「高等学校情報科『情報 I』教員研修用教材」第 3 章への PyPEN の適用

中西 渉[1,a)]

概要：文部科学省「高等学校情報科『情報 I』教員研修用教材」の第 3 章「コンピュータとプログラミング」は Python 版がまず発行され，他言語版も順次発行される予定とのことである．本研究では，筆者が開発しているプログラミング学習環境 PyPEN でこの教材の各学習を実行できるかを検証し，既に発行されている JavaScript 版，VBA 版と比較した．また，必要に応じて若干の追加開発を行った．

1. はじめに

2022 年度から実施される高等学校学習指導要領の情報科では「情報 I」が選択でない必履修科目となり，高校生のほぼ全員が情報科でプログラミングを学習することになった．どのような環境で実習を行うかということが関係教員の間で話題になる中，人工知能やデータサイエンスが注目されていることや，文部科学省から発行された「高等学校情報科『情報 I』教員研修用教材」[1]（以下では「教材」と表記する）で最初に取り上げられたことなどから，Python に注目が集まっている．「教材」では他の言語版も順次発行するということではあるが，筆頭に取り上げられたことにはインパクトがあった．

筆者はこれまで大学入試センター試験で用いられてきた DNCL という言語を元にしたプログラミング学習環境 PEN を元に PenFlowchart や WaPEN を開発してきたが，上述した流れを受けて DNCL を Python 風にアレンジしたプログラミング学習環境 PyPEN を開発しているところである [2]．

予告通り「教材」のプログラミングに対応する第 3 章について JavaScript 版や VBA 版も公開されたが [3]，Python 版のプログラム部分だけを差し替えているので対応しきれていない学習内容がいくつかある．これが完全でなくていいのであれば PyPEN でもそれなりに実習の役に立つのではないかと考え，比較検討を行うことにした．

本稿では第 2 章で関連する先行研究，第 3 章で PyPEN について述べた後，第 4 章で「教材」への PyPEN や他言語の対応について評価し，第 5 章で考察する．

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

2. 先行研究

2.1 Web 上のプログラミング学習環境

Web 上のプログラミング学習環境にはさまざまなものがあり，DNCL に限っても WaPEN[4]，Bit Arrow の DNCL（どんくり）[5]，Tetra[6]，BlocklyPEN，wPEN[7] などがある．一方，Bit Arrow では Python の実行も可能になっており [8]，ライブラリを使うためにサーバ上での実行もサポートするなどの工夫がなされている．

オンラインで多くの言語のプログラミングを学習できるサイトも paiza.io[9] をはじめ，さまざまなものがある．track[10] は複数言語に対応しやすいプラットフォームであるとのことだが，企業での研修等を対象にしており，一般に公開しているものではない様子である．Wandbox[11] はサイト上でプログラミングができる他，ソースや Docker イメージも配布されているのだが，筆者の手元で環境構築がまだできていないので細かい検証はしていない．

2.2 プログラミング学習環境における統計機能

ドリトルは言語を拡張して，統計・グラフ機能を付加する実装が行われた [12]．ファイルから読み込んだテーブルオブジェクトに対して操作を行い，統計量を計算したり何種類ものグラフが簡単な命令で書けるようになっている．

統計ではある程度多くの数値を扱いたいので，外部ファイルを読み込めるようにしたい．ブラウザ上の JavaScript でローカルファイルを読み出すことには制限があるのだが，オンライン版のドリトルでは Web ストレージ上にファイルを置くことでデータの読み込みを実現している [13]．

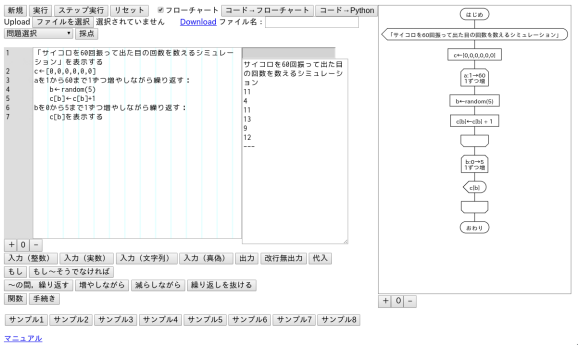

図 1　PyPEN の実行画面

表 1　PyPEN と DNCL の違い

PyPEN	DNCL
もし～ならば： 　命令 そうでなければ： 　命令	もし～ならば ｜命令 を実行し，そうでなければ ｜命令 を実行する
～の間繰り返す： 　命令	～の間， ｜命令 を繰り返す

