

Cの学習経験を持つ高校生へのPythonの授業導入事例

間辺 広樹¹ 長島 和平² 並木 美太郎² 長 慎也³ 兼宗 進⁴

受付日 xxxx年0月xx日, 採録日 xxxx年0月xx日

概要：高等学校情報科の共通必修科目が、従来の科目選択制から「情報 I」へと統合され、プログラミングが必修となった。文部科学省が新たに公開した教員研修用資料のサンプルプログラムには Python が使われていることなどの影響もあって、「情報 I」の教科書や授業では Python が採用される可能性が高い。一方で、高等学校における Python によるプログラミングの授業事例は少ない。今後、小学校や中学校にてプログラミングを学習してきた高校生が増えていくことを考え合わせると、様々な状況を想定した Python の授業実践を積み重ねていくことが必要である。筆者らは、C 言語の学習経験を有する高校 2 年生に対し、「C から Python へとプログラムを書き換える授業」と「関数概念を理解させる授業」という 2 通りの Python の導入授業を行った。本稿では両授業の評価と授業を通して得られた学習効果についての知見を報告する。

キーワード：プログラミング教育, 情報科学教育, 情報 I, Python, Bit Arrow

The example of class of Python to high school students who experienced studying C

HIROKI MANABE¹ KAZUHEI NAGASHIMA² MITAROU NAMIKI² SHINYA CHO³ SUSUMU KANEMUNE⁴

Received: xx xx, xxxx, Accepted: xx xx, xxxx

1. はじめに

高等学校の共通教科情報科(以下、「情報科」と記す)は、高度情報化社会に対応した人材を育成するために、情報の収集・分析から発信までを総合的に学ぶために、2003年に新設された教科である。当初は、情報教育の3本柱である「情報活用の実践力」を重視する科目「情報 A」、「情報の科学的な理解」を重視する科目「情報 B」、「情報社会に参画する態度」を重視する科目「情報 C」の3科目の中から、1科目を選択履修する形で始まった。その後の学習指導要領

改訂では内容が整理され、科目も「社会と情報」と「情報の科学」の2科目となった。学習内容としての「プログラミング」は当初よりすべての科目に収められていたわけではないため、一部の高校生が学習するに留まっていた。

ところが、2020年度より施行される新学習指導要領 [1] では、高等学校情報科の共通必修科目が、従来の科目選択制から「情報 I」の1科目へと統合されることとなった。学習内容の構成は以下の通りである。

- 内容(1) 情報社会の問題解決
- 内容(2) コミュニケーションと情報デザイン
- 内容(3) コンピュータとプログラミング
- 内容(4) 情報通信ネットワークとデータの利用

内容(3)に「コンピュータとプログラミング」が作られたため、すべての高校生がプログラミングを学ぶこととなったことが、これまでとは大きく異なる点である。

¹ 神奈川県立柏陽高等学校, Hakuyo High School, Yokohama, Kanagawa, 247-0004, Japan

² 東京農工大学, Tokyo University of Agriculture and Technology, Koganei, Tokyo, 184-8588, Japan

³ 明星大学, Meisei University, Hino, Tokyo, 191-8506, Japan

⁴ 大阪電気通信大学, Osaka Electro-Communication University, Neyagawa, Osaka, 572-8530, Japan

この改訂の背景には「情報の科学的な理解に関する指導が必ずしも十分ではない」ことや「情報やコンピュータに興味・関心を有する生徒の学習意欲に必ずしも応えられていない」といった課題があった。更に、「生徒の卒業後の進路等を問わず、情報の科学的な理解に裏打ちされた情報活用能力を育むことがいっそう重要となってきた」ことが改訂の理由としてあげられている。これらのことから、「情報 I」では、すべての高校生に対して従来の課題を解決するようなプログラミングの授業が求められていると言える。

