

# 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

中西 渉<sup>1,a)</sup>

**概要:** 大学入試センター試験「情報関係基礎」で用いられるプログラミング言語 DNCL の, Web ブラウザ上での実行環境を JavaScript で開発した. さらにフローチャートでのプログラム作成環境などを整えることにより, 拙作である PenFlowchart の代替になりえるものになったと考えている. これを WaPEN(Web-aided PEN) と名付けて公開を開始した.

**キーワード:** プログラミング教育, DNCL, フローチャート, HTML5

## WaPEN... Implementation of Flowchart etc. in the Execution Environment of DNCL on the Web Browser

WATARU NAKANISHI<sup>1,a)</sup>

**Abstract:** I have developed an execution environment of DNCL on the web browser with JavaScript. By adding the flowchart and so on, I think that it became a substitute for PenFlowchart which is my work. I have named it WaPEN (Web-aided PEN) and started publishing.

**Keywords:** Education of programming, DNCL, Flowchart, HTML5

### 1. はじめに

2022 年度から施行される高等学校新学習指導要領 [1] が公開され, 教科「情報」は単元「コンピュータとプログラミング」を含む「情報 I」が必修となることが事実となった. 現行の学習指導要領ではプログラミングが明示的には含まれていない「社会と情報」を選択している高校が約 8 割であると言われており, 新学習指導要領への対応に苦慮しているところがあるものと思われる.

筆者は勤務校での「情報 B」「情報の科学」の授業におけるプログラミングの実習で 2006 年度から PEN[2] を用い, またそれにフローチャートでプログラムを生成できる機能を付加した PenFlowchart[3] を 2011 年度に開発した. これらは大学入試センター試験「情報関係基礎」で用いら

れている擬似言語 DNCL[4] を拡張したプログラミング言語 xDNCL[5] でプログラムを作成することができる.

ところがこれらは Java アプリケーションであるため, 学校によっては導入できない場合があると聞いている (授業担当者の判断では新しいアプリケーションや JRE の導入が禁じられているなどのため). そこで Web ブラウザ上で動作する xDNCL の実行環境を作成して昨年の情報教育シンポジウムで発表した [6]. さらに開発を進め, PenFlowchart の代用として十分使えるものになったと考えられるので, これを WaPEN(Web-aided PEN) と名付けて公開した. 本稿ではこの WaPEN の概要について述べる.

### 2. 先行研究

#### 2.1 DNCL によるプログラミング学習環境

前述したように, DNCL を拡張した xDNCL の実行環境には PEN や PenFlowchart がある. PEN は大阪学院大学情報学部西田研究室と大阪市立大学大学院創造都市研究科

<sup>1</sup> 名古屋高等学校  
Nagoya Senior High School

<sup>a)</sup> watayan@meigaku.ac.jp

松浦研究室の共同プロジェクトとして作られた、初学者向けプログラミング学習環境である。単にプログラムの実行ができるだけでなく、入力支援ボタンや自動インデントなどのプログラム入力支援機能や、実行速度調整やステップ実行などの実行制御機能を備えている。PenFlowchartは、フローチャートによってプログラムを生成する機能を、筆者がPENに付け加えたものである。それぞれの実行画面を図1、図2に示す。

