

# 我々が用いるプログラミング学習環境について

名古屋高等学校 中西渉  
watayan@meigaku.ac.jp

2022年6月24日

## 1 はじめに

新しい学習指導要領による学習が開始され、高1で履修する学校では既に「情報I」の授業が始まっている。前の学習指導要領の「社会と情報」「情報の科学」から新しく追加された要素はいくつかあり、プログラミングについてもこれからは必須となる。

本稿では、プログラミングの実習を授業で行うにあたって、どのような環境が必要であるかについて考察を行う。とはいえ、筆者および勤務校のPC環境は明らかに多数派とは異なることや<sup>1)</sup>、私立ゆえに情報教室の管理が自由であることなどから、以下で述べることは各学校の抱える事情にはそぐわないことも多々あることと思われる。そのような点については後の質疑やグループ討論で取り上げていただければ幸いである。

## 2 プログラミング学習環境

### 2.1 教科書で想定されている環境

情報Iの各教科書で、主に用いられているプログラム言語を表1にまとめた。以下学習環境につ

いて述べる。ただしVBAとScratchについては自ずと使用環境が決まるので、ここでは言及しない。

Pythonの実習環境としては、特に明示されたものはない。ただし日文「情報I」巻末資料ではIDLEを用いているほか、第一「高等学校 情報I」ではGoogle Colaboratoryが使われている。

JavaScriptの実習環境としては、OS付属のテキストエディタとWebブラウザを用いたものが多い。そのためHTMLファイルにscript要素として埋め込むことになる(図1参照)。数研「高等学校 情報I」「情報I Next」はJavaScriptのコードだけが書かれているが、実際には2.2.3で後述するようにWebブラウザ上での実行を想定していると思われる。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>タイトル</title>
  </head>
  <body>
    <script>
      <!--ここにコードを書く-->
    </script>
  </body>
</html>
```

図1 JavaScriptを埋め込むためのHTML

1) どちらもDebian GNU/Linuxのデスクトップ環境(KDE)である。筆者はVMWare上の仮想環境でWindows 11を用いることがある。

## 2.2 プログラミング学習環境

情報教室の PC にはソフトウェアを自由にインストールできないとか、スペックの関係で重いソフトウェアは使えないという話もよく聞く。これについては然るべきところに働きかけていくしかないとは思いますが、解決していないのが現実だ。生徒に持たせている端末についても学校ごとの事情があるのだが、むしろこちらの方が自由が利くのかもしれない。

低スペックの PC でも Excel はそれなりに動くことを考えると、VBA での実習が現実的であるという判断は頷けるものがある。

### 2.2.1 テキストエディタについて

教科書では OS 付属のテキストエディタ（具体的には Windows の「メモ帳」）が使われているらしきスクリーンショットが見受けられるが、これで実習を行うことはできれば考えたくない。入力補完や自動インデントどころか行番号さえないので非常に不便であるし、生徒に「テキストエディタはワープロの劣化版」のような印象を持たれるのは避けたい。

無料で使えるテキストエディタはいろいろあるのでそれらを使いたい。インストールが自由にできない環境でも、Sublime Text[1] や NotePad++[2], サクラエディタ [3] などはポータブル版もあるので、どこかのフォルダに展開してしまえばそれだけで使える。Sublime Text は日本語化が面倒ではあるが、生徒が今後出会うソフトウェアはメニューが英語表記のものもあるだろうから、あえてそのまま…というのは乱暴にすぎるだろうか。

筆者は Visual Studio Code[4] を愛用しているが、バージョンアップが非常に頻繁になされるため、情報教室のクライアントに導入するには至っていない。また PC スペックによっては、動

作が重すぎて使いづらいのも事実である。

### 2.2.2 Python の実行環境について

Python の実行環境としてはいろいろなものが考えられる。筆者は勤務校では Google Colaboratory を使いたいと考えている。これは学校で用意した生徒アカウントに Google アカウントを紐付けているからだが、学校によっては使用できるような設定がされていない場合もあるという話を聞くことがある。

Google Colaboratory については、テキストとコードを合わせて管理できることや、テキストがマークダウン記法で書けることなどをメリットと考えている。しかしセルをまたいで変数などが共有されるため、注意しないと予想外の動作が起きてしまうことがあるのは難点ではある。

ローカルマシン上で実行するのはいろいろな方法が考えられる。

- (1) コマンドプロンプト（または PowerShell）で手書き
- (2) IDLE で手書き
- (3) テキストエディタで書いてコマンドプロンプトで実行
- (4) テキストエディタで書いて IDLE で実行
- (5) IDLE のエディタで書いて IDLE で実行
- (6) テキストエディタで実行
- (7) IDE で実行
- (8) Jupyter Notebook などで実行

(1)(2) は変数への代入や print の結果を確認する程度ならいいが、それ以上のところでは使いたくない。(3)(4) はコードを保存したフォルダについて迷わないかが心配である。その点 (5) なら保存したフォルダがどこであっても実行はできる。

(6)(7) は Visual Studio Code などのテキストエディタや、Visual Studio などの IDE を想定している。端末の状況が許さないこともある

かとは思いますが、ブレークポイントや変数のウォッチなどのデバッグ機能が自由に使えるメリットは大きい。もっとも IDLE のデバッグ機能でもそういったことはできるのだが。

なお、(7) や (8) は書いてみただけである。Visual Studio や Jupyter Notebook がローカルにインストールできる環境が用意できるような状況があるのなら、誰も困りはしない…。

### 2.2.3 JavaScript の実行環境について

JavaScript の実行環境としては、教科書は Web ブラウザを使うことを想定しているようである。JavaScript 実行環境として Node.js[5] を使うことも考えてみたが、表 1 で JavaScript を使っている教科書はすべて入出力に `prompt`, `alert`, `document.write` のどれかを使っているので、Web ブラウザ上でないとそのままでは実行できない。出力だけなら `console.log` を使えばいいのだが、入力を変数に読み込むのは難しい<sup>2)</sup>。

したがって、テキストエディタで書いて Web ブラウザで動作を確認して…という流れを繰り返すことになるだろう。その際、図 1 に書いたように HTML ファイルに埋め込むか、図 2 のように外部ファイルにして読み込むかは検討に値する。前者は 1 ファイルですべて完結するのでわかりやすいが、前後の HTML をいちいち書かなくてはいけないのが面倒である。後者はファイル名だけを書き換えれば済むが、複数ファイルを管理する手間が生じる。

いずれにしてもテキストエディタと Web ブラウザの複数ウィンドウを行き来しながら作業することになるので、ファイルを置くフォルダを間違えることや、書き換えたファイルの保存忘れ・リロード忘れなどのトラブルが予想される。これは Python の項でも述べたことであり、ひいてはあらゆる実

2) おそらく `readline` モジュールを使うのが一般的と思われる。

習で問題になることでもあるので、妙案があれば教示いただきたい。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>タイトル</title>
  </head>
  <body>
    <script src="hoge.js"
      type="text/javascript">
    </script>
  </body>
</html>
```