表 2　関数・手続きの定義

関数　関数名（引数,...） 　命令 　～を返す	手続き　手続き名（引数,...） 　命令

2.3　「教材」に関する検討

兼宗 [14] は「教材」第 3 章の学習 11～17 について，収録を予定されている各言語が対応しているかについて述べ，言語によっては外部ライブラリを使わなくてはいけないことに触れている（特にグラフィック，Web API など）．一方で岡本ら [15] は「教材」がまだ Python 版しか公開されていなかったときに，各学習の多くが JavaScript でも実現できることを Monaca や Node.js を用いて検証した．さらにライブラリに関する先の指摘に対して，Python でもグラフィック表示は `matplotlib` を用いているのだから同様ではないかと疑問を呈している．

3.　PyPEN について

3.1　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，言語は大学入試センター試験「情報関係基礎」で使われている DNCL を Python 風にアレンジした独自言語である．またコードをフローチャートに表したり，フローチャートを作ることでコードを生成することもできる（現状ではフローチャートは自作関数や自作手続きには対応していない）．JavaScript だけで作られているので，Web サーバには特別な設定は必要ない．

Google chrome での動作画面を**図 1** に示す．左側のテキストエリアにコードを入力し，「実行」ボタンで実行すると結果が中央のテキストエリアに表示される．「コード→フローチャート」ボタンでコードをフローチャートに変換できるし，フローチャートを編集すれば即座にコードに反映される．下側の「入力支援ボタン」を使うことで構文の雛形が入力されるので，少ないキー入力でプログラムを作ることができる．

変数表示画面はないが，「変数を確認する」という命令でスコープ内の変数をすべて表示するようにした．また，パーサが出力する構文エラー位置の表示が日本語に対応しておらず読みにくかったので，Jison で生成したパーサにパッチを当てて日本語でエラー表示するようにしている．

3.2　Python「風」言語

Python 風といっても，似ているのは構文の表現方法など一部に過ぎない．分岐やループにおいて，実行する命令のグループをインデント（基本はスペース 4 個）で表現する．また，インデントの前の行末には「：」が置かれる．こういった点ではむしろ「なでしこ」[16] に近い表現と言えるかも知れない．PyPEN と DNCL での分岐・ループの表現の違いを**表 1** に示す．

また変数の定義を廃止し，変数は型を持たず，代入したときに生成されるようにした．値自体は整数，実数，文字列，真偽の型を持つので，「a に整数を入力する」というように入力命令に型の指定を追加した．配列の添字には 0 から始まる自然数だけでなく文字列を使えるようにした．これで連想配列のような扱いができる．

関数・手続きは**表 2** のように定義する．関数は命令や計算の中で使われて値を返すものであり，手続きはそれ自体が命令となる．Python ではこういった区別はないし，前述したどんくりや Tetra では関数で統一しているようであるが，PyPEN では実装上の都合で区別している．

3.3　サンプルプログラム呼び出しボタン

図 1 のサンプルボタンを押すと，予め用意したサンプルプログラムが入力される．そのサンプルは `sample.js` というファイルにデータとして書かれており，これを書き換えることで自分の授業に合わせたものに差し替えることができる．

とはいえ，この書き換えを手作業で行うのは実際にはかなり面倒であるため，サンプルプログラム 1 つ 1 つをファイルに保存しておいて，それらを集めて `sample.js` を生成するプログラム MakeSample を開発した [17]．これ

