

プログラミング言語や実行環境によって 学習効果に差が生じるのか？

愛知県立小牧高等学校教諭 井手 広康

1. 情報Ⅰの4つの領域

高等学校では令和4年度より新学習指導要領が施行され、ついに「情報Ⅰ」がスタートを切った。「情報Ⅰ」は、「(1)情報社会の問題解決」、「(2)コミュニケーションと情報デザイン」、「(3)コンピュータとプログラミング」、「(4)情報通信ネットワークとデータの活用」の4つの領域から構成される。これら4つの領域の名称に含まれている「情報デザイン」、「プログラミング」、「データの活用」の単元が新たに「情報Ⅰ」で必修化され、従来の学習内容と比較して幅広い高度な内容が求められるようになった。加えて、情報活用能力が言語能力と同様に「学習の基盤となる資質・能力」と位置付けられたことや、令和7年度大学入学者選抜における大学入学共通テストへの「情報」の導入が正式に決定されることを受け、我々現場の教員は、これまで以上に緊張感をもってひとつひとつの授業に臨んでいるところである。

なお、高度な内容が求められているのは生徒だけでなく、教員にとっても同様である。先に挙げ

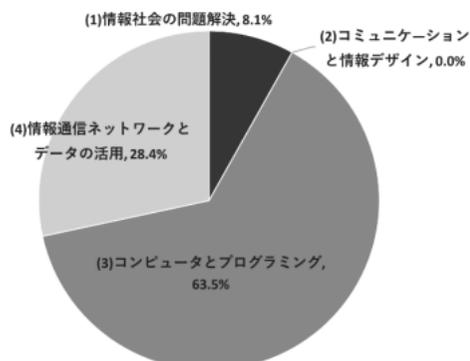


図1 「情報Ⅰ」において実施が最も不安な領域

げた「情報デザイン」、「プログラミング」、「データの活用」といった新しい単元は、情報科の教員であっても経験のない領域である場合も少なくない。愛知県高等学校情報教育研究会が令和4年4月に実施したアンケートによると、図1に示すように、「情報Ⅰ」において実施が最も不安な領域は「コンピュータとプログラミング」(63.5%)という回答であった(回答者数74名)。旧学習指導要領では、教科「情報」は「社会と情報」と「情報の科学」の選択必修科目で構成され、プログラミングの単元は「情報の科学」のみに含まれていた。「情報の科学」を設置している高等学校が全国で約2割であったことを鑑みると、「コンピュータとプログラミングの単元が最も不安」という結果は概ね想定通りであるといえる。

本稿では、筆者が過去に実施した2つのプログラミングの授業実践について紹介する。1つ目の授業実践は、プログラミング言語にPythonを使用し、クラスごとに実行環境を使い分けて授業を行ったものである。使用した実行環境によって、学習効果にどのような差が生じるのか検証した。2つ目の授業実践は、複数のクラスで4つのプログラミング言語(Python, VBA, JavaScript, Scratch)を用いて授業を行ったものである。使用したプログラミング言語によって、学習効果にどのような差が生じるのか検証した。以下、これら2つの授業実践について解説する。

2. 実行環境によって学習効果に 差が生じるのか？

1つ目の実践事例は、2019年度に実施したプロ

表1 プログラミングの授業内容

回	項目	授業内容
1	導入	Pythonの概要
2	変数	変数と代入
3	データ型	データ型とキャスト
4	条件分岐①	if文 (if-else)
5	条件分岐②	if文 (if-elif-else)
6	リスト	リストの操作
7	繰り返し①	for文 (range)
8	繰り返し②	for文 (リスト)
9	関数	関数の操作
10	総合演習	ユークリッドの互除法

表2 Pythonの実行環境

クラス	使用した実行環境
1組	IDLE
2組	Jupyter Lab (Notebook形式)
3組	Jupyter Lab (TextFile形式)
4組	Jupyter Notebook
5組	Google Colaboratory
6組	Spyder
7組	Visual Studio Code

プログラミング教育であり、本校1年生7クラス280名を対象としたものである。表1に示したようにプログラミング言語にはPythonを使用し、全10回をかけてプログラミングの基本的な事項について演習を行った。

第1回目では、Pythonの概要やプログラミング言語におけるPythonの位置付けについて説明した。第2回目から第9回目において、変数やリストなどプログラミングの基本概念に関する理解、条件分岐や繰り返しなど操作に関する演習を行った。第10回目には、総合演習として「ユークリッドの互除法」のプログラミングを行った。

これら全10回の授業を、クラスごとに実行環境を使い分けて実施した。ここで、表2にクラスごとに使用した実行環境を示す。IDLEはPythonと同時にインストールされる実行環境、Jupyter Lab, Jupyter Notebook, Spyder, Visual Studio CodeはAnacondaに標準装備されている実行環境である。なお、Jupyter LabはNotebook形式と

