Webブラウザ上のプログラミング学習環境 PyPENの改良

中西渉[1,a)]

概要：筆者は高校の情報科の授業で用いるために，大学入試センター「情報関係基礎」で用いられているプログラミング言語 DNCL の構文表記を Python 風にして，Web ブラウザ上で動作するプログラミング学習環境 PyPEN を開発した．本稿では，これまでにいただいた意見を元に行った改良について報告を行うとともに，今後の開発方針についての考えを述べる．

キーワード：プログラミング教育，日本語プログラミング，高等学校「情報 I」，HTML5，Python

Improvement of PyPEN Programming Learning Environment on Web Browser

Wataru Nakanishi[1,a)]

Abstract: I developed PyPEN, a programming learning environment that runs on a Web browser. It uses a new language similar to Python and DNCL. In this paper, I will report on the improvements based on the opinions I have received so far, and state my thoughts on future development policies.

Keywords: Education of programming, Programming with Japanese, Informatics I, HTML5, Python

1. はじめに

コロナ禍の影響でよくわからなくなっているが，今年度から小学校でプログラミング教育が導入されることになっている．高校の情報科でも 2022 年度の学習指導要領改定からは必履修科目「情報 I」で全生徒がプログラミングを学習することになるが，2015 年度の調査ではプログラミングが含まれる「情報の科学」を履修している生徒の割合は 2 割程度であり，プログラミングを指導してこなかった教員も一定数いると考えられる [1]．

文部科学省もこれに対応して教員研修用教材 [2] を提供した．その「情報 I」の教材の第 3 章「コンピュータとプログラミング」で最初に取り上げられたことや，人工知能ブームなどから，昨今 Python に注目が集まっているように思われる．

筆者はこれまで Web ブラウザ上で動作するプログラミング学習環境である WaPEN[3] を開発してきたが，この流れに乗って構文を Python 風にアレンジした PyPEN を開発した [4]．本稿ではその後 PyPEN に施した改良点および今後の開発方針について考えるところを述べる．

2. PyPEN について

2.1 概要

PyPEN は Web ブラウザ上で動作するプログラミング学習環境であり，初学者向けプログラミング学習環境 PEN で用いられていたプログラミング言語 xDNCL の構文を，一部 Python 風のインデントで表現するようにしたものである．動作画面を**図 1** に示す．