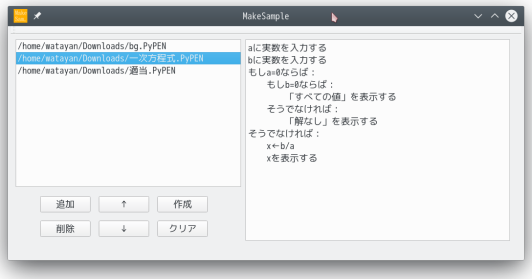

図 2　MakeSample の実行画面

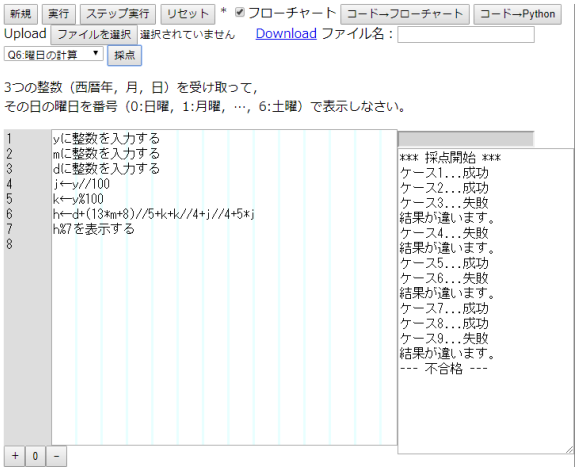

図 3　自動採点の様子

は WaPEN でも共通して使えるものであり，多くの OS で使えるように Qt[18] を用いて作ったのだが，「WaPEN や PyPEN 自体が Web アプリケーションなのだから，これらのツールもブラウザ上で使えるようにするといいのではないか」との助言をいただいているので，今後はその方向に転換する予定である．実行画面を図 **2** に示す．

3.4　自動採点

問題を選択して表示し，いくつかのテストケースを入力して結果が予期したものであるかを自動的に判定する機能を追加した．テストケースは `answer.js` というファイルにデータとして書かれるので，これもサンプルプログラムと同様に MakeAnswer というプログラムで生成できるようにしている．判定画面を図 **3** に示す．

3.5　Python コードの出力

PyPEN の言語は Python に直訳しやすく作ったのではあるが，もちろん Python そのものではない．そこで「コード→ Python」ボタンを押すと別窓を開いて Python のコードを出力するという機能を追加した（図 **4**）．出力されたコードはそのまま Python で実行できるものである．グラフィック命令には対応していないが，乱数や数学関数のようにモジュールを必要とするものについてはそれらを import する命令も出力される．

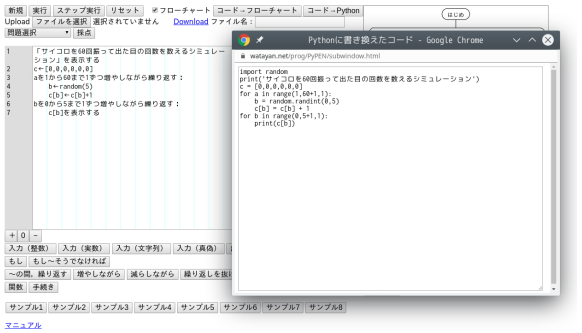

図 4　Python のコードを出力した様子

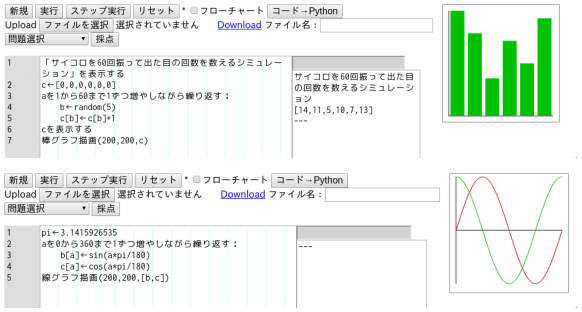

図 5　グラフ描画の例

3.6　簡単なグラフ作図

PyPEN にはグラフィック命令があるので，それを使って折れ線や四角形を描けばグラフ表示もできるのだが，そのためのプログラムを書かなくてはいけないのは面倒である．そこで，配列をそのまま棒グラフや折れ線グラフにする機能を追加した．実行例を図 **5** に示す．今のところ棒グラフは 1 次元配列，折れ線グラフは 2 次元配列を引数にとるものとしているが，これはいずれ統一してどちらでもいいようにする予定である．

4.　「教員研修用教材」への対応

「教材」で取り上げられている各学習のプログラムについて，Python，JavaScript，VBA でのサンプルコードが掲載されているか，また PyPEN でそれを再現できるかを表 **10** にまとめた．以下，主に PyPEN での対応状況を述べる．