図 2 JavaScript を外部ファイルから読み込む HTML

また、デバッグの際には各ブラウザのデベロッパーツールを使用することになる。筆者は Python の項で述べたように、ブレークポイントと変数のウォッチができれば十分と考えているので、それほどややこしいことにはならないとは考えている。繰り返しになるが、Visual Studio Code などデバッグ機能を有するテキストエディタを使うことも候補として考えていいたいだろう。また、スマートフォンのアプリを Web ブラウザ上で開発できるプログラミング環境 Monaca[6] を用いることも考えられる。

## 3 おわりに

大学入学共通テストに情報が追加されることが決定する以前に、大学入試センターからは検討イメージとしての試作問題 [7] やサンプル問題 [8] が公開されている。それらのプログラミングに関する問題では、日本語をベースにした擬似言語である DNCL が用いられている。これまでのセンター試験や共通テストでも「情報関係基礎」で DNCL[9] が用いられてきたが、試作問題やサ

ンプル問題のDNCLは構文がPythonに似たものになっている(図3)。

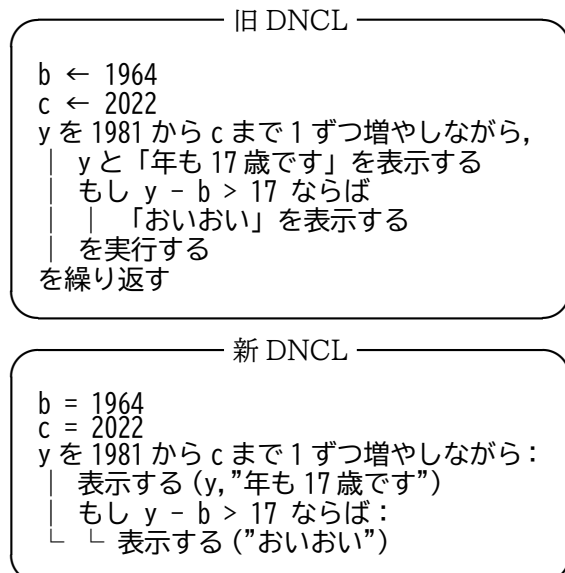


図3 DNCLの新旧比較

旧 DNCL には PEN[10] を始めとする実行環境が作られてきたが、新 DNCL についても筆者の PyPEN[11] や「つちのこ」[12], wPEN[13], 「どんぐり」[14] などの実行環境が対応している。

