

Web 上のプログラミング学習環境の紹介

中西 渉

watayan@meigaku.ac.jp
名古屋高等学校

2017 年 11 月 21 日

目次

- ① はじめに
- ② 各プログラミング学習環境
 - テキストによるプログラミング
 - ブロックによるプログラミング
- ③ WaPEN について
 - PEN → PenFlowchart → WaPEN
 - 機能の違い
 - PEN の xDNCL との違い
 - 想定している使用環境
 - 使用実績
 - 余談：PEN とブロックプログラミング
- ④ おわりに

1 はじめに

2 各プログラミング学習環境

- テキストによるプログラミング
- ブロックによるプログラミング

3 WaPEN について

- PEN → PenFlowchart → WaPEN
- 機能の違い
- PEN の xDNCL との違い
- 想定している使用環境
- 使用実績
- 余談：PEN とブロックプログラミング

4 おわりに

教科書で使われるプログラミング言語

「情報の科学」で使われるプログラミング言語といえば...

- Excel の VBA
- JavaScript
- ドリトル

教科書で使われるプログラミング言語

「情報の科学」で使われるプログラミング言語といえば...

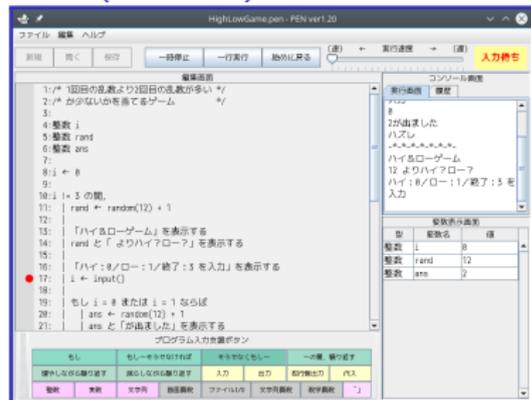
- Excel の VBA
- JavaScript
- ドリトル

理由は？→特別なソフトウェアが不要

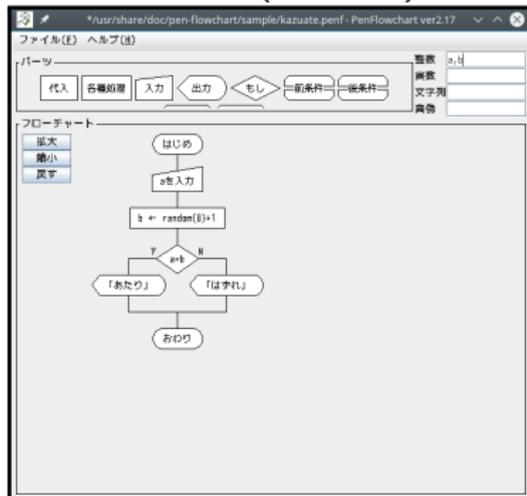
勤務校のプログラミング学習環境

OpenOffice.org の BASIC(2004～)

PEN(2006～)



PenFlowchart(2011～)



Web 上のプログラミング学習環境

PEN, PenFlowchart は Java アプリケーション

→インストール, JRE が必要

→使えない学校も

Web 上のプログラミング学習環境

PEN, PenFlowchart は Java アプリケーション

→インストール, JRE が必要

→使えない学校も

→ Web アプリケーションなら準備不要

① はじめに

② 各プログラミング学習環境

- テキストによるプログラミング
- ブロックによるプログラミング

③ WaPEN について

- PEN → PenFlowchart → WaPEN
- 機能の違い
- PEN の xDNCL との違い
- 想定している使用環境
- 使用実績
- 余談：PEN とブロックプログラミング

④ おわりに

Bit Arrow...の前にドリトル

ドリトル V2.38 (14 May 2015) [usr/share/doc/driftor/sample/ch3/strasure.dtl]

