プログラミング学習環境 PyPEN の開発

中西渉

watayan@meigaku.ac.jp
名古屋高等学校

2019 年 11 月 2 日

1 はじめに

DNCL を用いたプログラミング学習環境の開発

- PenFlowchart (2011)
- WaPEN (2017)

Python の流行に合わせて... DNCL を Python 風にアレンジ

- PyPEN (2019)

2.1 DNCL の実行環境

DNCL

- センター試験「情報関係基礎」の擬似言語
- BASIC や C を日本語に直訳した感じ
  - 言霊，なでしことは違う
    ↑日本語の文としての自然さ
  - ドリトルとも違う
    ↑オブジェクト指向言語「らしい」記述

言霊

歩幅を整数型とする。
歩幅に 100 を入れる。
4 回だけ{歩幅だけ進む。右に 90 度回転する。}を行う。

なでしこ

年齢は 2018 から 1966 を引く。
もし年齢が 20 以上ならば
　　「成年」と表示。

DNCL

```
a ← 2019-1964
もし  a=17  ならば
  |   「おいおい」を表示する
を実行する
```

PEN

- DNCL を拡張した xDNCL
- 大阪学院大学・大阪市立大学で開発

PenFlowchart

- フローチャートでプログラムを作成する機能を追加
- xDNCL を採用（実行部分は PEN そのまま）
- 筆者が開発 (2017)

2.2 Web アプリケーションとしての実装

PEN，PenFlowchart，ドリトルは Java アプリケーション

- 複数の OS で動く
- インストールに制限がある学校も...