プログラム入力には画面下部の「入力支援ボタン」を使うことで，キー入力を大幅に減らすことができる．プログラムはフローチャートを編集することでも作成することが

[1] 名古屋高等学校
Nagoya Senior High School
[a)] watayan@meigaku.ac.jp

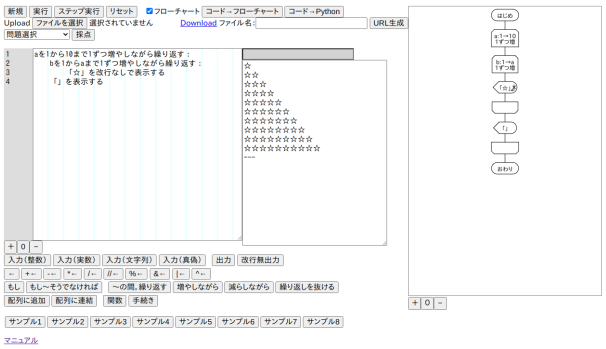

図 1　PyPEN の動作画面
Fig. 1　Screenshot of PyPEN

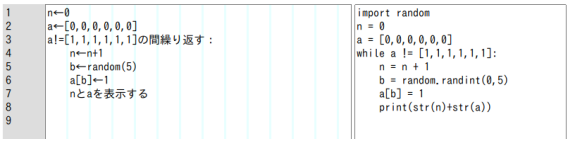

図 2　Python のコードへの変換
Fig. 2　Converting to code for python

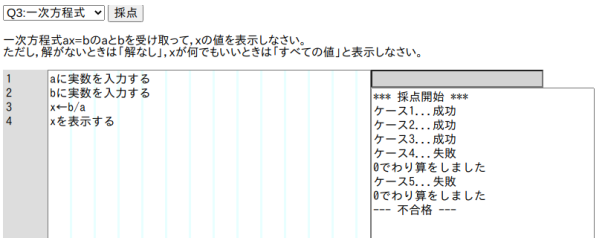

図 3　自動採点
Fig. 3　Auto marking

でき，逆に「コード→フローチャート」ボタンでプログラムをフローチャートに変換することができる．ただし自作関数・自作手続きを含むプログラムにはフローチャートは対応していない．

「コード→ Python」ボタンは，Python に変換したコードを表示するためのものである（グラフィック命令などには非対応）．これはコピーして Python 上で実行することを想定しており，必要なモジュールがあれば import 命令も出力するようになっている．変換した様子を図 **2** に示す．

サンプルプログラムは `sample.js` というファイルを書き換えることで差し替えることができる．また `answer.js` というファイルに問題を設定し，指定した入力に対し期待通りの出力が得られるかを自動採点することもできる（図 **3**）．

PyPEN 自体は JavaScript で作られており，Web サーバまたはローカルディスク上にファイルを設置して，ブラウザで読み込むことで動作させることができる．

3.　先行研究

3.1　日本語プログラミング環境

前述した通り，PyPEN の命令などの表記は PEN[5] で用いられてきた xDNCL を元にしている．筆者は PEN にフローチャートを付加した PenFlowchart[6] を開発した．

ところで，Python 風にインデントでブロックを表現する方法は，日本語プログラミング言語「なでしこ」[7] のバージョン 1 で既に用いられていた．バージョン 3 ではブロック終わりを示す「ここまで」が追加されているが，PyPEN がインデントによるブロックを処理するために内部で用いている方法はこれに近いものである．

3.2　Web 上のプログラミング学習環境

PEN や PenFlowchart は Java アプリケーションなので多くの OS 上で動作させることができるが，最近 Java 自体のインストールが敬遠される傾向にあるように思われる．そこで筆者は PenFlowchart とほぼ同機能のものを Web ブラウザ上で動作できる WaPEN を開発した．

同じ問題をドリトルも抱えていたと思われるが，これも Bit Arrow[8] という形で Web アプリケーション化されている．

Scratch[9] は言うまでもなく広い校種で使われているし，スマートフォン用アプリ開発環境 Monaca[10] を使った実践も多く行われている [11]．

4.　PyPEN の改良

以下，PyPEN に対して行った改良点について述べる．なお，これらの改良を行うにあたって，PyPEN を Internet Explorer 11（以下，IE11）で動作させることを断念した．学校現場では IE11 が使われることは多いようではあるが，Microsoft 社自身がネットサービスでの IE11 のサポートを打ち切って Microsoft Edge への移行を勧めている状況であるから [12]，影響は限定的であると考えている．

さらに後述するように，配列や辞書に関する文法を一部変更したため，これまで動作していたプログラムが動かなくなる可能性がある．

4.1　演算子の追加・変更

これまで PyPEN の演算子は四則演算に限ったものであったが，これに`&`，`|`，`^`，`~`といったビット演算を加えた．さらに`+←`や`^←`などの複合代入演算子も追加した．

また，代入に←だけでなく`=`が使えるようにした．←という記号は Pascal や Python3.8[13] の代入式で導入された`:=`同様，代入という動作の非対称性を示すために有効だと考えているのだが，他の言語の経験がある生徒は代入に`=`を使うことが手癖になっていることもあるので，それに対応して`=`でも良いようにしたのである．それに合わせて，先に述べた復号代入演算子についても`+=`といった表記ができるようにしている．なお，比較で用いる`=`は以前から`==`にすることもできるようになっている．

