

## 高等学校情報科におけるプログラミング教育の方法

—プログラミング言語 Python を用いて—

情報科 大井良知・堀江光代

### 【要旨】

令和 3 年度後期の「情報の科学」において、Python を使用したプログラミングの授業を行った。授業では、条件分岐や繰り返し処理などの必要最低限の文法を学習した後、応用的な実習に移った。応用的な実習では、既存のプログラムを改造する形で、生徒独自のコードを作り上げていくということをした。実践や試験の結果から、プログラミング教育の方法について課題はいくつかあったものの、新課程の情報Ⅰに向けての展望が見いだされた。

キーワード：情報科 プログラミング Python 新学習指導要領

### 1. はじめに

平成 30 年に告示された新しい高等学校学習指導要領（以下、「新学習指導要領」とする）の実施が令和 3 年度から始まった。新学習指導要領においては、様々な教科で科目構成の見直しが行われたが、その中で情報科に関しては、従来の科目「社会と情報」「情報の科学」を必修科目の「情報Ⅰ」と選択科目の「情報Ⅱ」に改められた。その背景には、今後の Society 5.0 の到来を迎えるのに、ICT に関わる知識や技能を有する人材を教育によって確保しておく必要があることにある。従来の「社会と情報」「情報の科学」においては、「情報の科学」にプログラミング学習の内容が含まれていたが、実際には約八割の学校において「社会と情報」が履修されており、そのことから高校卒業までにおいて、プログラミングについて知っている者は少数派であるという状況であった。

そこで、政策的に現在、プログラミング教育が初等・中等教育全体で推し進められている状況である。今回の学習指導要領改訂においては、小学校においてプログラミング教育が必修化され、中学校においてもそれを発展させる形でプログラミング教育が充実するよう、特に技術・家庭科の技術分野において改訂が加えられている。そして、高校において「情報Ⅰ」においてプログラミング教育が含まれており、これによって高校卒業までにおいて

全員がプログラミング教育を受けることになる仕組みになった。また、これに加えて、大学入試共通テストにおいても情報から問題が出題されることになり、そのサンプル問題中にも、プログラミングに関する出題が見られる。これらの流れからプログラミング教育については、全校種において、必須のものとなった。

本校舎では、従来から「情報の科学」を設置しており、その中でプログラミング教育を行ってきた。また、2020 年度からはプログラミング言語に Python を導入した。本年度の実践は、過去の授業を土台に、新学習指導要領においてプログラミング教育をどのように行っていくのが最良か、その方法を考えながら行ったものである。

### 2. 授業の概要

(1) 期間 令和 3 年度後期 (9 月～翌 3 月)

(2) 場所 CAV (コンピュータールーム)

(3) 対象 第 2 学年

(4) 単元目標

- ・プログラミング言語の文法を習得し、それに基づいてプログラミングができる
- ・プログラミングの仕組みを理解し、プログラム実行後の操作やエラー処理などが適切にできる。
- ・アルゴリズムの内容を理解し、プログラムの機能を変更する際にどの部分を書き換え

ればよいかを自ら判断できる。

#### (5)単元計画

- 1 時間目：入力と出力
- 2 時間目：文字列と数字のそれぞれの変換
- 3 時間目：条件分岐処理
- 4 時間目：繰り返し処理
- 5 時間目：関数
- 6 時間目：モジュールの活用
- 7 時間目：ピンポンゲームのアルゴリズム
- 8 時間目：ウィンドウ操作と画像の表示
- 9 時間目：プログラムの改造①
- 10 時間目：プログラムの改造②

### 3. 授業の内容、および方法

Python は、近年注目度が飛躍的に高まったプログラミング言語で、現在では機械学習など人工知能に関係するアルゴリズムの作成とも相性が良いとされ、IEEE(米国電気電子学会)の発表したプログラミング言語人気ランキングでは2017年以来、1位を維持している。また、新学習指導要領の情報Ⅰの教科書にもほとんどの教科書でプログラミングの分野において取り上げられている。それをふまえて、基本的な文法から授業で取り扱うことにした。なお、本授業は新学習指導要領ではなく、従来の「情報の科学」のもとで行ったものなので、Pythonの学習には、副教材(数研出版の『ミニマニュアルシリーズ プログラミング入門 Python 編』)とプリントを用いた。