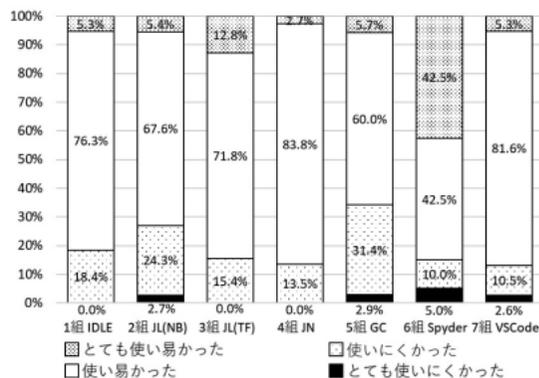


図2 質問「実行環境は使い易かったか？」の回答

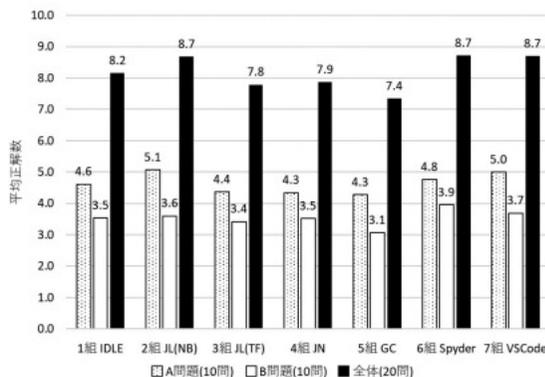


図3 確認テストの結果 (A問題/B問題/全体)

TextFile形式を使い分けて使用した。Google ColaboratoryはGoogleアカウントが必要となるが、ブラウザで動作するためインストールは不要である。

これらの実行環境を使用して授業を行い、最後に確認テストとアンケートを実施した。まず、「実行環境は使い易かったか？」という質問に対する回答結果を図2に示す。図2を見ると、回答に多少のバラツキは見られるが、どの実行環境も「とても使い易かった」と「使い易かった」という肯定的な意見が7～8割を占めていることがわかる。ただし、6組で使用したSpyderだけが「とても使い易かった」の割合が42.5%と非常に大きい結果を示した(カイ二乗検定の結果からもSpyderに有意差が見られた)。自由記述の回答からは「変数エクスプローラーが使い易かった」という記述が多く見られた。「変数エクスプローラー」はSpyder独自の機能であり、変数やリスト

表3 プログラミング言語と実行環境

組	人数	言語	実行環境
1	40	Python	Google Colaboratory
2	40		
3	40	JavaScript	Google Chrome デベロッパーツール
4	40		
5	39	VBA	Excel VBE
6	40		
7	40	Scratch	オンライン版 Scratch 3.0

表4 プログラミングの授業内容

回	授業内容	授業内容
1	変数	変数と代入
2	条件分岐	if文
3	繰り返し①	for文
4	繰り返し②	while文
5	配列／リスト	配列／リストの操作
6	関数	関数の操作
7	総合演習	数当てゲーム
8	確認テスト	サンプル問題（第2問）

の中身がリアルタイムで表に表示されるというものである。print関数を使用せずとも変数やリストの中身が変化の様子を確認できたことが、このような結果に影響したと考えられる。

次に、図3に確認テストの結果を示す。確認テストは全20問であり、A問題「プログラムの結果を記述する問題」の10問と、B問題「コードの一部を記述する問題」の10問から構成される。図3を見ると、A問題と比較してB問題の平均点が低いという傾向がいずれの実行環境においても表れている。A問題とB問題の難易度に差があったことも考えられるが、A問題のような演繹的にプログラムの結果を考察する問いよりも、B問題のような帰納的に結果からコードを推測させる問いの方が苦手傾向にあると推測できる。なお、カイ二乗検定の結果、実行環境ごとに点数の有意差は認められなかった。これらのことから、実行環境によって使い易いと感じることに多少の差が生じることはあるが、プログラミング問題の解答におい

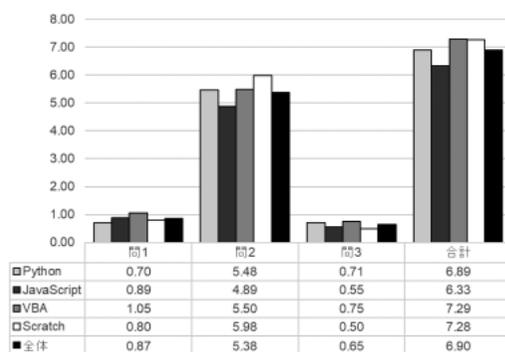


図4 サンプル問題（第2問）における平均点

ては大きな差は生じないと考える。

3. 言語によって学習効果に差が生じるのか？

2つ目の実践事例は、2021年度に実施したプログラミング教育であり、本校1年生7クラス279名を対象としたものである。この授業実践では、表3に示したように4つのプログラミング言語と実行環境を使用した。この実践事例で使用したPython, JavaScript, VBA, Scratchは、いずれも「情報I」の教科書に使用されているプログラミング言語である。ほとんどの高等学校において、これらの言語を使用してプログラミング教育を実施すると考えられる。これらのプログラミング言語と実行環境をクラスごとに使い分けて、表4に示したように全8回の授業を行った。