実行 編集

プログラム (3/11)

```

1 // 「宝物探しゲーム」 (「ドリトルで学ぶプログラミング」Activity4)
2
3 // タートルを操作する (ステップ1)
4 カメ太 = タートル! 作る。
5 左ボタン = ボタン! "左" "LEFT" 作る。
6 左ボタン = 動作 = 「カメ太! 30 左回り」。
7 右ボタン = ボタン! "右" "RIGHT" 作る。
8 右ボタン = 動作 = 「カメ太! 30 右回り」。
9
10 // タートルを前進させる (ステップ2)
11 時計 = タイマー! 作る 200 回数。
12 時計! 「カメ太! 16 歩く」実行。
13
14 // 宝物を画面に置く (ステップ3)
15 宝 = タートル! 作る "tombo.gif" 変身する ペンなし。
16 宝! (乱数 (600) - 300) (乱数 (300) - 150) 位置。
17 宝! 作る (乱数 (600) - 300) (乱数 (300) - 150) 位置。
18 宝! 作る (乱数 (600) - 300) (乱数 (300) - 150) 位置。
19
20 // 宝物を拾う (ステップ4)
21 カメ太 = 衝突 = 「| 相手 | 相手! 消える」。
  
```

実行! 中断 戻る 保存 印刷 終了 方眼紙

ドリトル V2.38 (14 May 2015) [usr/share/doc/driftor/sample/ch3/strasure.dtl]

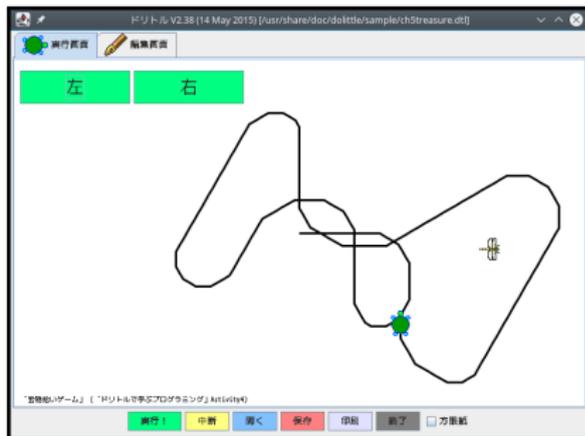
実行 編集

左 右

宝物探しゲーム (「ドリトルで学ぶプログラミング」Activity4)