1時間目から7時間目までの授業は、基本的に副教材に沿って進めた。また、各授業においては、副教材の内容をより詳しく説明したプリントを適宜用意した。授業は、基本的に最初の十数分でその時間に取り扱う内容について説明し、残りの時間を実習に割り当てるという構成で進めた。

このうち、3時間目の条件分岐と4時間目の繰り返し処理は、アルゴリズムにおいて特に重要となる部分なので、重点的に学習をした。また、Pythonには、既存のプログラムを現在作成中のプログラムに部品として組み込んで使用することができるモジュールというものがあるが、この使い方についても授業では重点的に取り扱った。

冬休み明けの8時間目から10時間目までの

授業は、授業者が作成したプリントを中心に実習中心の授業を行った。8時間目は、冬休み前の7回の授業の復習もかねて、そこから少し発展的な内容も含めて表示するウィンドウの大きさを変えたり、任意の画像ファイルを表示したりするプログラムの作成をした。なお、画像ファイルの形式はjpegとpngを主に使用した。9時間目と10時間目の授業においては既存の時間を読み替えるプログラムと、BMIを計算するプログラムをそれぞれ用意し、これまでに学んだ内容をふまえてオリジナルの要素のあるプログラムに改造させるという実習を行った。プログラムの改造においては、生徒の多くが、主に入力欄の位置を書き換えたり、画像データを任意で変え、それに応じて入力した数値などが表示される位置を書き換えたりといったような、コードの改変を行っていた。

### 4. 試験と評価

単元中の授業のほとんどにおいて、生徒には実習中に作成したプログラムをその時間の成果物としてオンラインで提出させた。そして、その提出物をもとに、平常点を評価した。

また、本校舎では例年、情報科は期末考査のみを実施している。そのことから、後期期末考査、つまり学年末考査において、プログラミングの内容について出題した。

出題内容は学習した内容すべてであるが、大きく2種類に分けて出題した。1つはPythonの基本的な知識や簡単なプログラムの読解に関する出題で、選択式(※資料1参照)、もう1つはプログラムの実行の結果や、改変によって発生するエラーなどに関する出題で、記述式(※資料2参照)である。

出題については、選択式の問題で基本的な知識やアルゴリズムの仕組みに関する理解を問う問題を、記述式の問題で基本的知識や仕組みへの理解をふまえ、より進んだ考察を試す問題をそれぞれ作成した。特に記述式においては、実社会において、プログラミングでの開発などの過程では、エラーが発生した際にその原因は何かを考え、修正するということが必要になるので、エラーの原因を考えさせるという形で、問題発見の力を試すところ

までを出題に含めた。

試験の結果は、学年平均が約67点で、クラスごとの差はあまり見られなかった。また、ほとんどの生徒についてバランスの良い正解率が見られ、上記のPythonの出題範囲についても、その正答率は、学年平均点とあまり変わらないものであった。そのことから、今回の単元を通してのPythonというプログラミング言語の理解は、良好なものと言える。

## 5. 課題と展望

今回の単元の実践において、課題として感じたのは、まずPythonの文法の多さからして、授業時間中にすべての内容を扱いきることが難しいという点である。本校舎では当該年度、WWLの共同実施校ということもあってカリキュラムの関係上、「情報の科学」に割り当てられた単位数が1であった。そのため、条件分岐や繰り返し処理といった特に重要度の高い部分を中心に教え、アルゴリズムを理解するために必要最低限の文法を扱った後すぐに、応用的な実習に移った。それでは、2単位であれば十分かという点、そうでもないだろう。プログラミングの理解度は生徒によって差が大きく、今回の単元の中では文法やアルゴリズムへの理解に時間がかかる生徒へのサポートも十分とは言えなかった。それらのことから、文法についてはどれぐらいのことを授業で取り扱うべきかを、今後は精査していく必要がある。また、プログラミングはオンラインで学習できるシステムやサービスが、各民間企業から提供されている。そのため、文法の学習については可能な範囲でそのようなシステムやサービスに委託する形で生徒が授業時間外で学習できるようにし、それらの復習と応用をまとめて行えるような授業内容、単元を考えていく必要があるだろう。