→Web アプリケーション化

PenFlowchart→WaPEN
ドリトル →Bit Arrow

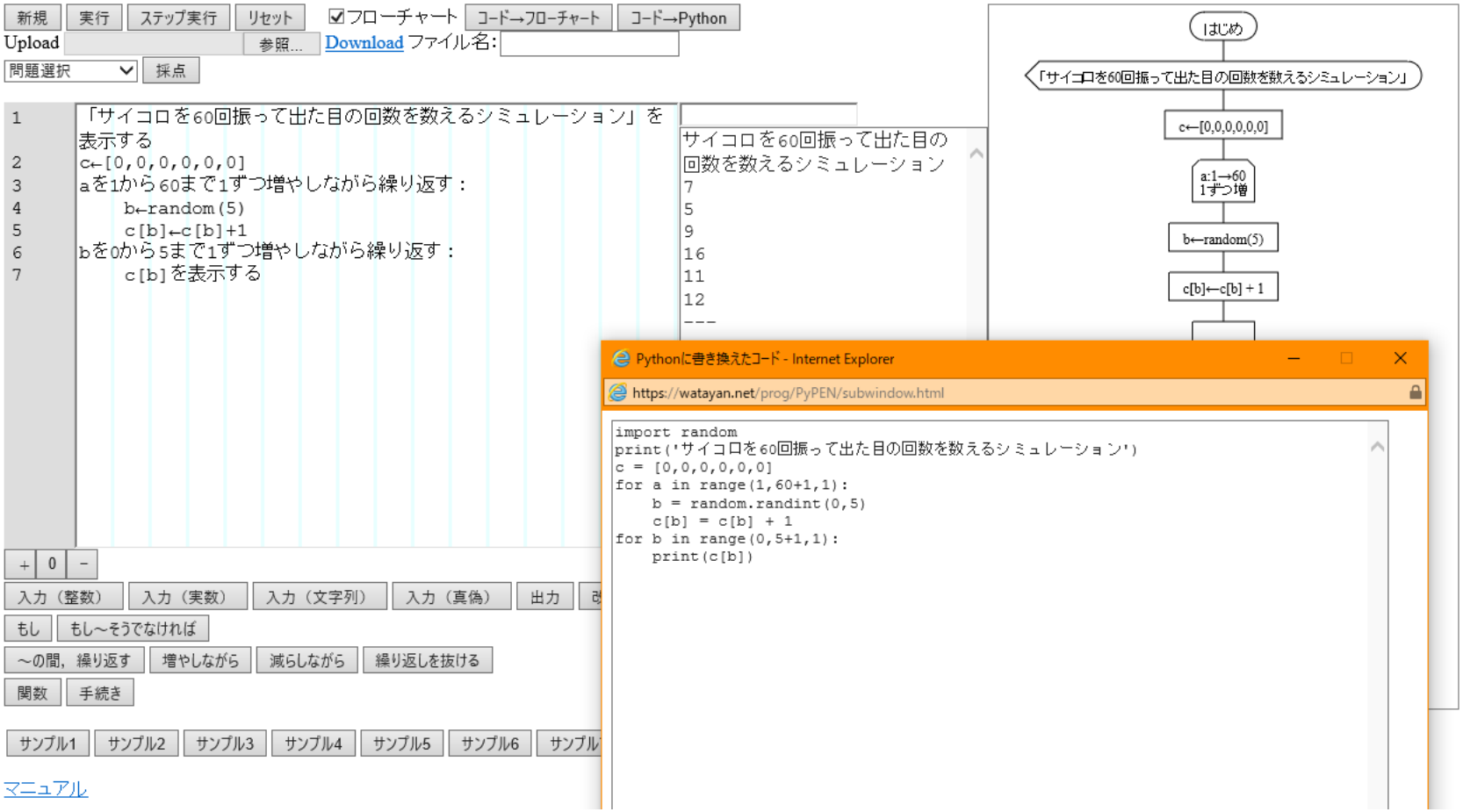
Web ブラウザ上で動く DNCL 環境（のうち筆者が知ってるもの）

- WaPEN
- どんくり（Bit Arrow）
- WaPEN
- Tetra

3.1 概要

- Web ブラウザ上で動作するプログラミング学習環境
- DNCL を Python 風にアレンジした独自言語
- コード ↔ フローチャート
  （自作関数・手続きには未対応）

動作画面



- 入力支援ボタン
- 自動インデント

3.2 動作環境

動作確認：

- Internet Explorer 11
- Microsoft Edge
- Google Chrome
- Firefox
- Safari

HTML，CSS，JavaScript だけで実装

- Web サーバには特別な設定不要
  データベースも PHP もいらない
- ローカルファイルで実行も可能

3.3 Python‘風’ 言語

Python 風といっても...

- 命令ブロックをインデントで表現
- ブロック直前の行末に「：」
- 変数宣言の廃止
- 除算演算子を Python に合わせた
  - `/` 余りを出さない割り算（商は実数）
  - `//` 余りを出す割り算（商は整数）

PyPEN と DNCL の比較

PyPEN

もし～ならば：
　　命令
そうでなければ：
　　命令

DNCL

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

PyPEN

～の間繰り返す：
　　命令

DNCL

～の間，
　｜　命令
を繰り返す

PyPEN と Python の比較

PyPEN

もし～ならば：
　　命令
そうでなければ：
　　命令

Python

```
if ～:
    命令
else:
    命令
```

PyPEN

～の間繰り返す：
　　命令

Python

```
while ～:
    命令
```

日本語の文として読み下せるか

- 「命令」を塊として読めるか
- 命令は用言で終わる「文」

「もし～ならば，～を表示する，を実行する」を許容するか

- 「を実行する」（END IF に相当）を削除するとエラー

変数宣言の廃止

- 代入するときに変数生成
- 値は型（整数・実数・文字列・真偽）を持つ
- 変数は型を持たない

xDNCL の入力

a を入力する

PyPEN の入力

a に整数を入力する

関数・手続き

- 関数と手続きを区別する（Python と異なる）

  関数　値を返す
  　　命令や計算式の中で呼ばれる
  手続き　値を返さない
  　　それ自体が命令

関数の宣言

関数　関数名（引数,…）：
　　命令
　　～を返す

手続きの宣言

手続き　手続き名（引数,…）：
　　命令

※発表原稿では行末の「：」が抜けていました

3.4 サンプルプログラム

サンプルボタン → 用意したサンプルが入力される
sample.js に予め書いておく
※自分の授業に合わせたサンプルに差し替えてほしい

sample.js の例

```
"use strict";
var sample=[
"a を 1 から 10 まで 1 ずつ増やしながら繰り返す：\n"+
"    b を 1 から a まで 1 ずつ増やしながら繰り返す：\n"+
"        「☆」を改行なしで表示する\n"+
"    「」を表示する"

,
"a ← 0\n"+
"b ← random(8)+1\n"+
...
];
```