4.1　学習 11　コンピュータの仕組み

学習 11 ではオーバーフローと小数計算の誤差を扱っている．JavaScript でこれらが実現可能であることは [15] で述べられているが，PyPEN では少し事情が違ってくる．たとえば JavaScript では有限の範囲で扱えない大きい値（たとえば 1.8×10^{308}）を変数に代入しても `Infinity` として扱われるだけでエラーにはならないのだが，PyPEN では構文解析のときに数値に変換するので，その時点で構文エラーとなってしまって実行できなくなる（表 **3**）．そのため表 **4** のようなコードで確認することになるであろう．

表 3　大きすぎる数をコードに書いた時の振る舞い

コード	実行結果
x ← 1.8e308 x を表示する	1 行目構文エラーです オーバーフローしました

表 4　オーバーフローのプログラム

コード	実行結果
x ← 1.7e308 x を表示する x+0.1e308 を表示する	1.7e+308 実行時エラーです 3 行目:オーバーフローしました

表 5　整数オーバーフローのプログラム

コード	実行結果
x ← pow(2,52)-1+pow(2,52) x を表示する x+1 を表示する	9007199254740991 実行時エラーです 3 行目:整数で表される範囲を越えました

また PyPEN では値を整数と実数で区別して扱っているので，**表 5** のように整数で扱える $\pm(2^{53}-1)$ を境にして実行エラーを起こすことができる[*1]．

なお，計算誤差の例として 0.28 – 0.27 を表示すると 0.01 でなく 0.010000000000000009 のようになることは PyPEN でも容易に確認できる．勤務校の授業では `0.1*2=0.2` と `0.1*3=0.3` の真偽が異なることを例にとって説明している．

4.2　学習 12　外部装置との接続

学習 12 では外部装置の例として micro:bit が取り上げられているが，PyPEN はこれにはまったく対応していないし，他の機器についても対応する方法が思いつかない．

4.3　学習 13　基本的プログラム

学習 13 は「順次」「分岐」「反復」とその組み合わせであるが，これは基本的なものなので問題ない．

4.4　学習 14　応用的プログラム

学習 14 では「配列・リスト」「乱数」「関数」「Web API」が扱われている．これらのうち Web API 以外の 3 つについては基本的なものなので問題ない．

PyPEN では Web API へのアクセスは実装されていない．そもそも [15] で指摘されているように，ブラウザ上の JavaScript で実装されている PyPEN では CORS の制限があるので，外部の Web API を利用することは難しいと考えられる [19]．同じオリジン，たとえば PyPEN を設置したサイトと同じサイトの API にならアクセスすることは可能であるが，公開されている外部の情報にアクセスするのでないと API を使う意義が薄れてしまう．

ところで「教材」の Python 版は JSON 形式で結果を返す API を使用しているが，VBA 版では CSV で返すものを用いている．VBA で JSON を解釈する方法はないわけではないようだが，64bit 版の Excel では VBA–JSON など外部ライブラリが必要になるようなので，このようにすることも仕方ないと思われる．

[*1] `pow(x,y)` は x の y 乗を計算する関数で，x が整数かつ y が非負整数のときは整数型の値を返すので，`pow(2,53)` と書くとオーバーフローしてしまう．

表 6　預金の複利計算のプログラム

```
riritsu ← 0.05
yokin ← [100000]
i を 1 から 10 まで 1 ずつ増やしながら繰り返す：
    risoku ←整数 (yokin[i-1]*riritsu)
    yokin[i] ← yokin[i-1]+risoku
yokin を表示する
線グラフ描画 (200,200,[yokin])
```

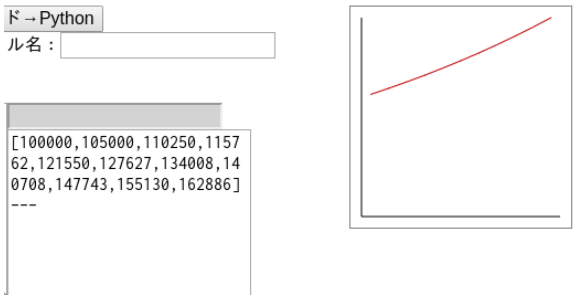

図 6　預金の複利計算のグラフ

4.5　学習 15　アルゴリズムの比較

学習 15 では探索やソートのアルゴリズムが扱われているが，これも基本的なものなので問題ない．

4.6　学習 16　確定モデルと確率モデル

学習 16 はいろいろなシミュレーションを扱っている．計算はどの環境でもできるが，グラフでの表現については JavaScript では難しいとして「教材」では扱われていない．しかし JavaScript でも `plotly.js` などを使うことでグラフの作成が容易にできることは [15] に述べられている通りである（学習 17 に関しても同様）．