1回目から6回目までの6回で、変数、条件分岐、繰り返し、配列／リスト、関数について学習する。次の7回目に、これまでに学習した内容を踏まえた総合演習として「数当てゲーム」のプログラムを作成する。最後の8回目において、2021年3月に大学入試センターが公開した大学入学共通テスト「情報」サンプル問題の第2問「プログラミング問題」を解答し、その後、事後アンケートに回答する。

ここで、図4にサンプル問題の解答結果（平均点）を示す。なお、第2問は、問1～3の3つの大問に分けられ、それぞれ3個、10個、6個、計19個の問いから構成される。全体の平均点としては6.9点であったが、言語で比較しても大きな差が生じていないことがわかる。t検定の結果から

表5 4つのプログラミング言語を授業で使用する際の注意点

言語	注意点
 Python	<ul style="list-style-type: none"> ・プログラムのブロックを理解できていない生徒が多く、インデントの設定ミスによるエラーが多発した ・Python特有であるステートメント末尾の「:」を付け忘れてエラーになることが多かった ・繰り返しであるrange()を理解できていない生徒が多かった (例: range(1,4)は1から3までである)
 JavaScript	<ul style="list-style-type: none"> ・他のプログラミング言語と比較して全体的に入力文字数が多く、スペルミスによるエラーが多発した ・中括弧 { } によるブロックの概念を理解できておらず、対応する { } が書けてないエラーが多かった ・繰り返しであるfor()の表記を理解できていない生徒が多かった (例: for(n=1;n<5;n++)のような表記)
 VBA	<ul style="list-style-type: none"> ・配列の操作方法としてArray関数を使用したり、要素の指定に [] ではなく () を使用する点に注意する ・変数宣言Dimを教えるか否か、あるいは教えるタイミングに注意する (最初に詰め込みすぎない) ・変数の値をRangeやCellsでセルに代入するか、MsgBoxでメッセージとして出力するかどうか
 Scratch	<ul style="list-style-type: none"> ・他のプログラミング言語とは違いブロック型プログラミング言語であるため根本的に仕様が異なる ・プログラムの文法 (特に、リスト、比較演算子、条件分岐、繰り返しの表記) が大きく異なる ・数字の半角/全角の区別がほとんど付かず、全角が原因で上手く動作しないことが多発した

も、言語ごとで有意差は見られなかった。これらのことから、授業で使用するプログラミング言語によって、プログラミング問題の解答に大きな影響を及ぼすことはないと推測する。

4. プログラミング教育を実施するにあたって

本稿では2つの授業実践について紹介し、プログラミング教育で使用する言語や実行環境によって、学習効果に大きな差が生じないことを説明した。ただし、授業で使用するプログラミング言語によって、いくつか注意しなければならない点もある。ここで、2つ目の授業実践において、筆者が実際に4つのプログラミング言語を使用した所見として、注意して欲しい点を表5にまとめた。

なお、大学入学共通テストのプログラミング言語として、大学入試センター独自の疑似言語であるDNCLが用いられる。このDNCLの表記がPythonと酷似していることから、プログラミング言語にはPythonが良いという声をよく耳にする。しかし、実際に4つのプログラミング言語で授業を実践した感想として、いずれのプログラミング言語も良い面と悪い面があり、Pythonがこれら

の中で特段秀でていたとは感じなかった。プログラミング教育を実践する上で大切なことは、プログラミング言語でも実行環境でもなく、「プログラミングの考え方」を学び、それを実際の問題解決の場面で活かせる力を育むということである。選定したプログラミング言語の特性を生かした授業を展開することが、「情報I」のプログラミング教育で求められることだと感じる。

もう1つ、「入試対策のためのプログラミング教育 (情報教育) にはいけない」ということも意識しなければいけない。このことは、前文部科学省教科調査官の鹿野利春先生、現教科調査官の田崎丈晴先生、そして大学入試センター試験問題調査官の水野修治先生も繰り返し発信していることである。試験問題には必ず答えがあるが、今の子どもたちに必要な力は、「答えのない社会」を生き抜くための情報活用能力や問題発見・解決能力である。入試対策のためだけの何の面白味もない情報教育にだけにはしないよう、私たち現場の教員は、情報科の指導者としての責務を全うしなければいけないと強く感じる。