理解が困難なようすである．

【Y の日誌から】

「今やっている範囲の内容 * を蔑ろにしてしまうと後々ややこしいバグを招いてしまう．しっかり覚えていこうと思った.」

（注 * : リストを変数に代入することは参照を代入していること）

⑵4　2021/6/14

【学習内容】

・リスト型についての演習問題

・リストの各要素をカンマで区切り, 最後の要素との区切りは and にして文字列にするプログラム（カンマ区切り）の作成

【Y の日誌から】

「もっと，もっと，リストを使う問題が必要だと感じた．カンマ付けは結局できなかった．また再度挑戦しようと思う.」

⑵5　2021/6/17　…⑩（遠隔）

【学習内容】

ゼミでは，リストへの操作メソッドを学ぶ.

・メソッド，タプル，参照，ミュータブル，イミュータブル

リストはミュータブル（変更可能）で文字列はイミュータブル（変更不可）で，リストとほぼ同じデータ型のタプルがイミュータブルであることを知り，リスト型とタプル型の使い方を確認する．また，⑵3の自習では理解が困難であったリストの参照についての解説を聞く.

【Y の日誌から】

「ミュータブル，イミュータブルについては，曖昧な理解だったので，改めて説明を聞いて，なるほどなと思った.」

⑵6　2021/6/24　…⑪（対面）

【学習内容】

・リストや辞書を参照で渡すのではなく，複製をつくる copy() 関数，deepcopy() 関数を知る.

この日から，対面授業となる.

【Y の日誌から】

「今日は，久々に対面授業だった．やはり友達と相談しながら一緒に考えていくのがとても楽しいと感じた．完璧にコードを作るのはとても難しいので，順を追ってひとつずつ作っていき，最後に完璧なものを作ればよいと分かった．結局，授業内でコードを作り切ることはできなかった．しかし，家に帰ってもう一度やり直してみると一発でうまくいった.」

Ⅴ．辞書型のデータ構造の学習に入る（6/25 ～ 8/3）

⑵⑺ 2021/6/25

【学習内容】

・辞書型のデータ構造

【Y の日誌から】

「今日は 5 章に進み始めた．辞書が登場した．前の章で出てきたリストとよく似ていて紛らわしいと感じた．for ループや list() 関数など既習内容を使って進めていく必要があるので，その都度調べて確認し定着させていきたい.」

⑵⑻ 2021/7/1 …⑫（対面）

【学習内容】

ゼミでは，リストの演習として，

・⑵④のカンマ区切りの課題を扱った．被験者たちは，自習ではできているつもりのようだが，実はできていない．課題が求めているプログラムの仕様を解説し，1 つの文字列として変数に格納することとリストの要素数を固定しないことを要請した．時間内には解決できず，また後で考えるとのことで解決は保留している．

【Y の日誌から】

「完成したと思っていたので，全然できていなくて少しショックだった．何回も書き直してはエラーの繰り返しで大変だった．結局完璧なものに完成しなかった.」

このカンマ区切りの課題では，リストの要素数は任意でありそれに対応できる汎用的なプログラムにしなければならない．そのため単に固定して与えられたリストを処理するプログラムよりも難易度は上がる．プログラムの有用性を考えるなら，そのような仕様にすることとその必要性を理解することは避けられない．

2021/7/8 …⑬（対面）

【学習内容】辞書とデータ構造

2021/7/15 …⑭（対面）

【学習内容】辞書とデータ構造

ゼミの⑬，⑭では，辞書型のデータ構造を扱うが，Y の日誌に記述はなかった．プログラミング言語の習得にはデータ構造の考え方は重要であるが，このときはまだ，その重要性，有用性も十分に理解できていないようすで，日誌には記述がない．

⑵⑼ 2021/7/22 …⑮（対面）

【学習内容】

ゼミでは，リストとは違う辞書型のデータ構造を学ぶ．キー・バリュー・ペアをもつ構造である．

・辞書の操作をするメソッド keys()，values()，items()，get()，setdefault()，整形表示 pprint()，

pformat()

・辞書の要素となる組（キー・バリュー・ペア）には，リストのような順序はないこと．

【Y の日誌から】

「5 章に入った．（辞書は,）リストとよく似ているようで少しずつ違う部分があり，ややこしいと思った．テキストに載っているコードに少し付け加えて実行してみると，自分が予想した通りの動きをしたので，その瞬間，楽しいと感じた．テキストの範囲にとどまらず，どんどん自分が気になったことは試してみようと思う．今日は，タプルは必要？何のためにあるのという疑問が出てきた．今すぐに答えは見つからないと思うので，何度も使っていくうちに，必要性について気付くことができればいいなと思っている．」