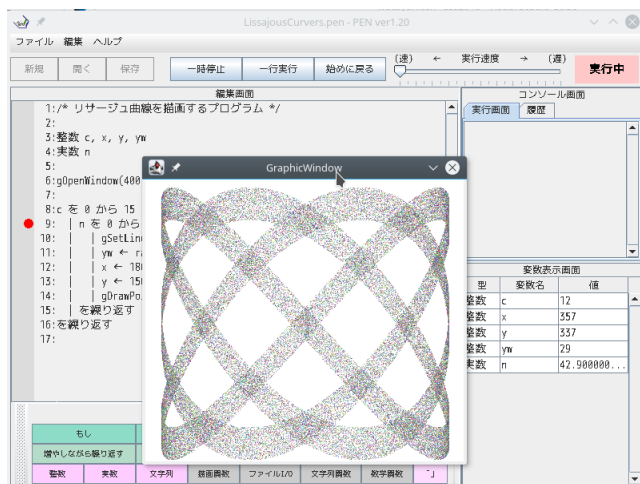


図1 PENの実行画面  
Fig. 1 Snapshot of PEN

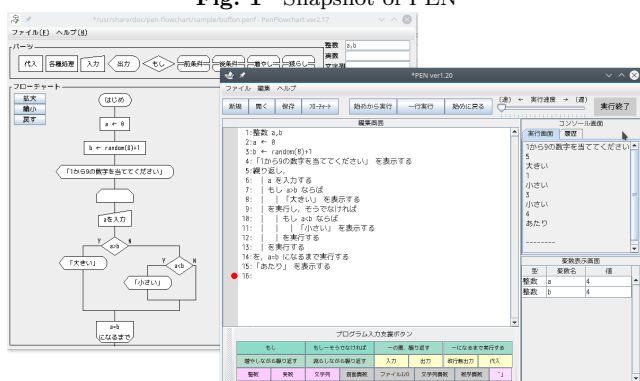


図2 PenFlowchartの実行画面  
Fig. 2 Snapshot of PenFlowchart

また、PENにはOpenBlocksを使用することによって、ビジュアルブロックによるプログラミングができるよう拡張したoPEN [7]などの実装もある。

## 2.2 チャートによるプログラミング

フローチャートや構造化チャートからプログラムを生成するシステムは、古くから存在している。たとえば塩澤らはフローチャートからVisual Basicのプログラムを生成するプログラム [8]を、斐品らはPADでプログラムが作成・実行できるJPADet [9]をそれぞれ開発しており、またFlowgorithm [10]のようにフローチャートのままプログラムを実行できるほか、十数種類の言語のプログラムを生

成することができるものもある。

また喜多らは逆にプログラムをフローチャートに変換して実行経路を確認できるAvis [11]を、今泉らはCで書かれたプログラムのネストの構造などを可視化するツールazur [12]を開発した。

筆者が開発したPenFlowchartやこの論文で公開するWaPENはそれらのシステムに対して後発であるが、以下の点で高校での実施についてはより良い点もあると考える。

- 実行環境をWindowsに限定しない。
- 現在出版されている教科書に記載されているフローチャートに近い表記である。

## 2.3 Webブラウザ上のプログラミング学習環境

ドリトル [13]は学校現場で広く使われているが、インストーラ版はPENやPenFlowchartと同様Javaアプリケーションであるため、同じような面倒が考えられる。そこで今はインストールせずに使えるオンライン版やBit Arrow版を用いることができるようになってきている。

Bit Arrow [14]はドリトル以外にもJavaScriptや簡易Cに対応しており、便利な構文が用意されているのでゲームなどが簡単に作れるようになってきている。作ったプログラムはサーバ上に保存され、容易に実行結果を確認できるだけでなく、そのアドレスをQRコードで出力することでスマートフォンなどの上で実行することもできるようになっている。

Scratchは校種を問わず広く用いられている。現在公開されているScratch 2.0はFlashアプリケーションであり、将来的にブラウザがサポートしなくなることが決まっているが、開発中のScratch 3.0はHTML5のアプリケーションである。

paiza.IO [15]はギノ株式会社が運営しているオンラインのプログラム実行環境で、20以上の言語に対応しており、ブラウザ上でコードを書いて実行結果を得ることができる。IDEにあるようなコード補完機能などもあり、いろいろな言語を試しに試してみるなどの用途にも適している。

## 3. WaPENの概要

### 3.1 プログラムの概要

WaPENはJavaScriptで実装したxDNCLの実行環境である。必要なファイルを配置すればWebブラウザだけで実行でき、Webサーバ上でデータベースなどの設定をする必要はない。実際にはWebサーバ上に置く必要もなく、ローカルマシンにファイルを展開しておいてindex.htmlをブラウザで開くだけでもよい。

図3に実行画面を示す。左の入力欄にコードを入力するのだが、画面下部の入力支援ボタンを使うと構文の雛形が入力され、ユーザは「変数」「値」などの部分を書き換えることでプログラムが出来上がる。また、入力欄で右ク

リックすることにより、定義済み関数などが入力できる。中央にあるのが実行結果の表示欄である。

右のフローチャート画面では、縦棒上で右クリックすることによって部品を追加し、各部品は右クリックメニューやダブルクリックで編集できる。フローチャートを編集するとただちにコードが修正されるが、コードを修正したときは「コード → フローチャート」ボタンを押さないとフローチャートに反映されない。