文字列はこれまで「」で表すのを標準にしていたが，これ

```
a ← [1,1]
i を 3 から 50 まで 1 ずつ増やしながら繰り返す：
    a に a[-2]+a[-1] を追加する
a を表示する
```

図 4　フィボナッチ数列の生成
Fig. 4　Generation of Fibonacci numbers

では二分アキのためにインデントのスペースの数を間違えることが考えられるため，""を標準にすることに変更した．互換性を維持するため「」も使えるようにはしているが，フローチャートからコードを生成するときなどには""を用いるようにしている．

4.2　配列操作を Python に近づける

これまでは変数に値を代入しなくても参照でき，配列も要求に応じて無条件に延長するようにしていたが，これは Python の振る舞いとは違ったものである．そのため「コード→ Python」ボタンで変換したプログラムが Python で実行できないということが起きていた．

そこで，値が代入されていない変数の参照や，配列の範囲外へのアクセスはエラーになるようにした．その代わり Python の `append` や `extend` のように，配列の末尾に値を追加したり配列を連結したりすることができるようにした．具体的には「《配列》に《値》を追加する」「《配列》に《配列》を連結する」という構文を追加した．

また，配列の添字に`-1` などの負数や `2:4` のようなスライス風のものも指定できるようにした．これを使えば，たとえばフィボナッチ数列を生成するプログラムを図 4 のように書くことができる．

ところで，Python では区別されているリストと辞書を，これまでは PyPEN ではまとめて配列として扱っていた．しかしそのままでは実装がかえって面倒になると判断したため，PyPEN でも配列と辞書を区別するようにした．それに対応して，これまで添え字をつければ勝手に配列や辞書と解釈していたのを，最低でも空の配列 `[]` や空の辞書 `{}` を代入していないと使えないようにした．この変更により，これまで動いていたプログラムに修正が必要になることがある．

4.3　グラフ機能の強化

PyPEN は PEN から引き継いだグラフィック機能を持っているので，それを使えばグラフ描画が可能ではある．しかしいちいち座標を計算して描画するのは面倒であるし，初学者にとってハードルが高いものになってしまう．棒グラフや折れ線グラフを簡単に書く構文は既に用意してあるが，非常に貧弱なものである．Monaca では `Plotly.js`[14] を利用して簡単なコードでグラフ描画ができているので [15]，PyPEN でも `Plotly.js` を取り込んでグラフ描画ができるようにした．実行イメージを図 5 に示す．

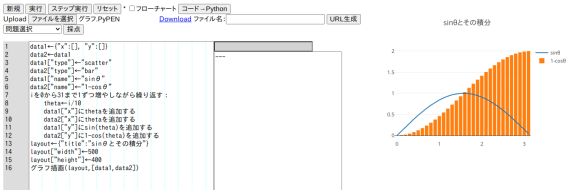

図 5　グラフ描画の例
Fig. 5　Sample of drawing graph

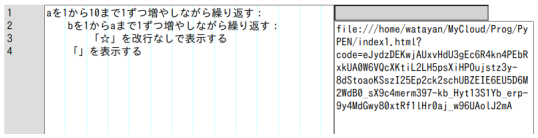

図 6　URL 生成画面
Fig. 6　Generating URL

現状では `Plotly.js` の表示用の関数 `newPlot` が要求するデータを直接渡す構文になっている．グラフの種類別にわかりやすい構文を用意することも考えられるが，`Plotly.js` があまりに高機能であるため，それら一つ一つについて個別の構文を用意することはかえって利用の幅を狭めることになると考えた．

4.4　URL でのプログラム読み込み

Bit Arrow は作成したプログラムをサーバに保存し，それを QR コードなどを用いて共有できるようになっている．しかし PyPEN はサーバを用意しなくても使えるようにしているので，同じ機能を実現することはできない．しかし作ったプログラムを何らかの形で共有する仕組みは用意したいと考えたので，URL の GET 文字列でプログラムを読み込めるようにした．