また、小学校と中学校では高等学校に先立ってプログラミングが必修となるため、高等学校ではより高いレベルの学習内容も期待される。実際、学習指導要領「情報 I」の解説でも、「コンピュータでの情報の内部表現と計算に関する限界」や「関数の定義・使用によりプログラムの構造を整理する」など、これまでの学習指導要領にはなかった高度な内容が記載された。生徒もプログラミングの学習経験を持った状態で、高等学校に入学してくると考えられるため、新学習指導要領への移行を考えると、様々な状況を想定した授業実践を記録として残しておくことが必要である。

本研究ではプログラミング言語の Python に着目し、プログラミング経験を有する学習者への導入授業を検討する。Python とは豊富なライブラリを有したスクリプト言語であり、データサイエンスや AI などの分野で多くの実績を持っているため、今後の情報化社会を仕組みの面から理解するためには欠くことのできない重要技術の一つである。また、文部科学省が新たに公開した教員研修用資料 [2] のサンプルプログラムには Python が使われていたため、「情報 I」の教科書や授業では Python が採用される可能性も高い。

処理系を授業用コンピュータにインストールする作業などが必要であったことなどから、高等学校における Python によるプログラミングの授業事例は少ないが、オンラインプログラミング学習環境である Bit Arrow [3] で Python が利用できるようになり、Web ブラウザ上で学習できるようになった。

そこで、C 言語（以下、「C」と記す）の学習経験を持つ高校生に対して、「C から Python へのプログラムの書き換え（以下、「プログラムの書換」と記す）」と「Python による関数概念の導入（以下、「関数概念の導入」と記す）」という 2 通りの実験授業を行った。本稿では授業後に実施した生徒へのアンケート結果などから、導入授業としての両授業の評価を行うと共に、授業を通して得られた学習効果についての知見を報告する。

2. 研究の背景

2.1 オンラインプログラミング学習環境 Bit Arrow

プログラミングの授業においては、学習環境が授業実

	C	Python
課題 3	<pre>#include <stdio.h> void main(){ int i, sum=0; for(i=1;i<=100;i++){ sum=sum+i; } printf("%d",sum); }</pre>	<pre>sum=0 for i in range(101): sum=sum+i print(sum)</pre>

図 1 C と Python のプログラム比較

施や教育効果に影響するため、筆者らは Bit Arrow を用いてプログラミングの授業を実践してきた ([4], [5])。Bit Arrow は東京農工大学、大阪電気通信大学、明星大学で共同開発されているオンラインプログラミング学習環境である。以下に、授業を実践する立場から見た Bit Arrow を使うことのメリットを示す。

- Web ブラウザ上で実習ができるために、新たにソフトウェアをインストールする必要がない
- 家庭での学習を可能にする
- 生徒の学習履歴を残せる
- 異なる言語を同一のインターフェースで学べる

Bit Arrow ではこれまでは C 言語、JavaScript、ドットル、DNCL が使えたが、令和元年になって、Python も使えるようになった。

2.2 C と Python の比較

C と Python をプログラムの記述という点で比較すると、次のような違いがある。

- C は { } で反復や分岐のブロックを作るが、Python は「:」とインデントで作る。
- C は変数や配列の型宣言が必要だが、Python は不要である。
- C は命令の末尾に「;」が必要だが、Python は不要である。

これらの違いを、同一の処理をするプログラムを C と Python とで比べた (図 1)。Python の方がコードの記述量を少なく、記号や文字を入力することが苦手な生徒にも、学習のしやすさを提供する。また、インデントによるブロックの作り方も、C に比べてわかりやすく、ストレスを感じずに学べる可能性がある。従って、本研究ではそのような学び易さを前提とした授業が導入に相応しいと考えた。

3. 研究方法

3.1 対象となる生徒のプログラミング学習経験

本研究では高校 2 年生を対象として 2 通りの実験授業を計画した。同生徒らは高校 1 年生の時に、ドットル、Progress, C を用いた 8 時間程度のプログラミングの授業

