

# Web ブラウザ上のプログラミング学習環境 WaPEN における関数・手続きの実装

中西渉

watayan@meigaku.ac.jp  
名古屋高等学校

全国高等学校情報教育研究会 和歌山大会  
2019年8月11日

- ① はじめに
- ② DNCL の実行環境
  - PEN, PenFlowchart
  - WaPEN
  - どんくり
- ③ WaPEN への関数・手続きの実装
- ④ おわりに

## ① はじめに

## ② DNCL の実行環境

- PEN, PenFlowchart
- WaPEN
- どんくり

## ③ WaPEN への関数・手続きの実装

## ④ おわりに

# はじめに

- 必修修教科「情報」(2003～) ... 科目は選択
  - 情報 A ・ 情報 B ・ 情報 C
  - 社会と情報 ・ 情報の科学

プログラミングは避けられがち

- 必修修科目「情報 I」(2022～)
  - (1)「情報社会の問題解決」
  - (2)「コミュニケーションと情報デザイン」
  - (3)「コンピュータとプログラミング」
  - (4)「情報通信ネットワークとデータの活用」

高校生全員がプログラミング

## 勤務校の話

- プログラミング学習を実施（2004～）
- PEN を使用（2006～）
- PEN を補助するツールの開発
  - PenFlowchart（2011～）
  - WaPEN（2018～）

## 本発表の要旨

- WaPEN に関数・手続きを実装
- 細かい改良
- 自動採点機能

## ① はじめに

## ② DNCL の実行環境

- PEN, PenFlowchart
- WaPEN
- どんくり

## ③ WaPEN への関数・手続きの実装

## ④ おわりに

# DNCL の実行環境

DNCL…センター試験「情報関係基礎」作問用擬似言語

## DNCL のプログラム例

```
a ← 24
b を 1 から a まで 1 ずつ増やしながら、
| もし  $a \% b = 0$  ならば
| | b を表示する
| を実行する
を繰り返す
```

# PEN, PenFlowchart

**PEN** 大阪で開発された初学者向けプログラミング学習環境

**PenFlowchart** PEN にフローチャートをくっつけたもの

Java アプリケーション

→ 多くの OS で動く

→ 最近では敬遠される傾向

自分の思うソフトウェアがインストールできないことも

## PEN の画面

\*JanKenGame.pen - PEN ver1.20

ファイル 編集 ヘルプ

新規 開く 保存 始めから実行 一行実行 始めに戻る (速) ← 実行速度 → (遅) 実行終了

編集画面

```

1:整数 a,n
2:n←1
3:a を 1 から 10 まで 1 ずつ増やしながら,
4: | n←n*a
5: | aと「!」とnを表示する
6:を繰り返す

```

コンソール画面

実行画面 履歴

```

5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800
-----

```

変数表示画面

型	変数名	値
整数	a	11
整数	n	3628800

プログラム入力支援ボタン

もし	もし〜そうでなければ	そうでなくもし〜	〜の間、繰り返す
増やしながら繰り返す	減らしながら繰り返す	入力	出力
整数	実数	文字列	数値関数
		ファイルI/O	文字列関数
			数字関数
			「J」

# PenFlowchart の画面

The image shows two windows from the PenFlowchart software. The left window, titled '\*PenFlowchart ver2.17', displays a flowchart for calculating the sum of integers from 1 to n. The flowchart starts with 'はじめ' (Start), followed by 's ← 1', a loop 's: 1 → 18 1ずつ増' (s: 1 to 18, increase by 1), then 'n ← s \* n', and finally 'sと「1+」とn' (s and '1+' and n), ending with 'おわり' (End). The right window, titled '\*PEN ver1.20', shows a program editor with the following code:

```

1: 整数 a,n
2: n ← 1
3: a を 1 から 18 まで 1 ずつ増やしながら,
4: | n ← n*a
5: | sと「1+」とn を表示する
6: を繰り返す
7:

```

Below the code editor is a table of program input/output buttons:

プログラム入力変換ボタン			
もし	もし〜そうでないれば	〜の値、繰り返す	〜になるまで実行する
増やしなると繰り返す	減らしなると繰り返す	入力	出力
実行	実行	実行	実行
実行	実行	実行	実行

# WaPEN

## PenFlowchart とほぼ同じ環境を Web ブラウザ上で実現 Web-aided PEN

The screenshot displays the WaPEN web interface. On the left, there is a code editor with a file named 'a.pn'. The code is as follows:

```

1  整数 a,b
2  a=1
3  bを1から100まで1ずつ増やしなが
4  | a-a*b
5  | bと「!」とaを表示する
6  を繰り返す
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

```

Below the code editor are several control buttons: 整数, 実数, 文字列, 入力, 出力, 改行無出力, 代入, もし, もし〜そうでなければ, ~の間, 繰り返す, ~になるまで実行する, 増やしなが, 減らしなが, 関数, 手続き.

