

総合学習・総合探究「プログラミング入門」実践報告

家本 繁 (数学科)

iemoto@tamacc.chuo-u.ac.jp

1 はじめに

今年度、新型コロナウイルス感染症 (COVID-19) によって学校教育のあり方が改めて問われる一年となった。この渦中に本校の総合学習・総合探究の一環として「プログラミング入門」を実施した。種々の制約があったため、今後に向けて課題も多く残ったがそれ以上に挑戦したことで得たことの方が多い。本稿は実践報告という位置づけであるが、特にプログラミング教育を実践するにあたって有益と思われる事項の共有に重点をおいて述べる。

2 教員に求められる資質・スキル

プログラミングに関する授業を立案する前提として、プログラミングの指導経験が必要なのかと問われれば、必要ないと言える。著者自身の専門は応用数学で、そのシミュレーションのために必要な最低限のプログラミング技術は有しているし、いくつかの言語を趣味の範囲で使うこともできる。しかし、計算機科学を専門とする研究者やエンジニアには速く及ばない。縁あってこれまで大学で C 言語の講義を担当してきたが、その初年度は指導経験を有していなかった。数学であれば問いに対する解法はせいぜい 2、3 個であるが、プログラミングにおいて学生が書くコードは一人一人全く異なると言っても過言ではない。そのため学生のエラーや癖を多く含むコードを読み、設計思想からエラーの所在までを瞬時に判断するのは難しく、それは経験を重ねてもやはり一定の時間はかかるものだとして現在は自分を納得させている。高等学校におけるプログラミング教育も成熟期に入れば話は別だが、少なくともこれから数年の黎明期においては玉石混濁が当然で「誰にでも初年度はある」と思い、立案して何ら問題ないと考えている。

一方で指導を考えているプログラミング言語については、所謂「入門」と名のつく市販の書籍を数冊は学ぶ必要がある。また、生徒の書いたコードが境界条件を含めて適切に動作するかの判断が求められる場合もあるため、様々なサンプルデータでコードの動作確認を自動で行うオンラ

インジャッジ機能のある AOJ (Aizu Online Judge)*¹や paiza ラーニングのスキルチェック*²等で問題を解きながらコーディングする経験を積んでおくとも良い。また、文部科学省による高等学校情報科「情報Ⅰ」「情報Ⅱ」の教員研修用教材*³や、有料となるがオンライン上でプログラミングの入門講座も多く提供されている。これらに取り組み、指導時のポイントやスライド作成の工夫等の参考にするのも良いだろう。

3 学習する言語の選定

ソフトウェアの品質評価を専門とするオランダの企業 TIOBE Software が発表している「TIOBE Index*⁴」はプログラミング言語の人気度合いを表す指標として知られている。2021年1月版での上位3つの言語は次のようになっていた。

1位	C
2位	Java
3位	Python

現在でも日本の多くの大学でCが指導されており、著者自身も学生時代CかJavaをきちんと学習しておけば、プログラミング言語の動作原理がよく分かり、他の言語の習得も容易になると指導された。実際この認識に間違いはないが、CやJavaを「きちんと学習」するためには、相当数の授業時間が必要となる。プログラミング言語はマイナーなものも含めると200以上あることを考えると、まずTIOBE Indexのような指標から現役プログラマーの多くが使っている言語を絞り込み、その上で学習目標を決め、その目標達成に最も効果的な言語を選ぶことを勧める。例えば、数学の統計分野の延長としてより身近なデータに対する分析を目標とするのであれば表計算ソフトの関数の扱いに留まっても良いだろうし、Rのような統計解析用の言語を選択することもあり得る。また、G Suite for Education(以降G Suite)を導入している学校であれば、フォームやスプレッドシート、ドキュメント等をより柔軟に接続させるための言語としてGAS(Google Apps Script)に取り組むのも面白いかもしれない。GASはJavaScriptがベースとなっている言語で、JavaScriptはWebアプリケーションの開発にも使われている人気の言語だ。生徒が生徒会活動や文化活動を主体的に進める上で不便に感じる事柄に目を向け、GASを用いてその解決や効率化を図ることができたらPBL(Project Based Learning)型の学習としても興味深い取り組みとなるだろう。