表 1 生徒のプログラミング学習履歴

回	言語	内容・ねらい
1	ドリトル	プログラミングへの馴れを目的としたお絵描き
2	ドリトル	プログラミングの楽しさを味わうためのリズム作り
3	Progress	英語への馴れと反復の概念の理解
4	Progress	反復 for と分岐 if の活用
5	Progress	for と if の組合せ, for の入れ子
6	C	文字列 (Hello World) の表示, 奇数偶数判定 (入力と if)
7	C	1 から 100 までの総和, 変数のデータ交換, 配列の最大値
8	C	バブルソート, 素数判定

を受講している (表 1)。

主な授業の流れは、ドリトルを導入として使い、お絵描きや音楽作りで楽しみながら正確に記述することの大切さを学ばせる。その後、C への架け橋のために開発された Progress というソフトで、英語に馴れさせ、段階的に C の表記へと近付かせる [6]。最後に C で教科書のサンプルプログラム程度の問題を考えながら作成させる、という流れである。

この実践を通して、生徒はプログラミングを習得することの必要性や、できなかったことが出来るようになった喜びなどを認識したが、簡単なアルゴリズムであってもそれを自らプログラムできるほどの力を有するには至らなかった。従って、1 年生の時のアンケートでは授業が難しかった、と多くの生徒が回答していた。

そこで、Python の導入として妥当と考えた 2 通りの授業内容を示す。

- (1) プログラミング言語の違いを意識しながら、過去に C で作成したプログラムを Python に書き換える授業。
 - (2) 言語の違いへの意識は最小限に押さえて、新しい概念 (関数の概念) を理解させる授業
- その理由を次節に示す。

3.2 授業 A 「プログラムの書き換え」

過去に C で学んだプログラムを Python に書き換えるという作業を行う授業では、表記の違いを意識しながら、過去に学んだアルゴリズムを再度学習することになる。そのため、以下の学習効果が期待できる。

- 言語 (表記) は異なってもアルゴリズムは同一であることに気付く。
- 過去の学習では曖昧だったアルゴリズムが理解できるようになる。
- それぞれの言語の良さに気付く。

そこで、これらの効果を確認するために、1 年次に学んだ以下のプログラムを、本実験授業でも活用することとした。

- 課題 1 : Hello World の表示 (10 回表示させる)
- 課題 2 : 奇数偶数判定
- 課題 3 : 1 から 100 までの総和

C と Python プログラム比較

	C	Python
Lesson01 Hello World	<pre>#include <stdio.h> void main(){ int i; for(i=1;i<=10;i++){ printf("Hello World!\n"); } }</pre>	
Lesson02 奇偶判定	<pre>#include <stdio.h> void main(){ int hakuyo; scanf("%d",&hakuyo); if(hakuyo % 2 == 0){ printf("Even number"); }else{ printf("Odd number"); } }</pre>	
Lesson03 総和	<pre>#include <stdio.h> void main(){ int i, sum=0; for(i=1;i<=100;i++){ sum=sum+i; printf("Goukei= %d\n",sum); } }</pre>	

図 2 授業 A で配布・説明したプリント

- 課題 4 : データの交換
- 課題 5 : 最大値

授業では、C で書かれた学習済みのプログラムをプリントに記して配布し、C と Python とでどのように表記が異なるのかを教師が説明し (図 2)、生徒にもメモ書きさせる。その後、実際に Bit Arrow 上の Python を使い、プログラムを実行して確認しながら進めさせる。

説明を要する内容は以下の通りである。

- 課題 1 : print 文 (表示), for 文 (反復), ヘッダや変数の宣言は不要
- 課題 2 : if 文 (分岐)
- 課題 3 : (特になし)
- 課題 4 : (特になし)
- 課題 5 : リスト・配列の概念

説明を要することの多くは課題 1 に集約されているため、途中で教師からの説明はやめて、生徒自身でプログラムを作成するように指示する。20 分程度の作業の後、課題の出来具合などをアンケート調査する。