On the right, there is a flowchart with the following steps:

```

graph TD
    Start([はじめ]) --> A1[a=1]
    A1 --> B100[b:1→100  
1ずつ増]
    B100 --> Calc[a=a*b]
    Calc --> Print([bと「!」とa])
    Print --> End([おわり])

```

At the top right, there are input fields for 整数 (a,b), 実数, 文字列, and 真偽, each with a 'サイズ' (Size) dropdown menu.

At the bottom left, there is a 'マニュアル' (Manual) link and a row of buttons labeled サンプル1 through サンプル8.

# どんくり

Bit Arrow... Web ブラウザ上のプログラミング学習環境  
JavaScript, C, DNCL (どんくり), Python に対応  
DNCL の仕様にはもっとも忠実 (?)

The screenshot shows the Bit Arrow web programming environment. At the top, there is a navigation bar with the following items: Bit Arrow, ファイル (File), 実行 (Execute), 保存 (Save), 設定 (Settings), and 使用方法 (Usage). Below the navigation bar, the file path is shown as 'factorial/sample1'. The main editor area contains a DNCL program with the following code:

```
DNCL sample1別ページで表示
1 n ← 1
2 a を 1 から 10 まで 1 ずつ増やしながら,
3   n ← a*n
4   n を表示する
5   を繰り返す
```

Below the code editor, there is a dialog box titled '実行画面ダイアログ' (Execution Screen Dialog). The dialog box contains the output of the program, which is the factorial of 10:

```
1
2
6
24
120
720
5040
40320
362880
3628800
```

① はじめに

② DNCL の実行環境

- PEN, PenFlowchart
- WaPEN
- どんくり

③ WaPEN への関数・手続きの実装

④ おわりに

# WaPEN における関数・手続きの実装

## 次期学習指導要領「情報Ⅰ」

…関数の定義・使用によりプログラムの構造を整理するとともに、性能を改善する工夫の必要性についても触れるようにする。

(「内容の取扱い」より)

## 関数の定義

関数 《関数名》 ( 《引数》 )

| ...

| 《値》 を返す

| ...

関数終了

## 手続きの定義

手続き 《手続き名》 ( 《引数》 )

| ...

| 手続きを抜ける

| ...

手続き終了

関数は値を返さなくてはならない

DNCL でいう「値を返さない関数」

いずれも呼び出すときは () を最後につける (引数がなくても)

## 関数の使用例

<pre> 1  整数 x, y 2 3  x を入力する 4  y を入力する 5  x を表示する 6  y を表示する 7  gcd(x, y) を表示する 8 9  関数 gcd(a, b) 10   整数 c 11   c ← a % b 12   もし c=0 ならば 13       b を返す 14       を実行し、そうでなければ 15       gcd(b, c) を返す 16       を実行する 17 関数終了 18 19 20 21 </pre>	<pre> 19661220  20190624 19661220 12 ---</pre>
--	--

## 手続きの使用例

```

1  整数 a[9],b[5]
2  a ← {59,34,24,99,38,10,29,89,81,6}
3  b ← {72,18,49,23,90,5}
4  summary(a)
5  summary(b)
6
7  手続き summary(x)
8  | 整数 s, i
9  | 実数 a
10 | s ← 0
11 | i を 0 から length(x)-1 まで 1 ずつ増やし;
12 | | s ← s+x[i]
13 | を繰り返す
14 | a ← (s + 0.0)/length(x)
15 | 「合計：」とs を表示する
16 | 「平均：」とa を表示する
17 手続き終了
18
19
20
21

```

```

合計：469
平均：46.9
合計：257
平均：42.833333333333336

```

```
----
```

## 細かい改良

- 「変数を確認する」を実装
- 配列が等しいかを比較できるようにした

## 「変数を確認する」の例

```

1  整数 a,b,c
2  a を入力する
3  b を入力する
4  繰り返し,
5  | c ← a%b
6  | a ← b
7  | b ← c
8  | 変数を確認する
9  | を, c=0 になるまで実行する
10 | a を表示する
11
12
13
14
15
16
17
18
19
20
21

```

49

```

*** 変数確認 ***
a:49
b:21
c:21
*** 変数確認 ***
a:21
b:7
c:7
*** 変数確認 ***
a:7
b:0
c:0
7
---