これは URL の末尾に`?code=...`の形式で指定するもので，プログラムを zlib で圧縮してから Base64 でエンコードした文字列が続く．ただ，普通の Base64 で使われる`+`，`/`，`=`は URL では別の意味で用いられるので，`+`の代わりに`-`，`/`の代わりに`_`を用い，パディングの`=`は省略することにした（いわゆる Base64URL）．

`code` を含んだ URL は画面の「URL 生成」ボタンで生成することができるので，それをリンクするような文書を作ることで，ソースコードが入力された PyPEN のページにユーザを誘導することができる．ただしこれは非常に長い URL になってしまうことがあり，ブラウザによっては長すぎる URL を受け入れないものがあるということなので，zlib で圧縮した時点で 1500 バイトを超える（つまり Base64 エンコードすると 2000 バイトを超える）ようなプログラムの URL は生成しないことにした．画面イメージを図 6 に示す．

```
[TITLE]
最大公約数
[QUESTION]
2 つの整数を受け取って，最大公約数を表示しなさい
[INPUT]
35
21
[OUTPUT]
7
[INPUT]
527215277
522902525
[OUTPUT]
1
[TIMEOUT]
100
```

図 7　自動採点の問題・解答ファイルの例
Fig. 7　Example file for automarking

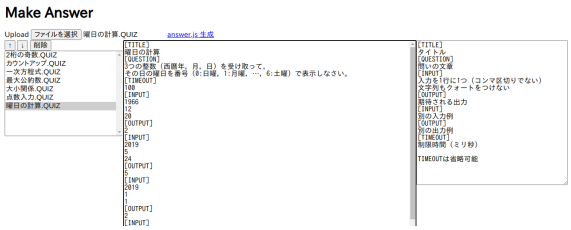

図 8　MakeAnswer の実行画面
Fig. 8　Screenshot of MakeAnswer

4.5　MakeSample, MakeAnswer の Web アプリ化

サンプルプログラムや自動採点の問題・解答はそれぞれ `sample.js`，`answer.js` というファイルに JavaScript の構文にしたがって記述するが，これを手作業で編集するのは面倒である．そこでこれらのファイルを生成するプログラム `MakeSample`，`MakeAnswer` を，複数 OS で利用できるように Qt[16] を用いて開発した [17]．`MakeSample` は複数の PyPEN のプログラムをまとめて `sample.js` を生成するものであり，`MakeAnswer` は図 **7** のような形式で自動採点の問題・解答を書いたファイル（拡張子は `QUIZ`）を用意すれば，それらをまとめて `answer.js` を生成するものである．

2019 年の情報教育シンポジウムでこれについて述べたところ，「OS を問わず利用できるようにするのであれば，Web アプリケーションにした方が良いのではないか」と助言をいただいた．Qt で開発した場合，ソースは共通ではあるもののビルドやインストーラは OS ごとに必要であるし，それほど頻繁に使うプログラムでもないといったことを考慮して，Web アプリケーションに切り替えることにした [18]．`MakeAnswer` の実行画面を図 **8** に示す．

4.6　その他

これまでは「コード→ Python」ボタンで生成した Python 用のコードを別ウィンドウに表示していたが，それを複数開いてしまった場合にどれが最新版かわからなくなってしまうなどの面倒があったので，コードは別ウィンドウを使わずに出力結果表示エリアに表示することにした．また複数の値を表示する場合，それらをコンマ区切りで指定できるようにした（表示は空白区切りになる）．

5.　今後の開発方針について

5.1　構文拡張

PyPEN のプログラミング言語は PEN で用いられてきた xDNCL に端を発するものであるが，Python 風の構文表記をすることなどによって既にまったく別のものになっている．たとえば WaPEN のプログラミング言語は xDNCL に近いものであるから，センター試験「情報関係基礎」の問題にあるプログラムをほぼそのまま実行することができるのであるが，PyPEN ではそれは適わない．流通している高校情報科の教材に DNCL 系の言語を採用しているものは（筆者の知る限り）ないのだから，PyPEN のプログラミング言語が DNCL に似た構文にこだわる必要はもはやない．