だったら DNCL で実習をやればちょうど入試対策にもなるのではないかと考えることは自然ではあるが、筆者はそれをするべきではないと考えている。もちろんアルゴリズムだけに着目すれば、これらの環境でも十分なことができるのではあるが、その先、つまりプログラミングを使って何らかの問題解決につなげようとしたときに、ある程度以上進めなくなってしまうからである。この件については、大学入試センター試験問題調査官も同じ趣旨の発言 [15] をしている。

もちろん入試に対応していくことは我々の現実的な責務である。しかしそれは教科書で用いられているプログラム言語で学習することで達成できるものだと考えている。実際、井手の最近の研究

[16] によれば、用いるプログラミング言語をクラスごとに変えて授業を行ったあとで試作問題のプログラミング問題を解かせたところ、結果に大きい差は生じなかったという。我々が「情報関係基礎」の過去問 [17] を参考にするにあたって、旧 DNCL から新 DNCL への読み替えが必要になる。「その程度」の読み替えが障害になってはいけない。

表 1 各教科書で用いられているプログラム言語

出版社	書名	使用言語			
		P	J	V	S
東書	新編情報 I	○	○		
東書	情報 I Step Forward! (理論編) (実習編)	○			
		○	○	○	○
実教	高校情報 I Python	○			
実教	高校情報 I JavaScript		○		
実教	最新情報 I			○	
実教	図説情報 I				○
開隆堂	実践 情報 I			○	
数研	高等学校 情報 I 巻末実習	○	○	○	
		○		○	
数研	情報 I Next	○	○	○	
日文	情報 I 巻末資料	○			
		○	○		
日文	情報 I 図解と実習				○
第一	高等学校 情報 I 巻末資料			○	
		○			

※ P=Python, J=JavaScript, V=VBA, S=Scratch

## 参考文献

- [1] Sublime HQ Pty Ltd. “Sublime Text”. <https://sublimetext.com/>.
- [2] Don Ho. “Notepad++”. <https://notepad-plus-plus.org/>.
- [3] Sakura Editor Organization. “サクラエディタ”. <https://sakura-editor.github.io/>.
- [4] Microsoft. “Visual Studio Code”. <https://code.visualstudio.com>.
- [5] OpenJS Foundation. “Node.js”. <https://nodejs.org/ja/>.
- [6] アシアル情報教育研究所. “Monaca Education”. <https://edu.monaca.io/>.
- [7] 情報処理学会. “大学入学共通テストへの「情報」の出題について”. <https://www.ipsj.or.jp/education/edu202012.html>, 2020.
- [8] 大学入試センター. “令和 7 年度以降の試験に向けた検討について—サンプル問題”. [https://www.dnc.ac.jp/kyotsu/shiken\\_jouhou/r7ikou.html](https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html), 2021.
- [9] 大学入試センター. “共通テスト手順記述標準言語 (DNCL) の説明”. <https://www.dnc.ac.jp/albums/abm00040701.pdf>, 2021.

- [10] 中村 亮太, 西田 知博, 松浦 敏雄. “プログラミング入門教育用学習環境 PEN”. 情報処理学会研究報告コンピュータと教育 (CE), Vol.2005-CE-081(104), 2005.
- [11] 中西 渉. “プログラミング学習環境 PyPEN の開発”. 日本情報科教育学会 第 13 回研究報告書, (5), 2019.
- [12] 大門 巧. “つちのこ”. <https://t-daimon.jp/tsuchinoko/>.
- [13] 下村 亮太, 中西 通雄, 松浦 敏雄, 西田 知博, 安留 誠吾, 宮本 友介. “初学者用ブロックプログラミング環境 wPEN の改良”. 情報処理学会研究報告コンピュータと教育 (CE), Vol.2021-CE-159(5), 2021.
- [14] 本多 佑希, 漆原 宏丞, 兼宗 進. “試験問題記述言語 DNCL の改定に合わせた「どんぐり」システムの修正と検討”. 情報処理学会研究報告コンピュータと教育 (CE), Vol.2021-CE-159(2), 2021.
- [15] 水野 修治. “大学入学共通テスト新科目「情報 I」～サンプル問題等とそのねらい～”. <https://www.wakuwaku-catch.net/kouen210801/04/>.
- [16] 井手 広康. “大学入学共通テストを見据えたプログラミング教育の言語選択に関する考察”. 情報処理学会研究報告コンピュータと教育 (CE), Vol.2022-CE-163(14), 2022.
- [17] 情報処理学会情報入試委員会. “情報関係基礎 アーカイブ”. [https://sites.google.com/a.ipsj.or.jp/ipsjnn/resources/JHK](https://sites.google.com/a/ipsj.or.jp/ipsjnn/resources/JHK), 2021.