PyPEN ではもともとあったグラフィック命令に加えて，配列を棒グラフや折れ線グラフにする機能を用いれば描画は容易にできる．たとえば預金の複利計算のプログラムは**表 6**，実行結果は**図 6** になる．

散布図については直に描画しなくてはいけないので，少し面倒になる．たとえばモンテカルロ法で円周率の近似値を求めるプログラムは**表 7**，実行結果は**図 7** になる．PyPEN のグラフィックでは左上が原点で y 軸が下向きになっているのでこのような表示になっている．

表 7　散布図を作成するプログラム

```
描画領域開く (200,200)
count ← 0
i を 1 から 2000 まで 1 ずつ増やしながら繰り返す：
    x ← random(200)
    y ← random(200)
    もし x*x+y*y<40000 ならば：
        count ← count+1
        塗色設定 (255,0,0)
    そうでなければ：
        塗色設定 (0,0,255)
    円塗描画 (x,y,1)
「円周率：」と count*4/2000 を表示する
```

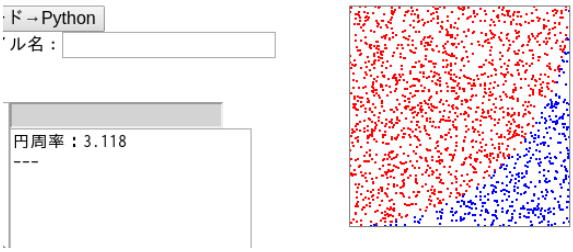

図 7　散布図のプログラムの実行結果

表 8　放物運動のプログラム

```
描画領域開く (100,100)
塗色設定 (255,0,0)
throw(45)
塗色設定 (0,0,255)
throw(60)
塗色設定 (0,255,0)
throw(30)

手続き throw(angle)：
    dt ← 0.01
    g ← 9.8
    v0 ← 30
    x ← 0
    y ← 0
    angle ← angle*3.1415926535/180
    vx ← v0*cos(angle)
    vy ← v0*sin(angle)
    i を 1 から 1000 まで 1 ずつ増やしながら繰り返す：
        vy1 ← vy-g*dt
        x ← x+vx*dt
        y ← y+(vy+vy1)/2*dt
        vy ← vy1
        円塗描画 (x,100-y,1)
        もし y<0 ならば：
            繰り返しを抜ける
```

4.7　学習 17　自然現象のモデル化とシミュレーション

学習 17 は自然現象を対象にしている以外は学習 16 と大差ない．たとえば放物運動については**表 8**，**図 8** のようになる．

4.8　学習 21　さまざまな形式のデータとその表現形式

学習 21 ではデータのさまざまな表現形式が紹介されている．「(4) キー・バリュー形式のデータの処理・蓄積」について Python でのコードが掲載されているが，これは PyPEN でもたとえば**表 9** のように実現できる．keys は連想配列のキーを並べた配列を返す関数である．

図 8　放物運動の結果

表 9　キー・バリューで扱うプログラム

```
atani["name"] ←"A 谷"
atani["homephone"] ←"0**-***-1111"
atani["mobilemail"] ←"ata@example.jp"
bkawa["name"] ←"B 川"
bkawa["mobilephone"] ←"0**-****-2222"
cyama["name"] ←"C 山"
cyama["mobilephone"] ←"090-****-3333"
cyama["email"] ←"cyama@example.com"
member ← [atani,bkawa,cyama]
i を 0 から length(member)-1 まで 1 ずつ増やしながら繰り返す：
    key ← keys(member[i])
    j を 0 から length(key)-1 まで 1 ずつ増やしながら繰り返す：
        key[j] と":"と member[i,key[j]] を表示する
    改行する
```

しかしこれについてはそのような構造のデータの入出力ができるということにつなげたいのであるから，Web API や File I/O を持っていない PyPEN では大した意味を持たない．

5.　考察

5.1　Web API に関して

前述したように，PyPEN はブラウザ上の JavaScript のプログラムであるから，Web API を使おうとしても CORS の制限にひっかかってしまう．回避策として考えられるのは Web API をバックエンドで操作するというものだが，それだと「ファイルを Web サーバに置けばそれだけで使える」という PyPEN のメリットを損なうことになる．もっとも，オプションとしてそれを付け加えることはできるかも知れないので，今後検討したい．