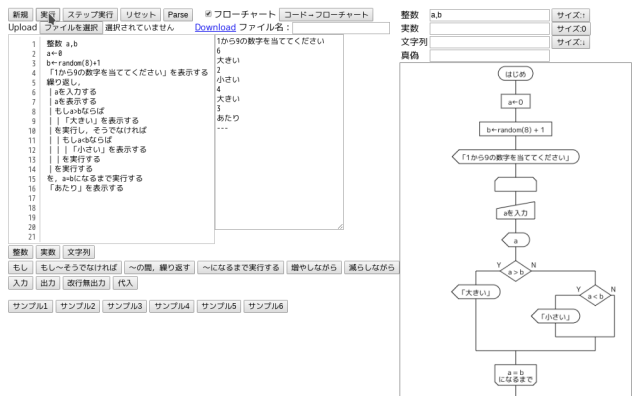


図 3 WaPEN の実行画面  
Fig. 3 Snapshot of WaPEN

画面下部には、設定したサンプルプログラムを呼び出すボタンを配置した。授業で動作だけを確認したいプログラムなどがあるとき、それをユーザに入力させる時間が節約できると考えたからである。

### 3.2 昨年公開時との差分

昨年の情報教育シンポジウム原稿執筆時点では、かろうじて動作する程度のものであったが、その後、以下の点を改良した。

- 変数宣言の必要・不要などを設定ファイルで切り替えられるようにした。
- フローチャートの実装。
- 無限ループが止められなかったのを、postMessage などを使うことで解決した [16]。
- ステップ実行をちゃんと 1 行ずつの実行になるように修正した。
- 配列を代入できるようにした。
- サンプルプログラムをプログラム本体から分離した。
- 読点の有無などの細かい間違いをある程度無視できるようにした。

### 3.3 動作確認

動作確認は表 1 にあげた環境で行った。開発は ES2015 で行っているが、対応していない Web ブラウザもあるので Babel でトランスパイルしている。

タブレット上の Web ブラウザでもプログラムの作成や

実行はある程度できるが、実用的だとは思えないし、作成したプログラムのセーブ・ロードにも難があるので、今後は考えないことにしたい。

表 1 動作確認したブラウザ  
Table 1 Operation check

Windows 10	Internet Explorer 11
	Microsoft Edge
	Google Chrome
	Mozilla Firefox
Mac OS Sierra	Safari
Debian GNU/Linux stretch	Google Chrome
	Mozilla Firefox

### 3.4 使用実績

2017 年度の夏に東京で行われた情報処理学会教員免許更新講習「情報教育と情報セキュリティ」で、平成 29 年度のセンター試験「情報関係基礎」の第 3 問を解く実習に WaPEN を用いた（このときはまだフローチャートを実装していなかった）。問題では図形のデータを延々と配列に入力する部分が省略されていたので、その部分をサンプルプログラムとして準備しておくことで、肝心の部分だけを作ることに専念できた。

勤務校の 3 学期の「情報の科学」の授業で、筆者の担当クラスでは WaPEN を用いて実習を行った。その他のクラスでは PenFlowchart を用いたが、使用したテキストが同じプリント教材であったので混乱はなかった。

## 4. 考察

### 4.1 フローチャートの実装

WaPEN のフローチャートの編集環境は PenFlowchart とはいくらか異なっている。パーツの上で右クリックして出てくるメニューで編集や削除をするのは同じであるが、パーツの追加は PenFlowchart では画面上部の一覧からドラック&ドロップするのだが、WaPEN では縦棒を右クリックして出てくるメニューから選択するようになっている。これは Web ブラウザの一画面にすべてを収めるためにパーツ一覧を削除したためである。また、カット&ペーストは WaPEN ではまだ実装していない。

前述した担当クラスの 1 つで 1 時間だけ PenFlowchart を使ってもらって感想を聞いたところ、どちらでも大差ないという意見が多かった。しかし「パーツの形で機能を覚えているので一覧があった方がいい」「カット&ペースト機能は欲しかった」という意見もあったので、対応を検討したい（メニューにアイコンを追加するなど）。

なお、PenFlowchart や WaPEN のフローチャートは何かの規格に正しく準拠したものではなく、独自の表記が混じっている。たとえば JIS[17] ではループ端にはループ名

がなくはないし、終了条件を書くのが決まりになっているが、WaPEN 等ではループ名はなく、継続条件を書くこともある (WHILE~WEND 型の場合)。また、JIS では分岐の出口が3つ以上であるような書き方が示されているので本当なら else-if や switch も表現できるので、適切な実装が思いつかないので WaPEN 等ではこれらをサポートしていない。FOR~NEXT 型のループにしても、JIS では「初期化・増分・終了条件」を書くことになっているが、WaPEN 等では独自の表記をしている。