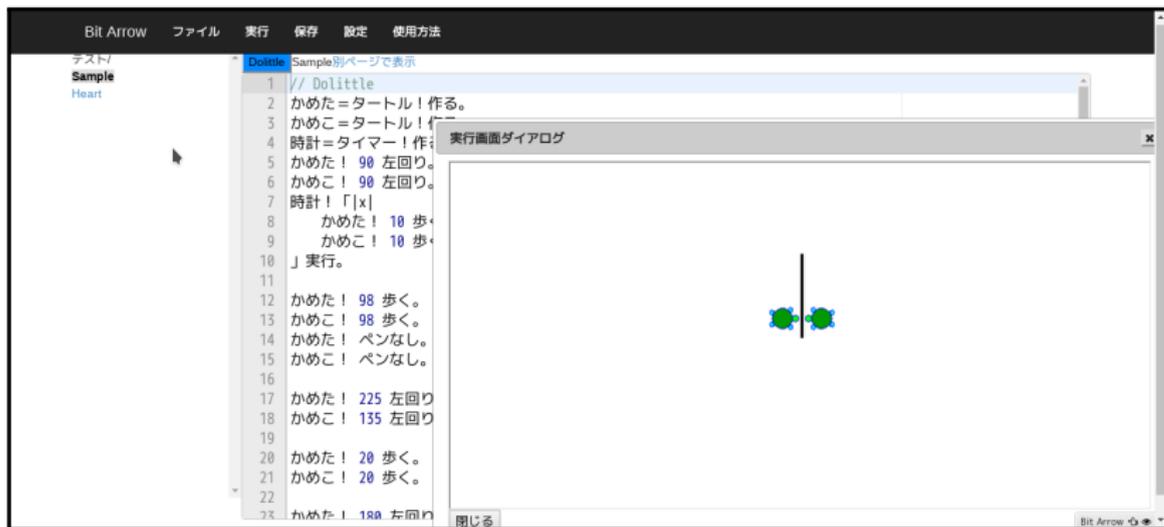
実行! 中断 戻る 保存 印刷 終了 方眼紙

Bit Arrow...の前にドリトル



Java アプリケーション
→ インストールが必要

Bit Arrow



Web ブラウザ上で動作

JavaScript, ドリトル, 簡易 C

JavaScript は便利な記述を追加

<http://bitarrow.eplang.jp/>

Bit Arrow



The screenshot shows the Bit Arrow IDE interface. The menu bar includes 'Bit Arrow', 'ファイル', '実行', '保存', '設定', and '使用方法'. On the left, there is a sidebar with 'テスト/Sample/Heart'. The main editor area shows a program with the following code:

```
1 // Dolittle
2 かめた=タートル!作る。
3 かめこ=タートル!作る "ayumiRed.png" 変身する。
4 時計=タイマー!作る 30 回数。
5 かめた! 45 左回り。
6 かめこ! 135 左回り。
7 時計! 「|x|
8   かめた! (12-(x)/4) 歩く (x-20) 左回り。
9   かめこ! (12-(x)/4) 歩く (x-20) 右回り。
10 「x==25」!なら「かめた!ペンなし。かめこ!ペンなし」実行。
11 」実行。
```

Below the code, a small window titled '別ページで開く' (Open in another page) is displayed, containing a QR code.

作ったプログラムをQRコード経由で実行することも可能

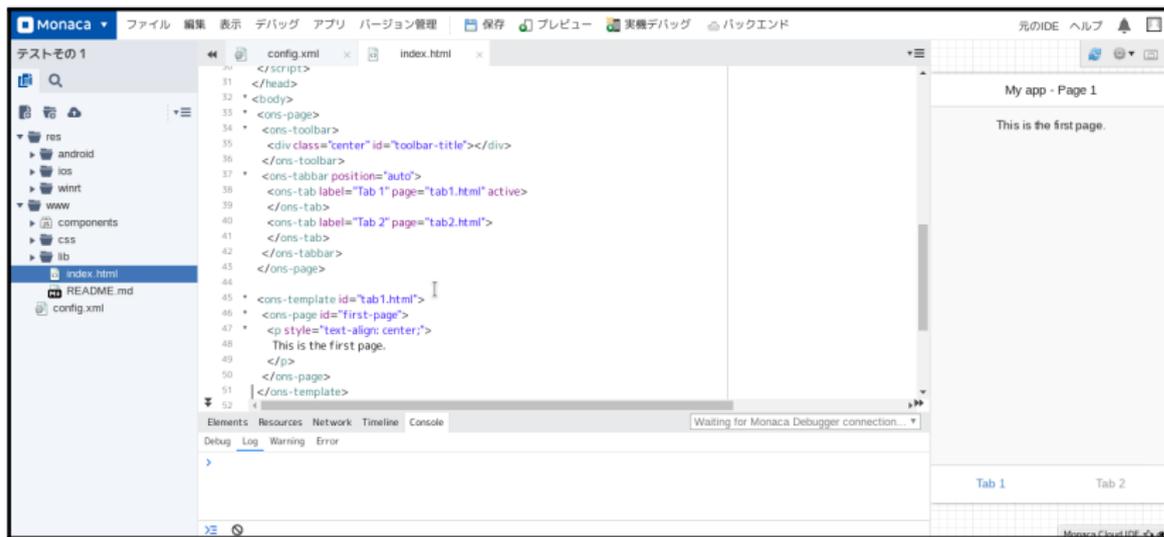
paiza.IO

```
2 process.stdin.setEncoding('utf8');
3 // Your code here!
4 var plus=function(a,b)
5 {
6   return a+b;
7 }
8
9 function reduce(a, f)
10 {
11   var s;
12   for(var i = 0; i < a.length; i++)
13   {
14     if(i == 0) s = a[i];
15     else s = f(s, a[i]);
16   }
17   return s;
18 }
19
20 var s = reduce([0,1,2,3,4,5,6,7,8,9],plus);
21 console.info(s);
```

ブラウザ上でプログラミング
→動作確認
多くの言語に対応
(C,VB,JavaScript,Java,...)

<https://paiza.io/ja>

Monaca



スマートフォンのアプリをブラウザ上で作成
自分のスマートフォンですぐに確認できる

<https://edu.monaca.io>

Scratch

The screenshot shows the Scratch web interface. The main stage displays a cat sprite (Sprite1) on a green hill with a red apple above it. The script area on the right contains the following code:

```
右向き矢印 キーが押されたとき
90° 度に向ける
10 歩動かす
次のコスチュームにする

左向き矢印 キーが押されたとき
-90° 度に向ける
10 歩動かす
次のコスチュームにする

がクリックされたとき
大きさを 100 % にする
ずっと
もし Apple に触れた なら
大きさを 50 % にする
```