```

## 配列の比較の例

```

1  整数 a[5],b,c,n
2  n ← 0
3  繰り返し,
4  | a ← [0,0,0,0,0]
5  | n ← n + 1
6  | b を 1 から 60 まで 1 ずつ増やしながら,
7  | | c ← random(5)
8  | | a[c] ← a[c] + 1
9  | を繰り返す
10 | a と n を表示する
11 | を, a = [10,10,10,10,10] になるまで実行
12
13
14
15
16
17
18
19
20
21

```

```

[16,12,6,10,9,7]55324
[13,5,4,12,11,15]55325
[14,12,8,7,11,8]55326
[15,10,8,7,11,9]55327
[14,9,8,9,12,8]55328
[9,11,8,8,9,15]55329
[13,7,11,9,8,12]55330
[13,9,11,9,9,9]55331
[9,7,13,13,8,10]55332
[4,10,9,20,10,7]55333
[11,7,11,13,8,10]55334
[7,7,14,10,14,8]55335
[10,5,11,7,14,13]55336
[7,9,15,13,7,9]55337
[6,15,12,13,6,8]55338
[10,12,10,5,7,16]55339
[9,10,10,6,16,9]55340
[10,10,10,10,10,10]55341
---
```

## 自動採点

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  整数 y,m,d,j,k,h
2  y を入力する
3  m を入力する
4  d を入力する
5  j ← y/100
6  k ← y%100
7  もし m<3 ならば
8  | y ← y-1
9  | m ← m+12
10  を実行する
11  h ← d+(13*m+8)/5+k+k/4+j/4+5*j
12  h%7 を表示する
13
14
15
16
17
18
19
20
21

```

```

*** 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 ---

```

- ① はじめに
- ② DNCL の実行環境
  - PEN, PenFlowchart
  - WaPEN
  - どんくり
- ③ WaPEN への関数・手続きの実装
- ④ おわりに

# おわりに

WaPEN の配布先 (GitHub の方がおすすめ)

- 筆者のサイト <https://watayan.net/prog/>
- GitHub <https://github.com/watayan/WaPEN.git>

## インストール方法

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

## 更新方法

```
git pull
```

自分のサイトに置くのがおすすめ → レンタルサーバとか

- ファイルを置くだけで使える
- サンプルとか自由に変更られる
- 他にもいろいろ置きゃえばいい

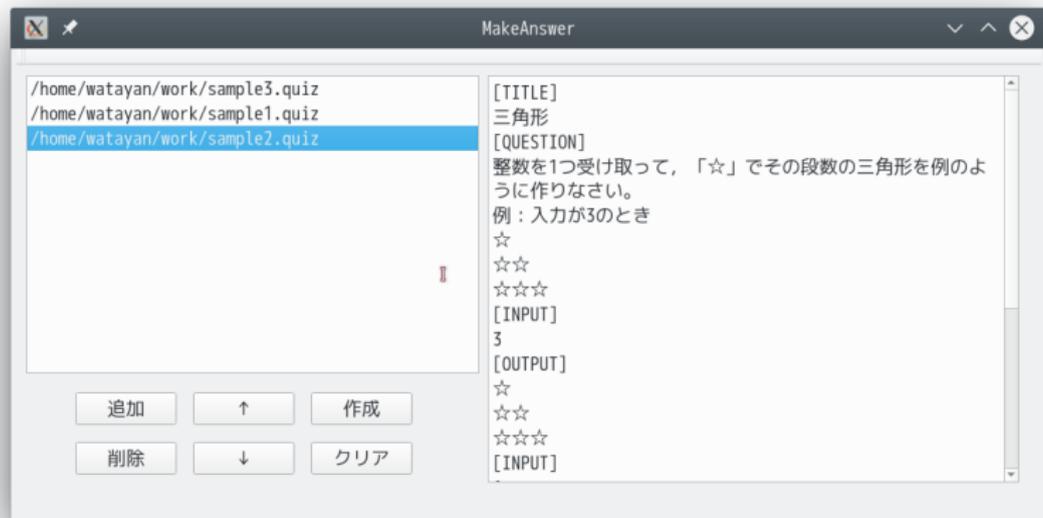
そのために開発

**MakeSample** サンプルプログラムの差し替え

**MakeAnswer** 自動採点問題・解答の差し替え

- Qt だからいろいろな OS に対応
- Windows 用インストーラは筆者のサイトに

## MakeAnswer, MakeSample の画面（ほぼ同じ）



# One more thing...

こういうの需要あるかなあ...

関数 prime(a) :

| 整数 n

| もし a=1 ならば :

| | false を返す

| n ← 2

| n<a の間繰り返す :

| | もし a%n=0 ならば :

| | | false を返す

| | n ← n+1

| true を返す

prime(20190811) を表示する