次に、プログラミングの授業においては、コンピューターを使うことが必須なので、そのコンピューターのスペックや起動状況によって生徒の学習進度に差が出てしまうという点も課題である。本校舎ではCAV内に設置しているデスクトップ型のコンピューターを用いて授業を行っているが、コンピューター

の故障や処理上の不具合による停止などによって、個々の生徒の実習が中断されてしまうことがしばしば起こった。このような生徒自身によらないトラブルによって学習進度に差が出てしまうのは問題であるし、何よりも生徒自身の学習意欲にも悪影響が出かねない。コンピューターそのものをはじめとして、授業中に問題なく使用できる設備の確保が必要である。

他にも、プログラミングで取り扱う内容によっては、他のライブラリやモジュールを使えるようにする必要があるが、そうするとそれらは外部からダウンロードなどして取り入れる必要があり、そこでセキュリティとの兼ね合いで問題が発生する点や、情報科における他の分野を取り扱う時間とのバランスをどうするかといった点も課題として残っている。これらの課題について、適切な対処がなされた授業設計を、今後考えていく必要があるだろう。

しかし、一方で、生徒の提出物や試験の結果から、Pythonによるプログラミング学習はそれほど困難なものではないということが明らかになった。また、生徒の多くが既存のプログラムの改造の実習において、それぞれ実際にユニークなものを作ることができていた点を考えると、文法やアルゴリズムを理解した上で、ゼロからコードを書くのではなく、既存のコードに手を加えて自分の作品をつくるという形の実習が有効であると考えられた。これを活用すると、今後の展望として、市販の様々なプログラミングの参考書にあるサンプルコードを土台として、より多様なプログラミングをさせることができるだろう。

## 6. おわりに

Society5.0の時代を迎えようとしている現在、プログラミングに関する知識やアルゴリズムのしくみに関する理解はもはや上級のスキルではなく、必須のリテラシーとなりつつある。それに対応するように、GIGAスクール構想や大学共通テストへの情報の追加など、教育における変化も大きなものとなっている。

しかし、これは表面的なものであって、根本的に学ぶべきものにそこまで大きな変化が



生じたものではない。このプログラミング教育も、情報科が生まれる前から、数学科において BASIC というプログラミング言語を用いて取り扱われていた。ただ、当時それは数学教育の中心にあるものではなく、選択的に興味のある者が学べば良い、という位置づけに過ぎないものであった。それが、時代が変わり、社会的にも必要なものになってきた。だからこそ、教科や科目の再編のなかで、数学科ではなく、情報科という新しい教科の中で取り扱うようになったということに過ぎない。言い換えると既にある学習分野で、いままであまり注目されなかったところに、スポットライトが当たるようになった、ということである。これが意味するものは、決して、奇をてらった新しい教育が求められているわけではない、ということである。

プログラミング教育において必要なことは、教科の垣根を越えて、そういった過去の教育内容をふまえた上で、そこに新しい要素を組み入れながら、社会が求める人材に必要な能力に適う力を授業の中で身に着けさせること

であろう。アルゴリズムの計算処理には数学的な知識が必要であるし、プログラミングは言語を用いるものでもあるので、国語や英語に結びつく要素もある。

また、プログラミング分野に限らず、情報化社会と言われる中で社会に関する教養もこの情報科では必要になるし、「データサイエンス」という分野を学習内容に含んでいることから理科と結びつくものもある。それらをふまえて、新学習指導要領における情報科教育は、なされていくべきものと言えるだろう。

## 参考文献

- ・文部科学省『高等学校学習指導要領』（平成30年3月公示）
- ・文部科学省『高等学校学習指導要領解説』（平成30年7月公示）
- ・田中賢一郎・澤田千代子『そろそろ常識？マンガでわかる「Python 機械学習」』C&R 研究所
- ・及川えり子『Python 超入門 モンティと学ぶはじめてのプログラミング』オーム社