しかし，これを手で直すのは...対応策は 3.7 で

3.5 自動採点機能

問題を表示
いくつかのテストケースで正誤を判定
answer.js に次の要素を書く

1. タイトル
2. 問題文
3. 入力値（値の配列）の配列
4. 期待される出力結果の配列
5. 出力終了までの制限時間（ミリ秒：省略時は 1000）

新規　実行　ステップ実行　リセット　* ☑ フローチャート　コード→フローチャート　コード→Python

Upload　ファイルを選択　選択されていません　Download ファイル名：

Q6:曜日の計算　採点

3つの整数（西暦年，月，日）を受け取って，
その日の曜日を番号（0:日曜，1:月曜，…，6:土曜）で表示しなさい。

```
1  yに整数を入力する
2  mに整数を入力する
3  dに整数を入力する
4  j←y//100
5  k←y%100
6  h←d+(13*m+8)//5+k+k//4+j//4+5*j
7  h%7を表示する
8
```

```
*** 採点開始 ***
ケース1...成功
ケース2...成功
ケース3...失敗
結果が違います。
ケース4...失敗
結果が違います。
ケース5...成功
ケース6...失敗
結果が違います。
ケース7...成功
ケース8...成功
ケース9...失敗
結果が違います。
--- 不合格 ---
```

＋　0　－

想定している場面

× 採点の手間を省く

○ 生徒が自分で正しいことを確認する

自分で実行ボタンを押せない生徒がいる

生徒 先生，このプログラム合ってますか？

教師 実行してみてどうだった？

生徒 まだ実行してません

教師 実行すればわかるんだからやってごらん

生徒 ...

そのハードルが下がるといいのだが...