3.3 授業 B 「関数概念の理解」

Python のわかりやすさに着目し、生徒にとっては未知の概念でも、ストレスを感じずに学べる可能性がある。そこで、「関数」の基本を理解させる授業を計画した。ただし、未知とは言っても「2 次関数」など数学の授業を通して関数という言葉に馴染み、考え方を身に付けているため、以下の学習効果が期待できる。

- プログラミングにおける関数の概念 (定義の仕方とその呼び出し方) を理解する。
- Python の表記に馴染む。
- アルゴリズムを考えることに集中する。

実験授業で活用しようと考えた題材は以下のプログラム

	数学的な表現	Python での関数表現
課題 1	定義: $f(x)=3x+1$ $y = f(-1)$ $y = \underline{\hspace{2cm}}$	def <u> </u> : $y = \underline{\hspace{2cm}}$ return y print f(-1)

図 3 授業 B で配布したプリントの課題記述例

である。

- 課題 1 : 1 次関数
- 課題 2 : 2 次関数
- 課題 3 : Hello World の表示 (10 回表示させる)
- 課題 4 : 絶対値
- 課題 5 : Fizz Buzz

まず、あらかじめ作成したプリントを配布する。プリントには「数学的な関数の表記と Python の関数の表記の違い」と「それぞれのプログラムのヒント」を記しておく(図 3)。それを用いて、教師が説明し、生徒にもメモ書きをさせる。その後、実際に Bit Arrow 上の Python を使い、プログラムを実行して確認させながら、進めさせる。

説明が必要な内容は以下の通りである。

- 課題 1 : 関数定義の仕方, $f(x)$ で関数が呼び出され return で値を返す
- 課題 2 : 累乗の表し方
- 課題 3 : 数値だけでなく文字列も扱える
- 課題 4 : if 文と for 文
- 課題 5 : Fizz Buzz のルール

関数の定義の仕方など、説明を要することの多くは課題 1 に集約されている。そのため、途中の課題では、説明を最低限に留め、生徒自身でプログラムを作成するように指示する。20 分程度の作業の後、課題の出来具合などをアンケート調査する。

3.4 授業アンケート

2 通りの授業を実施した後でそれぞれに、以下の項目についてアンケート調査し、授業の評価を行う。

- (1) 授業は難しかったですか? (4 択)
- (2) 課題について、できたと思うものに を付けて下さい。
- (3) あなたがプログラミング言語を学ぶなら C と Python どちらがいいですか (5 択)?
- (4) 今日の授業の感想を書いて下さい (自由記述)。

4. 検証実験

検証実験は、4 クラス (155 名) を対象に実施した。4 クラスを 2 クラス (A グループ・78 名と B グループ・77 名) に分け、A グループでは「プログラムを書き換える授業」、B グループでは「関数の概念を理解させる授業」を行った(表 2)。

表 2 生徒のグループ分け

グループ	授業内容	生徒数
A	プログラムの書き換え	78 名
B	関数概念を理解	77 名

授業は A, B どちらのグループも 1 コマ (65 分) を用いた。それぞれにプリント教材を用意し、課題 1 から課題 5 までを教員がヒントを出しながら、生徒にプログラムを作成させていく流れを作った。

5. 結果

5.1 授業中の活動の様子

授業中は教師が生徒が作成したプログラムや友人との会話などを観察した。どの授業でも多くの生徒が隣の生徒と一緒に実行結果を確認したり、ミスした箇所を探すなどの共同で作業していた。

授業 A では、課題 2 や課題 3 においては、自主的に進めていく生徒と、進めることができない生徒に分かれた。課題を進められない生徒の多くは、以下のように誤った記述をしていたことを確認した。

- 「変数の宣言は必要がない」という教師の指示にも関わらず「int i」と記述した。
- 反復で「for i in range(10)」と表現すべきところを、「for(i=0;i<9;i++)」と C の記述をした。
- 記述不要な「;」を print 文などの後ろに記述した。