## 資料1 学年末試験における Python の問題（選択式）

問1 以下の問いに答えなさい。

(1) Python というプログラミング言語の特徴を選びなさい。

①ブロックでプログラミングでき、直感的に分かりやすい。  
 ②日本語でプログラミングできる。  
 ③人工知能や Web アプリケーションに強く、少ない記述で実行できる。  
 ④機械で使う言語に近く、多くの言語の知識が学べる。

(2) Python のプログラミングをファイルとして保存した場合、拡張子はどれか。

① .jpg    ② .pdf    ③ .png    ④ .py

(3) Python でプログラムを記述する際の注意事項について、( ) にあてはまる語句を選びなさい。

プログラムの説明を書くコメントアウトは、(    )。

**A の選択肢**    ① // で始める    ② # で始める    ③ <!-- と --> で囲む    ④ /\* と \*/ で囲む

コロンを入力した後に Enter キーで改行すると、4 文字分下げられる。字下げは (    ) ともいう。Python では (    ) が非常に大きな役割を持っている。複数の文のまとまりを (    ) で表すので、(    ) を減らす際も注意する。

**I の選択肢**    ① インサート    ② インシデント    ③ インデント    ④ インポート

(4) プログラミングの基本的な用語に関して、適切な語句を選べ。

a データを一時的に保存しておく箱のようなもの  
 b サイコロの出目のように規則性がなく予測可能な数値のこと。ただし範囲を指定できる。  
 c 1 つの変数で複数の値をあつかうことができるもの。[ ] で囲む。  
 d プログラミングにおいて、よく使われる機能などをまとめたファイルのこと。  
 e アルゴリズムの基本構造のうち、条件に応じて実行する処理を分けるもの  
 f アルゴリズムの基本構造のうち、反復処理をさせるもの

a・b の選択肢	① 関数    ② 引数    ③ 変数    ④ 乱数
c の選択肢	① リスト    ② タプル    ③ 辞書    ④ 真偽値
d の選択肢	① コード    ② ライブラリ    ③ エディタ    ④ イテラティブオブジェクト
e・f の選択肢	① 順次    ② 条件分岐    ③ 繰り返し    ④ 代入

(5) 以下のプログラムに関して、空欄にあてはまる適切な数字をマークしなさい。

**A**

```

01 a = 1
02 b = 2
03 c = 3
04
05 c = a
06 a = b
07 b = c
08
09 print(a)
10 print(b)
11 print(c)

```

9 行目の print 命令で出力される値は (    A    ) である。  
 10 行目の print 命令で出力される値は (    イ    ) である。  
 11 行目の print 命令で出力される値は (    ウ    ) である。

**B**

```

01 a = 2
02 a += 1
03
04 print(a)

```

4 行目の print 命令で出力される値は (    エ    ) である。

**C**

```

01 def keisan(a):
02     b = a+3
03     return b
04
05 print(keisan(2))
06 print(keisan(4))

```

5 行目の print 命令で出力される値は (    オ    ) である。  
 6 行目の print 命令で出力される値は (    カキ    ) である。

**D**

```

01 def keisan(a,b):
02     c = a/b
03     return c
04
05 print(keisan(9,2))
06 print(keisan(7,3))

```

5 行目の print 命令で出力される値は (    ク    ) である。  
 6 行目の print 命令で出力される値は (    ケ    ) である。