<https://scratch.mit.edu>

Scratch

Scratch 2.0 は Flash アプリケーション
→ Flash は 2020 年以降使えない？

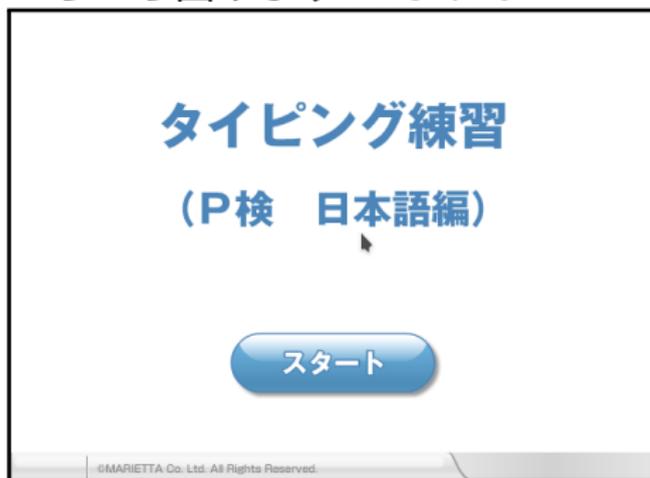
Scratch

Scratch 2.0 は Flash アプリケーション
→ Flash は 2020 年以降使えない？

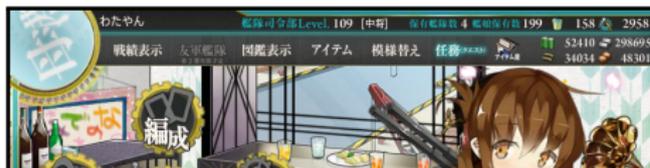
Scratch 3.0 は HTML5 で開発中

Flash がなくなると…

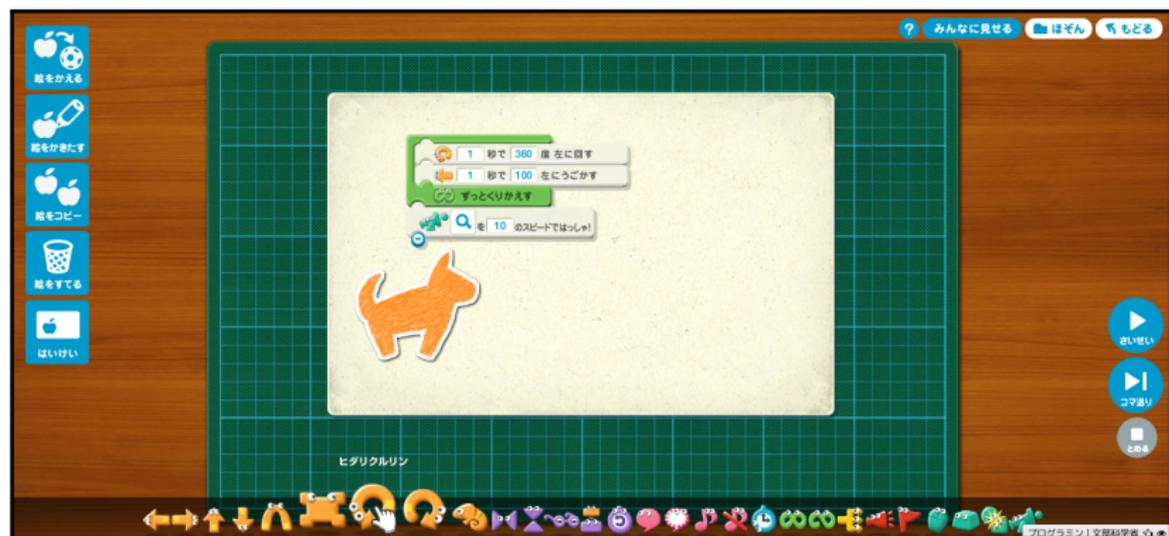
いろいろ困ります…よね？



↑ JavaScript 版は Firefox で不具合がある



プログラミング



文部科学省が運営
Flash アプリケーションなので...