⑶⓪　2021/7/25

【学習内容】

・辞書型を使って三目並べのデータ，ピクニックのデータ（参加者と持ち物）の格納の仕方を知る．

・三目並べのゲームの作り方を考える．

【Y の日誌から】

「三目並べのボードについてのコードを入力してみた．最初よくわからなかったけれど，pprint()で表示してみるとどんな構造になっているかが一目で分かった．ゲームとしては成り立っていなかった．私が書いたコードは 19 行で，完全なプログラムは 117 行あった．無理だと思った．」

「ピクニックの持ち物についてのコードを入力してみた．キーと値を関連付ける必要があったので（リストではなく,）辞書を使った．用途に合ったものを使っていくべきだということを改めて感じた．」これはデータ構造について考えることができたことを示す記述である．

⑶①　2021/8/2

【学習内容】

・辞書型の演習問題と演習プロジェクト．演習プロジェクトでは，ファンタジーゲームの持ち物リストを作成する．

【Y の日誌から】

「演習問題は比較的スムーズに出来た．その次の演習プロジェクトは難しくてできなかった．」

⑶②　2021/8/3　…⑯

【学習内容】

ゼミで，

・辞書を使って三目並べと演習プロジェクトのファンタジーゲームの持ち物リストを作成するソースコードを解析する．

・リストのデータ構造と辞書のデータ構造の違いについて確認し，リストは要素が順番に並んでいる，

辞書はキーと値が対応付けられているので順番はないことを理解する.

【Y の日誌から】

「演習プロジェクトでは，できたときは嬉しかったが，それと同時にやはりちゃんと考えながらやらないとなと感じた.」

これはプログラミングスキル習得について，⑮の日誌にある記述よりも進んだ見方であろう.

Ⅵ. 文字列を操作する学習に入る（8/5 ～ 8/26）

㉝ 2021/8/5

【学習内容】

・㉛で未完だった演習プロジェクト．持ち物リストから辞書に移す関数を作る.

・自習で文字列の章に入る.

【Y の日誌から】

「アイテム総数が 48 になるはずが，どうしても 46 になってしまう．分からなかった.

6 章文字列を進めた.」

㉞ 2021/8/6 …⑰（対面）

【学習内容】

ゼミでも 6 章の文字列の操作に入る．文字列操作は，今後，プログラムでファイル操作をするためには，その基礎となるスキルだと考えられる.

・文字列を操作するメソッド，upper()，lower()，isupper()，islower() 等.

【Y の日誌から】

「第 5 章の演習プロジェクト，昨日は全然できなかった．しかし今日自分が書いたコードを見てみると，[]（リスト）にすべきところを {}（辞書）にしていて，あっさりと完成した.

括弧ひとつでここまで変わるのかと痛感した．その後第 6 章（文字列操作）に進めた.

今日もモジュールのダウンロード方法が分からずこれ以上進めることはできなかった*．パスが通っていないらしい.」

（注 * : 6 章ではクリップボードを使うための「pyperclip モジュール」をインストールしなければならない．サードパーティ製モジュールのインストールには，コマンドプロンプトから，pip コマンドを使うという解説があって，それを実行しても失敗する．実は，各自のノート PC に Python プログラミング環境をインストールしたときに Path を通しておらず，コマンドがうまく動作せず，各自のプログラミング環境を再設定する必要があった.）

㉟ 2021/8/13

【学習内容】

・Python を再インストールする．インストール時には "Add Python to PATH" に☑を入れておく．
その後，pip コマンドで「pyperclip モジュール」をインストールした．

【Y の日誌から】

「スムーズにモジュールをインストールすることができた．試しにインタラクティブシェルに入力してみるとエラーも出ず，正常に動いたのでよかった．」

㊱ 2021/8/16

【学習内容】

・pyperclip モジュールを用いて，文字列のコピー＆ペーストをする．

【Y の日誌から】

「難しすぎて，あまり理解できなかった．クリップボードやバッチファイルなど知らない言葉が沢山あり，そもそも書いてあることはよくわからなかった．」

PC を使う基礎知識としてクリップボードやバッチファイルを知る必要があり，次のゼミで解説することになる．

㊲ 2021/8/17 …⑱（対面）

【学習内容】

前回㊱の自習では，この実行が何を意味するのか，何が起こるのかが分からず，この日のゼミで，（1コマを使って）解説が必要となった．

・（解説を受けて）クリップボード，Windows の環境変数，バッチファイルについて知る．

【Y の日誌から】

「分からなかった言葉の意味をある程度理解することができた．この状態でもう一度読めば書いてあることがわかりそう．急に難しくなってきた．」

分からなかったことも，解説のあとで復習したようで，ある程度理解できたことが記述されている．

㊳ 2021/8/22

【学習内容】

・前回㊲のところをもう 1 度読む．

【Y の日誌から】

「大体の内容は分かった．argv が何かよく分からなかった．実行しても何も起こらない．何ができるのか分からなかった．また日を改めて考えてみようと思う．」

コマンドラインで Python プログラムを実行するときに引数を指定する場合は，その扱い方を理解することは初学者だけの自習では難しい．ここでは，sys.argv がコマンドライン引数となるもののリストであり，argv[0] はプログラムのファイル名，argv[1] 以降がコマンドライン引数であると解説