Python への橋渡しとして PyPEN を用いることを考えれば，むしろ Python にある構文をもっと多く日本語で実現したほうがいいのではないかと考えている．今でも Python に翻訳したコードを出力する機能は実装しているが，たとえば `for ～ in` や `range` に相当するものが用意できれば，より自然な読み替えができるものになる．

しかし日本語での適当な表現が思いつかなくて，実装には至っていない．新しい言語なのだから自由に決めていいことではあるのだが，安直に決めてしまうとそれに縛られることになってしまう．たとえばこれまで PyPEN で配列を代入するときには `[ ]` と `{ }` の両方が使えたが，これはこの機能を実装するときに片方に決めきれなかったためにそうなっていたのであり，既に述べた配列関係の仕様変更により過去のプログラムの修正を余儀なくされた．このような「迷い」を極力排除したい．

5.2　変数のスコープの見直し

PyPEN では変数のスコープはグローバル変数と関数や手続き内のローカル変数でしか区別していない．今はグローバル変数が関数や手続きの中から参照・変更ともにできるようになっているが，これでは同じ名前のローカル変数を使ったときにトラブルを起こしやすくなると考えられる．おそらく，グローバル変数を変更するには Python の `global` に相当する指定を求めるよう，変数の管理ルールを修正する必要がある．

5.3　構文木の実行について

Web ブラウザ上のプログラミング環境の多くは，構文木を JavaScript のコードに変換して実行しているという．しかし PyPEN では構文木をそのままたどりながら実行して

おり，これが実行速度の面で不利であることは否めない．

プログラムの無限ループを中断できるようにするために `setTimeout` を使っていた部分を，`postMessage` を使ったもの [19] に置き換えることで速度向上を図ってはいるが，満足いくものではない．この点は筆者の力量不足によるものであり，引き続き改良を考えていきたい．

5.4 WaPEN へのフィードバック

PyPEN の開発に取り組み始めてから，WaPEN の開発が滞っている．勤務校の授業で用いるものを WaPEN から PyPEN に変えたためにそのようになっているのだが，PyPEN で行った改良の多くはこれまでの xDNCL のプログラムに影響を及ぼすこともなく，またいくつかの共通するバグフィックスも含まれているので，WaPEN をこのままにしておくのは良くないと考えている．

両者の違いは構文解析の部分がほとんどなので，機会をみて WaPEN も大改造を行って同じ機能を持たせられるようにしたい．

6. おわりに

PyPEN でできないことはとても多い．たとえば，なでしこバージョン 3 は Web アプリケーションであるからバージョン 1 のようにローカルファイルの操作はできないが，それでもプラグインによって AJAX や DOM 操作，ローカルストレージなど多くの機能を実現している．PyPEN にはそういった機能はなく，ブラウザの画面に表示されている入力ウィンドウと出力ウィンドウ，若干のグラフィックがすべてであり，その意味で非常に貧弱なものである．

では PyPEN もそのような「実用的」な機能を備えるべきなのだろうか．もちろんそれも一つの方向性ではあるが，あくまでも教育用のプログラミング学習環境として，他の本格的な言語への橋渡しをすることも重要な役割であると筆者は考えている．たとえばドリトルが 1 時間でゲームを 1 つ作る教材 [20] を用意しているように，PyPEN で短時間学習すれば Python に自然と移行できるような教材が必要なのだろう．

PyPEN は GitHub の以下のアドレスで公開している．筆者のサイトにも圧縮したファイルが置いてあるが，GitHub から入手した方が更新などの点で便利だと思われる．

`https://github.com/watayan/PyPEN.git`

本稿で述べた改良点について，実践に基づく多くの示唆をいただいた井上智香子氏に感謝する．

参考文献

[1] 文部科学省: 情報教育に関する資料，教育課程部会　情報ワーキンググループ（第 1 回）配布資料 (2015).