answer.js の例

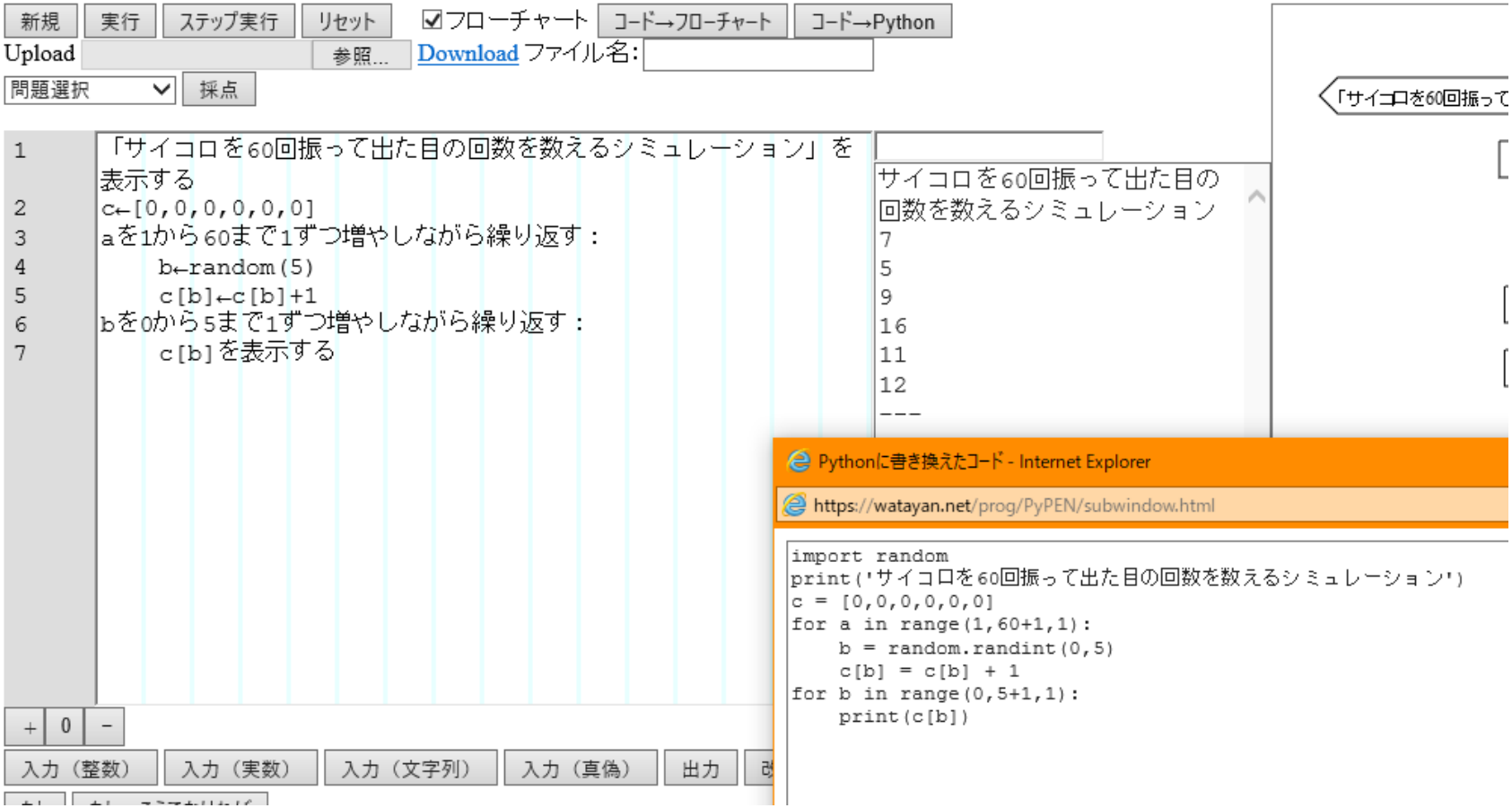
```
"use strict";
var Quizzes=[
    new Quiz(’敬称をつける’,’文字列を 1 つ受け取って，それの後ろに「さん」をつけて表示しなさい。’,
    [[12],[’わたやん’],[’さかなクン’],[’’]],
    [’12 さん’,’わたやんさん’,’さかなクンさん’,’さん’],100),
    new Quiz(’最大公約数’, ’2 つの整数を受け取って，最大公約数を表示しなさい。’,
    [[35,21],[21,35],[24,25],[23,23],[19661220,20190528],[527
    [7,7,1,23,36,1]),
…];
```

これを手で直すのも...対応策は 3.7 で

3.6 Python コードの出力

「コード→ Python」ボタン→ Python に変換したコードを表示

- コピーして Python で実行できる
- 必要なモジュールは import 命令も付加
- 関数・手続きの宣言は冒頭に移動
- 未対応要素（グラフィック命令など）があると変換できない