する必要があった.

㊴ 2021/8/24 …⑲（対面）

【学習内容】

ゼミで,

・パスワードロッカー（パスワード管理プログラム）の作成とコマンドプロンプトを使う.

・プログラムの実行をバッチファイルで行う.

【Y の日誌から】

「今回はコマンドプロンプトを使った．まさかあの黒い背景に白字の画面を自分が使うとは少し前の自分では想像できなかった．6 章の演習プロジェクトもできた．7 章（正規表現）に進んでみた．6 章より理解しやすい内容だった.」

㊵ 2021/8/26 …⑳（対面）

【学習内容】

ゼミで,

・クリップボードからテキストを取得し，各行の先頭に * を追加した新しいテキストをクリップボードにコピーするプログラムのソースを解析し，実行する.

・テキストパターンを検索するとき，正規表現を用いない場合と用いた場合のプログラムソースの比較をして，正規表現の有用性を知る.

【Y の日誌から】

「正規表現は種類が沢山あってややこしいけれど，知っていたら便利だと思った.

今日で授業が一旦終わってしまったが，ここでやめてしまうと中途半端だしもったいないので少しずつコツコツ進めてみようと思う.」

夏の終わり 8 月末まで続いた Python プログラミングのゼミも被験者 X の教育実習がそろそろ始まるので，ここで一旦，区切りとした．Y の教育実習開始は X より 1 か月後なので，引き続き 10 回の自習を続け，9/10 までに，ファイル操作までの自習を終えている.

4．小学校教員養成課程向けのプログラミング教育への提案

以上のように実際に行った Python プログラミングスキル習得過程とその指導から，小学校教員養成課程の大学生向けのプログラミング教育カリキュラムについて考える.

4.1．内容の検討

大学生に必要なプログラミングスキルとして，(1) コーディングまで含める場合と (2) コーディングまでは求めない場合が考えられるが, それらの学習項目を表 3 に示し, (1) と (2) を比較してみる.

表 3　コーディング習得とプログラミング的思考に必要な項目の比較

No.	(1) コーディングまで含める場合			(2) コーディングまでは求めない場合
	コーディング習得に必要な項目			プログラミング的思考に必要な項目
			Python での例	
1	入出力		print(), input()	
2	データ型とデータ変換	数値		
		文字列		
3	演算子	数学演算子	+,-,*,/,%,**,//	
		比較演算子	==, !=,<,>,<=,>=	
		ブール演算子	and, or, not	
4	変数と代入文			
5	フロー制御文	フローチャート		フローチャート
				順次
		条件文	if, elif, else	分岐
		ループ文	for, while	反復
6	関数	組み込み関数		
		ライブラリ	import	
		定義関数	def	
		戻り値		
7	スコープ	ローカル		
		グローバル		
8	データ構造	リスト		
		辞書		
9	ファイル管理			
10	その他	シェル	IDLE	
		ファイルエディタ		
		クリップボード		
		バッチ処理		
		コマンドライン		

コーディングまでを含める場合は，今回実際に行った習得内容に加え，9 項目目（No.9）にファイル管理も含めている．ファイルを開いて，読み，書きをするのは，プログラムによる代表的な処理だからである．また，コーディングまでを求めない場合とは，2016 年（平成 28）年 6 月に出された有識者会議での「議論の取りまとめ」[6] にあるように，小学校におけるプログラミング教育については「プログラミング的思考」などを育むことが目的であり，コーディングを覚えることが目的ではないとされていることから，指導者となる教員にもそれを適用すると考えると，特定の言語を覚えるのではなく，プログラミングの 3 つの基本構造である「順次，分岐，反復」を組み立てることが学習の中心であり，表 3 の（2）の（No.5）に示すとおり，その流れを視覚化することができるフローチャートは学習項目になる．

しかし，令和 3 年 4 月から，全国のほとんどの義務教育段階の学校においては，児童生徒の「1 人 1 台端末」及び「高速大容量の通信環境」が整っているので，Scratch をはじめとするビジュアルコー