Google chrome や Firefox には，指定さえすれば Response ヘッダに `Access-Control-Allow-Origin:"*"` を追加できる拡張機能があって，これを使えばオリジンが違っても Web API へのアクセスが可能になるのだが，特定ブラウザの特定拡張機能の使用を前提とすることも現実的ではない．

5.2　File I/O に関して

PyPEN で連想配列的なものを実装したのは，たとえば JSON 形式のデータ構造をそのまま扱えるようにという思惑であった．Web API によってそれが行えないのであれば，ファイル読み込みによって実現することが考えられる．そのためにはブラウザ上でファイルを読み込む必要があるのだが，ローカルファイルの操作には制限がある．これに

ついても Web ストレージを用いるなどの工夫が必要であろう．

5.3 DNCL やフローチャート，Python との関係について

PyPEN の言語は DNCL を元にしているとはいうものの，いくらか違う部分があるので独自言語と言えなくもない．たとえば入力も，PEN の xDNCL と違って型を指定することにしている（変数が型情報を持たないため）．そもそも元々の DNCL には入力そのものがない．

また Python に似せた言語といいつつも，DNCL 由来の書き方のままになっているところもあるので中途半端であることは否めない．たとえば割り算の`/`，`//`の使い分けは Python3 を真似たものになっているが，代入は`=`でなく←であり，等値演算子は`==`でなく`=`である．これが Python などを知っている学習者にとってはかえって混乱を生むことも考えられる．

そのようなことから，xDNCL と違ってセンター試験の問題が（ほとんど）そのまま入力できるわけでもないのだから，DNCL 系の書き方に必ずしも縛られる必要はないのではないかとも考えている．たとえば演算子を完全に Python にそろえてしまうことも考えられるし，配列に対する `push/pop` に相当するものや `for`〜`in` のような拡張 for 文など適切な日本語表現が思いつけば早く実装したいものもある．

一方 DNCL や xDNCL にあるのに今まで実装してこなかった構文，たとえば「`a += 1`」「`else-if`」「`switch`」に相当するものなどがある．これらについてはフローチャートでの表現が難しくなることを理由に実装してこなかったが，逆にそれを制約とも感じている．フローチャートをより充実させてこれらの表現ができるようにするべきなのか，それともフローチャートをある程度あきらめてしまうのか，なかなか決断できないでいる．

6. おわりに

「教材」で取り上げられている各学習のプログラムの多くは PyPEN で実現できることがわかったが，現状できていないものについては対応することは難しい．しかし実現すれば有益なものであるとも考えているので，今後の開発において検討を進める予定である．実際，本稿をまとめるにあたって新たに実装した機能も以下の通りいくつかある．

- 簡単な棒グラフ・折れ線グラフを簡単に描けるようにした．
- 配列の添字に文字列を使えるようにした．
- 連想配列のキーの配列を取得する関数 `keys` を実装した．

高校現場では教科書にどの言語が取り上げられるかについて興味が集まっており，実際に教科書に掲載されると，多くの学校ではその環境で実習を行うことがほぼ確定してしまうだろう．「教材」と同じように別言語版の「差分」が発行されれば，利用する側としては選択肢が増えることになると期待するのだが，そういうことにはならないだろうか．

最後に，PyPEN は筆者のサイト[*2]および GitHub で配付しているが，更新が容易であるという点で GitHub を推奨したい．

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

の実行後，`PyPEN` フォルダにある拡張子 `js-dist` のファイルを拡張子 `js` になるようにコピーすればよい．更新は `PyPEN` フォルダで

```
git pull
```

するだけである．

勤務校では PyPEN を用いた授業実践が始まりつつあるので，その結果についても検証を行いたいと考えている．授業の準備段階での PyPEN のバグ発見にあたって，多くの助言をいただいたことについて井上智香子氏に感謝する．