<http://www.mext.go.jp/programin/>

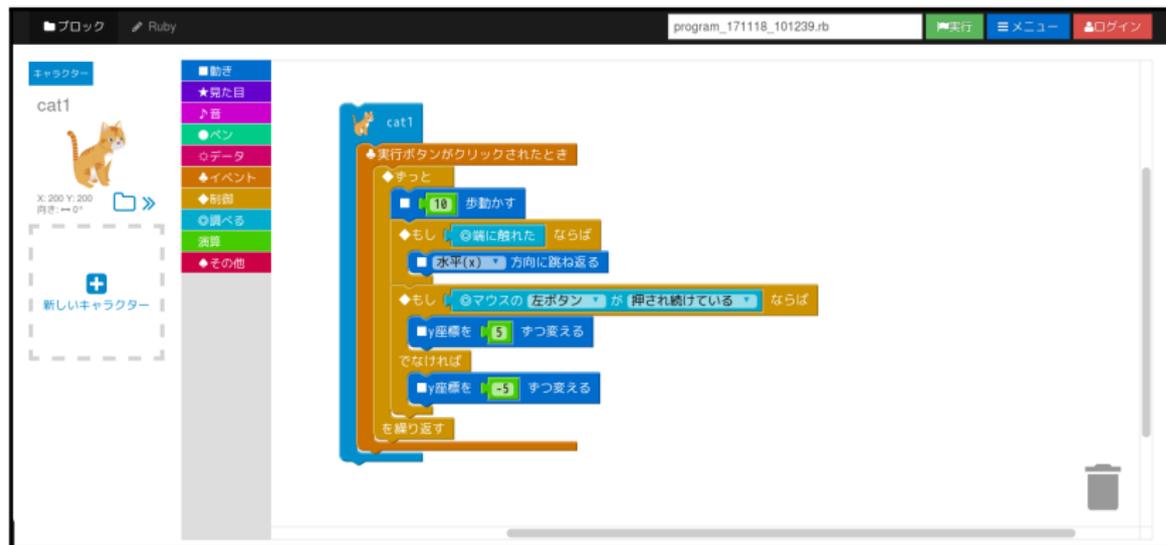
micro:bit の JavaScript ブロックエディター



小型ボード型コンピュータ micro:bit 用の開発環境
シミュレータで動作確認できる
ブロックと JavaScript の切り替えができる

<http://microbit.org/ja/>

スモウルビー



Ruby用の2Dゲーム開発ライブラリ Smalruby用の開発環境
ブロックとRubyの切り替えができる
実行環境はインストールしないとダメ？

<https://github.com/smalruby/>

1 はじめに

2 各プログラミング学習環境

- テキストによるプログラミング
- ブロックによるプログラミング

3 WaPEN について

- PEN → PenFlowchart → WaPEN
- 機能の違い
- PEN の xDNCL との違い
- 想定している使用環境
- 使用実績
- 余談：PEN とブロックプログラミング

4 おわりに

PEN → PenFlowchart → WaPEN

PEN

- Programming Environment for Novice
- DNCL を拡張した xDNCL

HighLowGame.pen - PEN ver1.20

ファイル 編集 ヘルプ

新規 開く 保存 一時停止 一行実行 始めに戻る (速) ← 実行速度 → (遅) 入力待ち

編集画面

```

1: /* 1回目の乱数より2回目の乱数が多い */
2: /* 少ないかを当てるゲーム */
3:
4: 整数 i
5: 整数 rand
6: 整数 ans
7:
8: i ← 0
9:
10: i = 3 の間,
11:   | rand ← random(12) + 1
12:   | 「ハイ&ローゲーム」を表示する
13:   | rand と 「よりハイ? ロー?」を表示する
14:   |
15:   | 「ハイ: 0/ロー: 1/終了: 3 を入力」を表示する
16:   | i ← input()
17:   |
18:   | もし i = 0 または i = 1 ならば
19:   |   | ans ← random(12) + 1
20:   |   | 「が出ました」を表示する
21:   |

```

コンソール画面

実行画面 設置

```

0
2が出ました
ハズレ
ハイ&ローゲーム
12 よりハイ? ロー?
ハイ: 0/ロー: 1/終了: 3 を
入力

```

変数表示画面

型	変数名	値
整数	i	0
整数	rand	12
整数	ans	2

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〜	〜の裏 繰り返す
繰り返しがら繰り返す	繰り返しがら繰り返す	入力	出力 実行無出力 代入
整数	実数	文字列	画面再読
ファイルI/O	文字列再読	数字再読	「」