また、課題 2 や課題 3 を自主的に進めていた生徒であっても、課題 4 や課題 5 についてはアルゴリズムを記述する部分で躓いていた。ここはむしろ表記の違いだけを意識しながら「同じ式を書けばよい」のであるが、そのことに考えが及んでいない生徒が多かった。

授業 B では、課題 2 や課題 3 においては、ほとんどの生徒が自主的に進めていた。ただし、課題 4 では、「数学で学んだ絶対値の定義をプリントに書いてみよう」という教師の指示に対応できない複数の生徒が、プログラムを作ることでもできていなかった。課題 4 では、多くの生徒がエラーの出ないプログラムは作れていたが、「3 と 5 で割り切れる数の時は FizzBuzz と表示する」というルールを実現させることができず、条件の書き方などを友人と話し合っていたり、個人で試行錯誤していた。

5.2 質問 (1) の集計結果

A と B の授業を実施した後のアンケートの各質問に対する回答の集計結果を示す。まず、図 4 は、質問 (1) 「授業は難しかったですか?」の集計結果である。

授業 A 「プログラムの書換」は 33.3 % の生徒が「とても難しい」と答え、「少し難しい」(51.3 %) と合わせて 84 % の生徒が難しさを感じていた。一方で、授業 B 「関数概念の導入」では「とても難しい」は 5.2 % と少なく、「少し難しい」(58.4 %) と合わせて約 64 % であった。この結

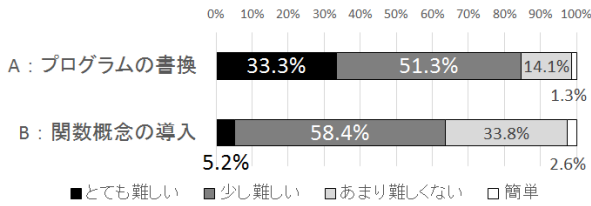


図4 授業別『授業の難しさ』の回答

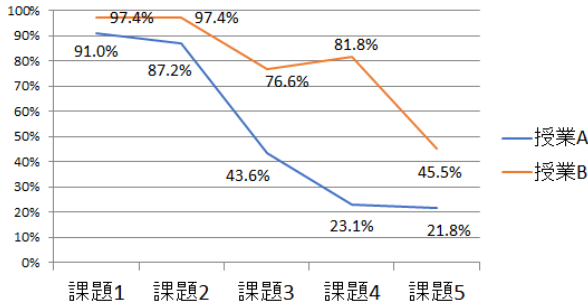


図5 課題ができたかどうかの自己評価

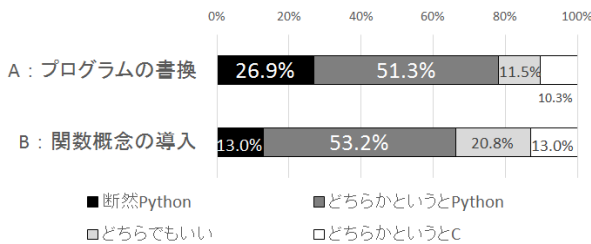


図6 授業別『どちらの言語で学びたいか』の比較

果から、授業 A「プログラムの書換」の方が、生徒にとって難しい授業内容だったと言える。

5.3 質問(2)の集計結果

質問(2)「課題について、説明を聞いて自分で作れたものに を付けて下さい」の回答の集計結果を図5に示す。

それぞれの課題が自分でできたかどうかを評価させたところ、どちらも授業後半になるにつれて正答率が下がっていった。特に、授業 A では、課題3以降の正答率が半数以下に減少した。

5.4 質問(3)の集計結果

質問(3)「あなたがプログラミング言語を学ぶなら C と Python どちらがいいですか?」の集計結果を図6に示す。

授業 A「プログラムの書換」と授業 B「関数概念の導入」共に Python で学びたいという生徒が多く、「入力の数値が少ないから」という理由をあげた生徒が多かった。一方で、1割程度の生徒が「どちらかという C」と答えていて、「Cの方が使い方がわかりやすい」や「Cの方が慣れているから」と理由をあげていた。