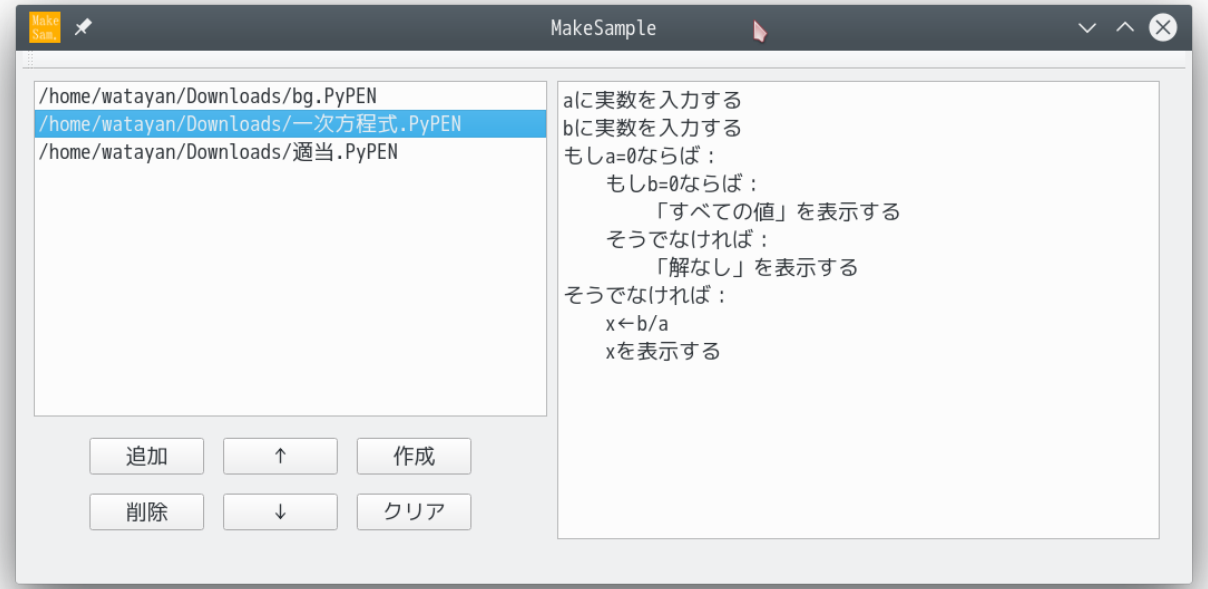

新規
実行
ステップ実行
リセット
フローチャート
コード→フローチャート
コード→Python
Upload
参照...
Download
ファイル名:
問題選択
採点
1 「サイコロを60回振って出た目の回数を数えるシミュレーション」を表示する
2 c←[0,0,0,0,0,0]
3 aを1から60まで1ずつ増やしながら繰り返す：
4 b←random(5)
5 c[b]←c[b]+1
6 bを0から5まで1ずつ増やしながら繰り返す：
7 c[b]を表示する
サイコロを60回振って出た目の
回数を数えるシミュレーション
7
5
9
16
11
12

「サイコロを60回振って
+
0
−
入力（整数）
入力（実数）
入力（文字列）
入力（真偽）
出力
Pythonに書き換えたコード - Internet Explorer
https://watayan.net/prog/PyPEN/subwindow.html
import random
print('サイコロを60回振って出た目の回数を数えるシミュレーション')
c = [0,0,0,0,0,0]
for a in range(1,60+1,1):
b = random.randint(0,5)
c[b] = c[b] + 1
for b in range(0,5+1,1):
print(c[b])

3.7 MakeSample, MakeAnswer

sample.js，answer.js の書き直し...手作業ではやりたくない
→MakeSample，MakeAnswer の開発

- 個々のサンプルや問題・解答ファイルを集めてそれらを生成
- Qt で開発→複数の OS に対応
  - 「Web アプリにしたら？」という提案



4 考察

なでしことの比較

- インデントでブロック…同じ
- 日本語としての自然さ…なでしこが上

PyPEN の「売り」は？

- フローチャート
- Python への「直訳」

本格言語への足がかり
フローチャート →PyPEN のコード →Python のコード

Python の魅力は豊富なライブラリにあるのでは？

- だったら Python を使うべき
- PyPEN で Deep Learning とかありえない

Bit Arrow の Python ではサーバモードでライブラリ使えるが [13]

- サーバとの連携が必要
- 悪意あるプログラムの排除が難しそう

(13) 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
情報処理学会情報教育シンポジウム論文集，Vol.2019，No.18，pp.122-129

DNCL にこだわる必要はあるか

- 既に改変している
- もっと Python 寄りでいいのでは
  - for in 構文とか使いたい

  → どんな表現にする？
- 自然な日本語 かつ Python に直結
  でも完全に同じにはならない
  - 変数のスコープとか

5 おわりに

勤務校では 3 学期に授業を実施
→DNCL での授業との比較・検討を行いたい

PyPEN は筆者のサイトと GitHub で公開（GitHub がお薦め）

インストール

```
git clone https://github.com/watayan/PyPEN.git
cd PyPEN
copy sample.js-dist sample.js
copy answer.js-dist answer.js
```

更新

```
git pull
```

不具合・改善点など，ご意見をください