とはいえ近年科学技術の分野で特に注目されているものの1つは、やはりAIや機械学習関連

*¹ <http://judge.u-aizu.ac.jp/onlinejudge/>

*² <https://paiza.jp/challenges>

*³ https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm

*⁴ <https://www.tiobe.com/tiobe-index/>

の技術であろう。これは本実践の受講生がプログラミングに興味をもった主な理由でもあった。2000年頃に第3次人工知能ブームが起こり、現在はその基本技術を様々な分野に応用した製品やアプリケーションが盛んに研究・開発されている。このAI関連技術に最も親和性が高い言語がTIOBE Indexで3位にランクインしたPythonだ。Pythonはscikit-learnやtensorflow等の機械学習関連のパッケージが整備され、文法も平易で例えばCやJavaのような変数や関数の定義における型付けが不要だ。またコードを実行する際にコンパイルが不要なインタプリタ言語でもある。そのため、AI関連の技術を扱ったり論理的思考力の養成を目標とするのであれば、これらに関する書籍や情報が豊富にあるPythonを利用すると良いだろう。ただし、Pythonのような柔軟性に富む言語を扱う場合、コンピューターの内部処理に対して生徒の注意が向きにくいいため、必要に応じて発展的内容として考える機会を設けても良いだろう。

4 学習環境の選定

以上の検討を経て学習に使用する言語が決まったら、次に実習に用いる統合開発環境 (IDE) を選定し、ローカル、あるいはクラウド上のいずれかで用意しなければならない。その際、次の観点を中心に使用するIDEを検討すると良いだろう。

- 基本的にはIDEはローカル上ではなく、クラウド上に用意されているものを使用する。学校の教育用端末にIDEがインストールされている場合もあるだろうが、自宅でも授業と同じ環境を構築可能かどうかを検討すべきである。
- 言語の基本文法が生徒に定着するのに時間がかかる。そのため使用するIDEがコマンドの補完機能を備えていることが望ましい。
- 使用するIDEの操作が直感的で使いやすいか。
- 使用するIDEに関する情報がネット上に豊富に存在するか。

本講座ではPythonを学習する言語に定めたため、クラウド型IDEとしてGoogle Colaboratory (以降Colab)を採用した。ColabはGoogleのアカウントがあれば使用可能なクラウド型IDEである。その他にもPythonが使えるIDEとしてCloud9, repl.it等があるが、次のような観点からColabを用いることにした。

- (1) 使用に際して新たにアカウント作成が不要 (本校はG Suiteを導入済み)
- (2) GPUなどのスペックが高く、本格的な開発にも耐えうる環境
- (3) 機械学習に関するデモが豊富に用意
- (4) コードの補完機能、スニペットの充実

ただし、ColabはPythonしか使用できないことに注意が必要だ。Cloud9やrepl.itはPython

以外の言語もサポートしている。

5 本校における実践概要

ここでは、本校の総合学習・総合探究の一環として今年度行なった「プログラミング入門」について、その概要をまとめる。

5.1 対象・人数

対象は高校1～3年で募集人数は15名程度とした。大学では4,50名のクラスを教員1名、TA2名で指導していたため、1人当たり10数名くらいが目配れる限界と考えてこの人数設定とした。実際に集まった受講者は20名であったが、大学生よりも高校生同士の方が相互に教え合う状況は作りやすかったため、教員1名でも全体に目を配って演習を行うことができた。

5.2 講座のねらい

以下を受講生募集時に講座のねらいとして掲示した。

医療分野における新薬の開発、配達における最短経路の提示というように、実際のサービスだけでなく先端研究においてもAIが一部の人の占有物ではなく、誰もが自身の仕事においてAIをパートナーに効率化を図っていく時代がもうすぐそばまできている。このような状況に鑑み、

- ① AIを取り入れたサービスやアプリ開発の敷居が年々低くなっていることを実際のアプリ開発体験を通して実感してもらうこと
- ② プログラミングの世界に入っていく道は1つでなく、きっかけは人の数だけある。様々な人の体験を聞く中で、様々な入り口に触れてもらう。
- ③ 社会のニーズをどう捉え、製品をデザインしていくか
- ④ いつかは腰を据えて本気でプログラミング言語の勉強をする機会を設けなければならない。最初の一步をこの講座でサポート

を本講座の目的とする。

5.3 開講期間, カリキュラム

当初1年間を通して全17回程度を想定し、関数や機械学習の仕組みまで時間をかけて取り組み、外部講師による出張講義も学習効果の高いタイミングで実施するよう計画していた。しかしながら、COVID-19のため2学期のみ全8回実施で完結するよう組み直した。以下に本実践のカ

リキュラムの概要を示す。

第 1 講 Colab の基本操作, Python の基本演算

Python の開発者・歴史・言語の特徴, 様々な演算子を活用した計算

第 2 講 変数, 文字列, 入出力, 条件文 (if 文)