[2] 文部科学省：高等学校情報科教員研修用教材，文部科学省（オンライン），入手先 ⟨https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416746.htm⟩（参照 2020-10-02）.

[3] 中西　渉: WaPEN...DNCL の Web ブラウザ上の実行環境におけるフローチャートなどの実装，情報教育シンポジウム論文集，Vol. 2018, No. 31, pp. 210–214 (2018).

[4] 中西　渉: プログラミング学習艦橋 PyPEN の開発，日本情報科教育学会第 13 回研究会報告書，pp. 19–22 (2019).

[5] 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，研究報告コンピュータと教育（CE），Vol. 2005-CE-081, No. 104 (2005).

[6] 中西　渉: PenFlowchart の開発，研究報告コンピュータと教育（CE），Vol. 2012-CE-113, No. 13 (2012).

[7] くじらはんど: なでしこ：日本語プログラミング言語，（オンライン），入手先 ⟨https://nadesi.com⟩（参照 2020-10-01）.

[8] 兼宗　進，並木美太郎，長　慎也: オンラインプログラミング環境ビットアロー（Bit Arrow），大阪電気通信大学，東京農工大学，明星大学（オンライン），入手先 ⟨https://bitarrow.eplang.jp/⟩（参照 2019-05-30）.

[9] MIT: Scratch–Imagine, Program, Share, (online), available from ⟨https://scratch.mit.edu/⟩ (accessed 2020-10-13).

[10] アシアル株式会社: Monaca—すべての人にアプリ開発を，（オンライン），入手先 ⟨https://ja.monaca.io/⟩（参照 2020-10-13）.

[11] 和歌山県教育委員会：きのくにＩＣＴ教育　高等学校＜共通教科情報科＞プログラミング教育　学習指導案，（オンライン），入手先 ⟨https://www.pref.wakayama.lg.jp/prefg/501100/ictforum_d/fil/kinokuniICT_hs.pdf⟩（参照 2020-10-13）.

[12] Microsoft: Microsoft 365 アプリの Internet Explorer 11 のサポート終了と Windows 10 での Microsoft Edge レガシー版のサービス終了，（オンライン），入手先 ⟨https://blogs.windows.com/japan/2020/08/18/microsoft-365-apps-say-farewell-to-internet-explorer-11/⟩（参照 2020-10-01）.

[13] Hettinger, R.: What's New In Python 3.8, Python Software Foundation (online), available from ⟨https://docs.python.org/3/whatsnew/3.8.html⟩ (accessed 2020-10-13).

[14] Plotly: Plotly JavaScript Open Source Graphing Library, (online), available from ⟨https://plotly.com/javascript/⟩ (accessed 2020-10-01).

[15] 岡本雄樹，辰己丈夫: 高等学校「情報 I」の研修資料を JavaScript 言語で実施するうえでの検討，情報処理学会研究報告　コンピュータと教育（CE），Vol. 2019-CE-151, No. 3 (2019).

[16] The Qt Company: Qt Cross–platform software development for embedded & desktop, (online), available from ⟨https://www.qt.io/⟩ (accessed 2019-05-29).

[17] 中西　渉: Web ブラウザ上のプログラミング学習環境 WaPEN の改良，情報教育シンポジウム論文集，Vol. 2019, pp. 130–135 (2019).

[18] watayan: WaPENTools, (online), available from ⟨https://github.com/watayan/WaPENTools⟩ (accessed 2020-10-01).

[19] Baron, D.: setTimeout with a shorter delay, (online), available from ⟨https://dbaron.org/log/20100309-faster-timeouts⟩ (accessed 2020-10-02).

[20] 兼宗　進: 1 時間で学ぶソフトウェアの仕組み，（オンライン），入手先 ⟨https://dolittle.eplang.jp/d1h⟩（参照 2020-10-02）.