## 4.2 DNCL への準拠

前述したように、PEN 等 (ここでは PEN, PenFlowchart, WaPEN をまとめて「PEN 等」と表記することにする) が用いている xDNCL は DNCL の拡張であるが、完全に互換しているわけではない。さらに WaPEN と PenFlowchart は PEN の xDNCL に若干の変更を加えている。具体的には、入力を「《変数》 ← input()」でなく「《変数》を入力する」にしていることや、else-if に相当する「そうでなくもし」や switch に相当する「《変数》の値に応じて」がないこと、関数や手続きを自分で作れないことである。そもそも DNCL には入力がないし、ループは FOR~NEXT 型と WHILE~WEND 型しかない (しかしセンター試験の問題には LOOP~UNTIL 型のループも登場している)。こういった構文の相違を表 2 にまとめておく。自作関数について DNCL では「?」となっているのは、問題に応じて予め定義された関数を使うことができるものの、その関数の作り方は決められていないからである。

表 2 構文の相違  
Table 2 Differences in syntax

	DNCL	PEN	WaPEN
入力	no	← input()	を入力する
else-if	yes	yes	no
switch	no	yes	no
LOOP~UNTIL	no	yes	yes
自作関数	?	yes	no
配列値を代入	no	no	yes

DNCL では変数は宣言なしで使うことができ、値を代入することで初期化される。PEN 等はデフォルトでは変数宣言が必須となっているが、PEN と WaPEN は設定ファイルの書き換えによって宣言なしで変数が見えるようになる\*1。また数値の整数型・実数型の違いについて DNCL の仕様書には何の言及もない\*2。しかし PEN 等ではこれを区別することにしている。たとえば 11.0/2\*2 は 11.0 であるが、11/2\*2 は 10 だという具合である。

\*1 PenFlowchart ではできない。

\*2 除算では「/」と「÷」が区別されており、後者は商の整数部分とされているが、前者については説明がない。

DNCL の仕様では「論理演算子に優先順位はなく、左側の論理演算が先に実行されますが、丸括弧『 ( ) 』で、演算の順序を指定することができます」となっている。つまり「FALSE かつ FALSE でない」は「(FALSE かつ FALSE) でない」と解釈して TRUE にならなくてはいけないのだが、PenFlowchart や WaPEN では「でない」の優先順位を高く評価して FALSE と判断していた\*3。WaPEN は取り急ぎこの問題を「修正」したが、これは C などのプログラミング言語を経験している多くのプログラマにとって違和感のあるものだと考えられる。

これらの点から、DNCL への完全な準拠に縛られることには疑問を感じる。センター試験の問題に対応できることは必要だが、それ以外の部分は自由に考えていいのではないだろうか。現に PenFlowchart や WaPEN は「a ← {1,2,3}」のような配列の代入を「勝手に」サポートしている。たとえばドリトルが統計機能を実装したように [18]、同じような発展を PEN 等がしてもいいと考えている。

なお、センター試験の平成 28~30 年度の情報関係基礎の問題 [19] の第 3 問について、WaPEN で実行できるかを検討し、本試験の問題については些細な変更で対応できることを確認している。実は試験問題に記載されているプログラムは、データがあらかじめ配列に代入してあるという前提で書かれているので、実行できるプログラムを作るためにはその初期化部分を別途作らなくてはいけない。その点では配列の値を代入できるようにしたことやサンプルボタンの実装がメリットとなると考えている。

## 5. おわりに

現在発行されている高校の情報の教科書では DNCL を用いているものは一つもない。そこで DNCL を用いて授業を行うためには独自にテキストを準備しなくてはならない。そのこともあってか、重田らが 2014 年から 2015 年にかけて行ったアンケート [20] によれば、指導されているプログラミング言語として DNCL をあげている高校はほぼ (?) 皆無であった。

このような状況にも関わらず DNCL に関わる開発を進めることに不安を感じないではない。しかし、JavaScript は HTML に関係する知識が必要になることがあるし、VBA では Excel について同様のことが言える。DNCL は純粋にプログラミングだけを考えられる環境として有効なものと言えないのではない。