入出力を伴うコーディング課題に対応するための基本

第 3 講 データ構造 (リスト), 反復文 (for 文, while 文)

反復文を扱うために必要なリストや range 関数などのイテラブルオブジェクトの基本

第 4 講 演習 (じゃんけんゲーム, 数当てゲームの改良)

前講までの知識 (主に if 文) を活用した簡単なゲームの作成

第 5 講 (外部講師) 中央大学での学び, 競技プログラミングについて知る

ICPC2019 の予選を突破した本校卒業生による講演, AtCoder の紹介

第 6 講 (外部講師) LifeIsTech によるアプリ開発ワークショップ

画像認識技術を用いた簡易レジシステムの開発

第 7 講 (外部講師) IBM によるデザイン思考ワークショップ

アプリ開発だけでなく, 幅広く応用できる「デザイン思考」の体験

第 8 講 データ構造 (集合), 画像認識 (学習済み CNN モデル VGG16 による識別体験)

自然言語の特徴分析, ごく簡単な暗号解読を例に集合の活用法, 画像認識の体験と全体のまとめ

Python は他の高級言語と比べて文法が平易なためすぐに習得できると想像していたが, 例えば四則演算以外の比較演算子や論理演算子等は馴染みがなく, 補完機能も働かないため, 演習時にこの部分に関する資料を見直す生徒が非常に多かった。これは外部講師による講義以外の学習に十分な時間を割けなかったことも要因として考えられる。

何事においてもそうだが意欲的な生徒は, 教えられたことに留まらず, 自分が実現したいことを目標に据え, それを実現するための方法を質問したり, インターネットを用いて積極的に調べる傾向がある。只ここで強調しておきたいのは, そのような生徒の疑問に全て教員が答えられる必要はないということだ。アイデアに共感し, 一緒に考え, 必要に応じてより詳しい人との繋がりを作るよう促せば良いのだ。自分に分からないことを質問されずに済むよう実習内容に制約をかけ過ぎると, 生徒の主体性の芽を摘むことになるだろう。題材やテーマは必要に応じて与えるが, それを元に「あなたが好きなように作ってみよう, 改良してみよう」という自由度の高い課題の設定を推奨したい。

5.4 学習環境 (IDE, 教材)

4 節で述べたが、本実践ではクラウド型 IDE である Colab を採用した。IDE を選ぶ際の観点を以下に再掲する。

- (1) 使用に際して新たにアカウント作成が不要 (本校は G Suite を導入済み)
- (2) GPU などのスペックが高く、本格的な開発にも耐えうる環境
- (3) 機械学習に関するデモが豊富に用意
- (4) コードの補完機能, スニペットの充実

Colab を採用したのは (1) に依るところが大きい。というのも、Colab であれば G Suite が提供する Classroom や Drive とシームレスに連携して、教材を用意・提供することができるからだ。また、(3) に関連する Colab の特徴を補足しておく、Colab にはチュートリアルとして「Colaboratory へようこそ^{*5}」が用意されており、基本的な操作法だけでなく、機械学習の興味を引くデモが多く紹介されている。このチュートリアルは講座の導入時だけでなく、一通り学習を終えた後の応用レベルの演習として再度活用することも可能である。

教材は授業用と自宅学習用に 2 つ用意した。授業用教材としては Colab の ipynb 形式ファイル (正確には Jupyter Notebook のファイル形式) で Python の文法に関する解説と、演習課題を独習できるよう細かい説明も含めて作成した。これとは別に説明用のスライドも作成した。また、自宅学習用に paiza ラーニングの学校フリーパスアカウント^{*6}を取得し、受講者へ配布した。paiza ラーニングのスキルチェックでは、複数の言語に対応したオンラインジャッジシステムがあるため、様々なレベルの面白い問題のコーディングに挑戦できる。また paiza ラーニングでは Python 以外の言語 (C, C#, HTML, Java, JavaScript, SQL, PHP 等々) に興味をもった際、それらも全て無料で学習することができる。

6 謝辞

最後に本講座の趣旨に賛同し、COVID-19 が猖獗を極める中、万難を排して興味深い講演を提供してくださった LifeIsTech の讃井様、松井様、丸本様はじめ多くのスタッフの皆様方、IBM の鳥井様、本校卒業生の小野様、ここで改めて感謝の意を表したいと思います。

^{*5} <https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>

^{*6} https://paiza.jp/works/lp/free_pass