ディング言語や，インタプリタ方式言語を使ったプログラミング（コーディング）であれば，児童にとっても全く手が出せないということはなく，小学校教員はいつまでもアンプラグドだけにとどまってはいられなくなる．そこで何かプログラミング言語を習得するか，たとえ少しでもコーディングができることが望ましい．したがって，小学校教員養成課程の学生向けのこれからのプログラミング教育のカリキュラムを考えるならば，コーディングまでを含めたい．そして，前章に示した実際の指導過程（実践）からも興味をもって主体的に取り組むならば，小学校教員養成課程の大学生がコーディングスキルを付けることは十分できると考えられる．内容の提案として，表 3 の（1）の（No.1 ～ 10）と照らして，

Ⅰ．コンピュータ操作の基本として，（1）ファイルエディタによる編集，保存，検索 （2）クリップボード （3）コマンドラインからのファイル操作や Path 等の PC 環境 （4）バッチ処理

Ⅱ．プログラミング操作として，（1）プログラミング環境の構築 （2）フローチャート （3）入出力とプログラムの実行 （4）データ型とデータ変換 （5）演算子 （6）変数と代入文 （7）フロー制御文 （8）関数 （9）ローカルスコープ，グローバルスコープ （10）データ構造 （11）ファイル管理

などが考えられる．ただし，時間配分については，まだ検討の余地がある．

5．まとめと考察

本研究では，小学校教員養成課程で学ぶ大学生を対象に Python のコーディングスキル習得を目指してもらい，その結果から，小学校教員養成課程の大学生向けのプログラミング教育カリキュラムについて考えることができた．

理解が難しい内容として，とくに，

⑴ ローカル変数，グローバル変数

⑵ データ構造　リスト型，タプル型

⑶ リスト本体とリストの参照

があげられる．意外だったのは，被験者たちにリスト型がなかなかイメージできないことだった．現在の高等学校数学では行列を習わないことも関係しているのかもしれない．

また，今回の実践から，コーディングスキル習得過程において情意面の効果も大きいことがわかる．【Y の日誌から】に示されたように，Y は次々と出るエラーに悩まされ心が折れそうなときもあったが，概して，楽しい，面白いという記述が多かった．そしてうまくいったときの達成感のためにもと前向きになり，初めて実用的な使い方が見えたときにはさらに学習意欲が高まっている．また暗号のようにしか見えなかったソースコードが，学習回を重ねて行くと，何をしているか読み取れるようになっていて，自分でも驚いて，コードを書くのが楽しくなってきたと結んでいる．

ゼミの⑪から対面授業になって，友達（X）と相談しながら一緒に考えていくのがとても楽しいと，さらにモチベーションを上げている．これに対して Z は，並行して進めている教員採用試験の受験勉強と卒業論文の制作があり，純粋にプログラミングを楽しむ余裕はなかったようである．

最後に一般的な私立女子大学の小学校教員養成課程において，コーディングを含めたプログラミン

グ教育が可能かという問いには，今回の実践を通して，次の①，②の条件で可能であると考える．

①コーディングスキル習得には，主体的に取り組むこと．そして自習することに多くの時間を費やす必要がある．

②学習を引っ張ってくれる指導者がいた方がよい．その指導者から自習だけでは気づかない点を指摘されることが重要となる．例えば，３．２．節の㉘で指導者が汎用的なプログラムを書くように促したこともその一つである．しかし，もし指導者がいなくても，ともに学ぶ仲間がいた方がいい．
とくに効果的な学習法としては，他人に対して，ことばで正確にソースの流れを説明するのがよい．

引用・参考文献

[1] 黒田昌克・森山潤,「小学校段階におけるプログラミング教育の実践に向けた教員の課題意識と研修ニーズとの関連性」, 日本教育工学会論文誌, 41 巻 Suppl. 号 , pp. 169-172, 2018, 日本教育工学会.

[2] 穴田恭輔,「小学校教員養成課程の大学生にさせたプログラミング体験とその結果からのプログラミング教育についての考察」, 教育諸学研究, 第34巻, pp.5-23, 2021, 神戸女子大学文学部教育学科.

[3] コンピュータサイエンスアンプラグド, https://www.csunplugged.jp/, (最終閲覧日:2021 年10月31 日).

[4] Al Sweigart著・相川愛三訳,『退屈なことはPythonにやらせよう―ノンプログラマーにもできる自動化処理プログラミング―』, 2017, オライリー・ジャパン.

[5]『前掲書[4]』, p.65.

[6] 文部科学省,「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」, 2016, https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm, (最終閲覧日:2021年10月31日).