参考文献

[1] 文部科学省：高等学校情報科「情報 I」教員研修用教材 (2019).
[2] 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 回研究会報告書，日本情報科教育学会，pp. 19–22 (2019).
[3] 文部科学省：高等学校情報科「情報 I」教員研修用教材第 3 章他プログラミング言語版，（オンライン），入手先 ⟨http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1421808.htm⟩（参照 2019-11-25）.
[4] 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).
[5] 本多佑希，兼宗　進：ブラウザ上で動作する DNCL 学習環境「どんくり」の開発，研究報告コンピュータと教育 (CE)，Vol. 2018-CE-147, No. 10 (2018).
[6] 大門　巧，大西建輔：ブラウザ上で動作する DNCL 処理系「Tetra」の開発，研究報告コンピュータと教育 (CE)，Vol. 2019-CE-151, No. 7 (2019).
[7] 大宮大地，松本嵩大，松浦敏雄，中西通雄：試験問題作成機能と学習及び受験用機能を持つ DNCL プログラミング環境，研究報告コンピュータと教育 (CE)，Vol. 2019-CE-148, No. 7 (2019).
[8] 長　慎也，長島和平，間辺広樹，兼宗　進，並木美太郎：オンラインプログラミング環境 Bit Arrow における Python 処理系，情報教育シンポジウム論文集，Vol. 2019, pp. 122–129 (2019).
[9] ギノ株式会社：paiza.io, (online), available from ⟨https://paiza.io/ja⟩ (accessed 2020-01-10).
[10] 新田章太，小西俊司，竹内郁雄：複数言語に対応しやすいオンラインプログラミング学習・試験システム track，情報教育シンポジウム論文集，Vol. 2019, pp. 114–121 (2019).
[11] @melponn，@kikairoya：Wandbox, (online), available from ⟨https://wandbox.org/⟩ (accessed 2020-01-10).
[12] 小林史弥，白井詩沙香，山澤昭彦，兼宗　進：ドリトルでのデータ処理機能とグラフ描画機能の開発，情報教育シンポジウム論文集，Vol. 2017, No. 27, pp. 178–181 (2017).

[*2] https://watayan.net/prog/pypen.html

表 10 「教材」各学習のプログラムについての対応表

タイトル		Python	JavaScript	VBA	PyPEN
学習 11	オーバーフローの発生	○	○	○	○
コンピュータの仕組み	コンピュータの計算誤差	○	○	○	○
学習 12	順次	○	○		
外部装置との接続	分岐	○	○		
	反復	○	○		
学習 13	順次の例	○	○	○	○
基本的プログラム	分岐の例	○	○	○	○
	反復の例	○	○	○	○
	分岐と反復	○	○	○	○
学習 14	リスト・配列の例	○	○	○	○
応用的プログラム	乱数の例	○	○	○	○
	関数で分割したプログラムの例	○	○	○	○
	WebAPI の例	○		○	
学習 15	線形探索の例	○	○	○	○
アルゴリズムの比較	二分探索の例	○	○	○	○
	選択ソートの例	○	○	○	○
	クイックソートの例	○	○	○	○
学習 16	預金の複利計算	○	○	○	○
確定モデルと確率モデル	同グラフ化	○		○	○
	乱数発生	○	○	○	○
	サイコロのプログラム	○	○	○	○
	同度数分布	○		○	○
	円周率	○	○	○	○
	散布図を作成	○		○	○
学習 17	放物運動	○		○	○
自然現象のモデル化と	個体の増加	○		○	○
シミュレーション	ランダムウォーク	○		○	○

[13] 小林史弥，本多佑希，白井詩沙香，兼宗　進：オンライン版ドリトルを用いたデータ分析学習環境の開発，情報教育シンポジウム論文集，Vol. 2018, No. 16, pp. 112–117 (2018).
[14] 兼宗　進，本多佑希：高等学校「情報 I」の研修資料におけるプログラミングの言語の扱い，情報教育シンポジウム論文集，Vol. 2019, pp. 168–175 (2019).
[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，研究報告コンピュータと教育 (CE)，Vol. 2019-CE-151, No. 3 (2019).
[16] クジラ飛行机：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 2020-01-12）.
[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] The Qt Company: Qt Cross–platform software development for embedded & desktop, (online), available from ⟨https://www.qt.io/⟩ (accessed 2019-05-29).
[19] MDN contributors: オリジン間リソース共有 (CORS)，MDN Web Docs（オンライン），入手先 ⟨https://developer.mozilla.org/ja/docs/Web/HTTP/CORS⟩（参照 2020-01-13）.