5.5 質問(4)の集計結果

質問(3)「今日の授業の感想を書いて下さい。」回答は自由記述であったが、同様の主旨のコメントが多数あったため、それらを以下のように7分類を用意し、それぞれ何名が記述したのかを数えた。

- 分類1「難しさ」
「難しくて全然わからなかった」のように、内容や作業の難しさが記述されていたものは「難しさ」として数えた。
- 分類2「簡潔さ」
「PythonはC言語と比べて変数の宣言もなく、簡略化していて入力が楽だった」のように、作業の簡潔さや楽かどうかが記述されていたものは「簡潔さ」として数えた。
- 分類3「楽しさ」
「Pythonの方が簡単だが、C言語の方がより本格的なプログラミングのように感じて楽しかった」のように、楽しさに関する内容が記述されていたものは「楽しさ」として数えた。
- 分類4「内容の理解」
「去年、C言語でプログラミングやその考え方を勉強していたから今回の授業では早く理解できた」のように、学習内容を理解できたと記述されていたものは「内容の理解」として数えた。
- 分類5「2言語からの学び」
「PythonとC言語の違いを学ぶことで頭が整理されたのでよかった」のように、2言語を学んだことで何らかの気付きを得たと記述されていたものは「2言語からの学び」として数えた。
- 分類6「アルゴリズム同一への気付き」
「PythonとCは全く違う言語だと思っていたが、根幹となる部分は同じだと知って驚いた」のように、アルゴリズムが同一であると気付いたものは「アルゴリズム同一への気付き」として数えた。
- 分類7「混乱」
「Pythonは文が少ないけど、C言語に慣れていたので、かえってややこしいと感じた」のように、2言語を学んだことでかえってわかりにくくなったと記述されていたものは「混乱」として数えた。

集計結果を図7に示す。

授業「プログラムの書換」では「難しさ」「簡潔さ」「2言語からの学び」「アルゴリズム同一への気付き」「混乱」が多く、授業「関数概念の導入」では「楽しさ・面白さ」と「内容の理解」が多かった。

6. 考察

2通りの授業を実施した結果から、導入授業としての適性を評価する。

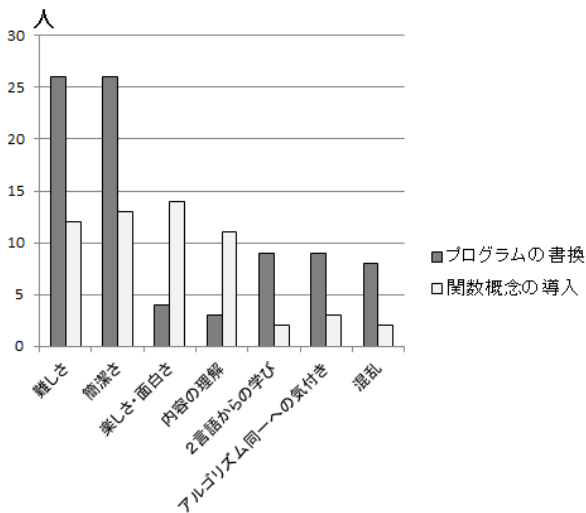


図7 授業別『コメント』の内容比較

6.1 授業A「プログラムの書換」について

授業Aでは、自主的に課題を進めることができない生徒が多かった。アンケート結果からも『授業の難しさ』について、「とても難しい(33.3%)」と「少し難しい(51.3%)」を選んでいて、合計84.6%もの生徒が難しさを感じていた。また、課題の正答率も全体的に低く、特に課題4「データの交換」と課題5「最大値を求める」で20%程度と低かった。

「既習のアルゴリズムをプリント教材を見ながらプログラミング言語の差異にのみ気を付けて作成すればよい」という課題であったため、当初はここまで生徒が難しさを感じる課題になるとは予想できしていなかったが、実際は、相当な難しさを感じさせる課題となった。