問 2 以下の問いに答えなさい。

(1) 以下の空欄を補い、3 回勝負のじゃんけんゲームのプログラムを作りなさい。

01 import <input type="text"/>	乱数を使う宣言
02 win = 0	勝った回数を数える変数を初期化
03 draw = 0	あいこの回数を数える変数を初期化
04 <input type="text"/>	
05 <input type="text"/>	3 回繰り返す
06 print("■じゃんけん,i+1,"回目")	「■じゃんけん i+1 回目」と表示
07 print(">0:グー、1:チョキ、2:パー")	>0:グー、1:チョキ、2:パー 表示
08 com = <input type="text"/>	com=0 から 2 の乱数
09 you = <input type="text"/> input("あなたの手は?")	you=キーボードから入力した数字
10 print("コンピュータの手=",com)	コンピュータの手を表示
11 hantei=(com - you + 3) <input type="text"/>	判定=(com-you+3)を3で割った余り
12 <input type="text"/>	
13 <input type="text"/>	判定が画ならば
14 print(">あいこ")	>あいこ と表示
15 draw += 1	draw 加算
16 <input type="text"/>	そうでなく判定が画なら
17 print(">勝ち (^v^)"y")	>勝ち (^v^)"y" 表示
18 win += 1	win 加算
19 <input type="text"/>	そうでなければ
20 print(">負け (ToT)m")	>負け (ToT)m 表示
21 print("----")	「----」を表示
22	
23 print("結果=3 戦",win,"勝",draw,"引分")	最終成績を表示

※は値を隠している

あ の選択肢	① tkinter	② math	③ random
い の選択肢	① for i in range(3)	② for i in range(1,3)	③ while i <= 3
う の選択肢	① random.random()	② random.randint(0,2)	③ random.choice()
え の選択肢	① int	② float	③ str
お の選択肢	① /	② //	③ %
か・く・この選択肢	① else	② elif	③ if
き・けの選択肢	① hantei == 0	② hantei == 1	③ hantei == 2

(2) このプログラムを実行すると、以下のように出力される。空欄にあてはまる選択肢を選びなさい。同じ解答が複数回使われる可能性もある。

■じゃんけん 1 回目  
>0:グー、1:チョキ、2:パー  
あなたの手は?

この際、キーボードから「1」を入力し、コンピュータの手が2の場合、(き)と表示される。  
また、キーボードから「2」を入力し、コンピュータの手が0の場合、(し)と表示される。

さ・しの選択肢

① ">あいこ"      ② ">勝ち (^v^)"y"      ③ ">負け (ToT)m"

さらに、キーボードから「3」を入力すると(くす)処理される。

なお、キーボードから「A」を入力すると(せ)処理される。

す・せの選択肢

① キーボードから「0」を入力した場合と同じように  
② キーボードから「1」を入力した場合と同じように  
③ キーボードから「2」を入力した場合と同じように  
④ 再度、「あなたの手は?」と入力促すメッセージが表示されるように  
⑤ エラーメッセージが表示され、異常終了として

## 資料 2 学年末試験における Python の問題（記述式）

問 5. 次のコードを読み、以下の問いに答えよ。

```
1 import tkinter
2 root = tkinter.Tk()
3 root.title("ウィンドウと画像の表示")
4 root.minsize(640, 480)
5 canvas = tkinter.Canvas(bg="white", width=640, height=480)
6 canvas.place(x=0, y=0)
7 img = tkinter.PhotoImage(file="01.png")
8 canvas.create_image(320, 240, image=img)
9 root.mainloop()
```

※このプログラムファイルと同じディレクトリ(フォルダ)内には以下の画像データがある。

※著作権保護の観点より、絵は省略

リンゴの絵	オレンジの絵	モモの絵	レモンの絵	ブドウの絵
01.png	02.png	03.jpg	04.jpg	05.jpg

- このプログラムを実行した場合、表示されるウィンドウのサイズはいくらになるか。  
「横幅×縦幅」の形で答えよ。
- このプログラムをそのまま実行した時、表示される果物を答えよ。
- このプログラムを実行した時、ブドウの絵を表示させたい場合、どの行をどう書き換えれば良いか。  
書き換えるべき行を答えた上で、書き換えたコードをすべて記せ。
- このプログラムを実行した時に表示されるウィンドウのサイズをより大きくしたい場合、どの行をどう書き換えれば良いか。書き換えるべき行を答えた上で、書き換えたコードをすべて記せ。
- このプログラムの 4 行目から 8 行目を下のように書き換えたところ、エラーが発生して実行できなかった。何が原因だと考えられるか、簡潔に答えよ。

```
root.minsize(320, 240)
canvas = tkinter.Canvas(bg="white", width=320, height=240)
canvas.place(x=0, y=0)
img = tkinter.PhotoImage(file="03.png")
canvas.create_image(50, 50, image=img)
```