筆者は WaPEN を素材として使うことを考えている。WaPEN は HTML5 で出来ているので、たとえばバックエンドに課題提出などの仕組みを作ることも容易であるし、実行部分に手を入れれば自動テストを行うことも可能であろう。実際、Blockly を用いてビジュアルブロックに

\*3 この点について指摘をいただいた佐久間拓也氏に感謝する。

よるプログラミングができるように WaPEN を拡張した BlocklyPEN が、大阪工業大学の岩崎氏によって作られた。

WaPEN は筆者の Web サイト\*4と GitHub\*5で公開している。

#### 参考文献

- [1] 文部科学省：学習指導要領等，（オンライン），入手先 [http://www.mext.go.jp/a\\_menu/shotou/newcs/1384661.htm](http://www.mext.go.jp/a_menu/shotou/newcs/1384661.htm)（参照 2018-06-02）。
- [2] 中村亮太：初学者向けプログラミング学習環境 PEN，（オンライン），入手先 <https://pen.moe.hm/>（参照 2018-06-02）。
- [3] 中西 渉：PenFlowchart，（online），available from <https://watayan.net/prog/#penflowchart>（accessed 2018-06-03）。
- [4] 独立行政法人大学入試センター：センター試験用手順記述標準言語（DNCL）の説明，（オンライン），入手先 <http://www.dnc.ac.jp/albums/abm.php?f=abm00004841.pdf&n=H23.dncl.pdf>（2011）。
- [5] 中村亮太：xDNCL 言語マニュアル，（オンライン），入手先 <https://pen.moe.hm/#lmanual/xdncl.md>（参照 2018-06-02）。
- [6] 中西 渉：DNCL の Web 上での実行環境の開発，情報教育シンポジウム論文集，Vol. 2017，No. 23，pp. 160–162（2017）。
- [7] 主原佑記，赤井昭仁，中村亮太，松浦敏雄：OpenBlocks を用いたプログラミング学習用ソフトウェアの開発，情報処理学会研究報告 コンピュータと教育（CE），Vol. 2014-CE-124，No. 9（2014）。
- [8] 塩澤哲時，岡部建次：フローチャートから Visual Basic プログラムコードを作成するプログラムの作成，情報処理学会研究報告 ソフトウェア工学（SE），Vol. 2001-SE-134，No. 99（2001）。
- [9] 斐品正照，徳岡健一，河村一樹：構造化チャートを用いたアルゴリズム学習支援システム，情報処理学会論文誌，Vol. 45，No. 10，pp. 2454–2467（2004）。
- [10] Cook, D.: Flowgorithm，（online），available from <http://www.flowgorithm.org>（accessed 2018-06-02）。
- [11] 喜多義弘，片山徹郎，富田重幸：プログラム自動可視化ツール Avis を利用した結合テスト実施のための実行経路抽出手法の提案，情報処理学会論文誌，Vol. 51，No. 9，pp. 1859–1872（2010）。
- [12] 今泉俊幸，橋浦弘明，松浦佐江子，古宮誠一：ブロック構造の可視化によるプログラミング学習支援環境 azur ～関数の動作の可視化～，情報処理学会研究報告 ソフトウェア工学（SE），Vol. 2010-SE-167，No. 11，pp. 1–7（2010）。
- [13] 兼宗 進：プログラミング言語「ドリトル」，（オンライン），入手先 <http://dolittle.eplang.jp/>（参照 2018-06-04）。
- [14] 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），（オンライン），入手先 <http://bitarrow.eplang.jp>（参照 2017-05-24）。
- [15] ギノ株式会社：paiza.IO，（online），available from <https://paiza.io>（accessed 2018-06-04）。
- [16] Baron, D.: setTimeout with a shorter delay，（online），available from <https://dbaron.org/log/20100309-faster-timeouts>（accessed 2018-06-03）。
- [17] 日本規格協会：JIS X 0121:1986 情報処理用流れ図・プログラム網図・システム資源図記号（1986）。
- [18] 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）。
- [19] 大学入試センター：過去 3 年分の試験問題，独立行政法人 大学入試センター（オンライン），入手先 <http://www.dnc.ac.jp/center/kakomondai.html>（参照 2018-07-15）。
  - [20] 重田桂子，植原啓介，村井 純：高校教科「情報」に関するアンケート調査と分析，情報教育シンポジウム論文集，Vol. 2015，No. 5，pp. 31–38（2015）。

\*4 <https://watayan.net/prog/#wapen>

\*5 <https://github.com/watayan/WaPEN.git>