この要因として、以下が考えられる。

- 既習のプログラムであっても、個々の命令の意味や、各課題のアルゴリズムを十分に理解できていなかった。
- 表記の違いを埋めるための思考と、アルゴリズムを記述するための思考とが混在していたため、問題の切り分けができない生徒が混乱した。

このように一度学んだ内容であっても、生徒のアルゴリズムの理解は不十分であり、アルゴリズムを実装する本質的な箇所と、そうではない箇所との区別を十分に示さなければ、一部の生徒には難しい課題になってしまうことがわかった。また、C言語を学んだからこそ「混乱した」という主旨の記述があったように、プログラムを書き換えるという作業は、初めてプログラミングを学ぶ授業とは異なる難しさがあった。今後、小学校や中学校で様々なプログラミングの学習体験を経てきた学習者が高等学校にてプログラミングを学ぶ際には、このような点が課題となり得ることが示唆された。

一方で、Pythonの記述の簡潔さを知ったことで、80%近い生徒がプログラミングを学ぶなら「断然Pythonがいい(26.9%)」と「どちらかというPythonがいい(51.3%)」

と回答した。更に「PythonとC言語の違いを学ぶことで頭が整理されたのでよかった」や「PythonとCは全く違う言語だと思っていたが、根幹となる部分は同じだと知って驚いた」という記述にもあるように、プログラムを他の言語に書き換えるという授業には、「アルゴリズム同一への気付き」を促す効果が期待できる。そのため、そのような内容を学習目標とする授業においては、効果的であることがわかった。

6.2 授業B「関数概念の理解」について

授業Bでは、自主的に課題を進めている生徒が多かった。アンケート結果からも授業Aと比べると授業Bの方が「難しい」と感じた生徒は少なかった。課題の正答率も授業Aと比べると高くなっていて、導入授業として適していた。

これは、数学の授業で学んだことを題材に用いたことで生徒にとってはわかりやすい課題になっていたことや、関数を用いたことで、アルゴリズムを実装する本質的な箇所と、そうではない箇所との区別ができるようになっていたことなどが、要因と考えられる。プログラミングで使う関数の概念は、生徒にとっては初めての学習内容であったが、プログラミングの導入に適している可能性がある。この検証については今後の研究課題としたい。

7. まとめ

C言語の学習経験がある高校生に対して、2通りのPythonの導入授業を試みた。両授業に共通して「Pythonの方が良い」という生徒の声が圧倒的に大きく、導入授業に適していることがわかった。ただし、異なる言語の差を埋める授業は学習者に混乱を生じさせることもあるため、今後様々な言語を学んだ生徒が入学してくることを考えると、プログラミング教育での新たな課題になることもある。「情報I」の実施に向け、そのような課題も考慮したプログラミング教育のあり方を検討していきたい。

参考文献

- [1] 文部科学省: 学習指導要領解説情報編
- [2] 文部科学省: 高等学校情報科「情報I」教員研修用教材 http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm
- [3] オンラインプログラミング学習環境 Bit Arrow: <https://bitarrow.eplang.jp/>
- [4] 間辺広樹, 長島和平, 並木美太郎, 長慎也, 兼宗進: 情報教育における音楽の利用, 音楽教育における情報教育の利用, 高等学校における複数言語によるプログラミング教育の提案, 情報処理学会論文誌教育とコンピュータ(TCE), 3(3), pp.29-41 (2017).
- [5] 間辺広樹, 長島和平, 並木美太郎, 長慎也, 兼宗進: 自宅で行うオリジナル作品制作の学習効果と問題点~オンラインプログラミング学習環境を用いて~, 情報教育シンポジウム論文集, 2017(15), pp.101-109 (2017).

- [6] 大門巧, 間辺広樹; 教育用プログラミングソフト「Progress」を用いた授業における学習効果について研究報告コンピュータと教育 (CE), 2019-CE-148(9), pp.1-8 (2019).