PEN → PenFlowchart → WaPEN

PenFlowchart

- フローチャートを作ることによってプログラム生成

The screenshot displays the PenFlowchart application window. On the left, a flowchart is shown with the following steps:

- はじめ (Start)
- aを入力 (Input a)
- $a \leftarrow \text{random}(8)+1$ (Assign a random value to a)
- Decision: $a=b$
 - If Yes (Y): 「あたり」 (Hit)
 - If No (N): 「はずれ」 (Miss)
- おわり (End)

On the right, the generated code is displayed in the '編集画面' (Edit Screen):


```

    1: 整数 a,b
    2: a を入力する
    3: b ← random(8)+1
    4: もし a=b ならば
    5: | 「あたり」 を表示する
    6: を実行し、そうでなければ
    7: | 「はずれ」 を表示する
    8: を実行する
    9:
    
```

At the bottom, a 'プログラム入力変換ボタン' (Program Input Conversion Button) table is visible:

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
繰り返しのほの繰り返し	繰り返しのほの繰り返し	入力	出力
整数	実数	文字列	数値変数
整数	実数	文字列	数値変数

PEN → PenFlowchart → WaPEN

WaPEN

- PEN, PenFlowchart → HTML5 で移植
- Web-aided PEN

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

Upload ファイルを選択 選択されていません [Download](#) ファイル名:

1	整数 a,b	3!=6
2	a←1	4!=24
3	bを1から100まで1ずつ増やしながら,	5!=120
4	a←a*b	6!=720
5	bと「!」とaを表示する	7!=5040
6	を繰り返す	8!=40320
7		9!=362880
8		10!=3628800
9		11!=39916800
10		12!=479001600
11		13!=6227020800
12		14!=87178291200
13		15!=1307674368000
14		16!=20922789888000
15		17!=355687428096000
16		18!=6402373705728000
17		実行時エラーです
18		4行目:整数で表される範囲を越えま
19		した
20		
21		

整数 サイズ:

実数 サイズ:

文字列 サイズ:

真偽

整数 実数 文字列

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 増やしながら 減らしながら

入力 出力 改行無出力 代入

サンプル1
サンプル2
サンプル3
サンプル4
サンプル5
サンプル6

```

graph TD
    Start([はじめ]) --> A[a←1]
    A --> B[b:1→100  
1ずつ増]
    B --> C[a←a*b]
    C --> D{bと「!」とa}
    D --> End([おわり])
  
```

機能の違い

なるべく PEN, PenFlowchart を踏襲したが...

- 変数表示画面が未実装
- 入力支援ボタンと右クリックメニューの使い分け
- フローチャート画面の UI 変更
- (カット&ペーストは未実装)

PEN の xDNCL との違い

あまり大きい影響はないはず

- 複数の代入文を 1 行に書けない
- 関数や手続きが作れない
- File I/O がない
- 「そうでなくもし」の構文がない
- $a \leftarrow [1, 2, 3]$ のように、配列にまとめて代入できる

想定している使用環境

準備

- ① 配付ファイルを取得
 - `http://watayan.net/prog/#wapen` からダウンロード
 - `https://github.com/watayan/wapen.git` から `git clone`
- ② `sample.js-dist` を `sample.js` にコピー
- ③ `sample.js` のサンプルプログラムを書き換える
- ④ `index.html` にも手を加えたり...

あとは `index.html` を読み込むとか、
Web サーバに置くとか

適当な Web サーバがなければ...

レンタルサーバで十分
(月 500 円くらい?)

一つ持っておくと、何かと便利



watayan.net



aichi-edu.info

使用実績

2017 年度 情報処理学会の教員免許更新講習
センター試験「情報関係基礎」問 3 のプログラムを実行する演習で
使用

- 会場の Mac に PEN, PenFlowchart をインストールするのが面倒
- 変数の宣言なしモードで実行したかった
- 試験問題にない部分のプログラムを配付したかった

授業では未使用（勤務校では 3 学期）

余談：PEN とブロックプログラミング

PEN のブロックプログラミングによる実装 oPEN

The screenshot shows a Blockly-based programming environment. The script consists of the following blocks:

- プログラムを起動 (Start program)
- 変数 x を初期化 (Initialize variable x)
- x に値を入力 (Input value for x)
- 条件分岐 (Conditional branch) with condition $x \% 2 = 0$
 - 真 (True) branch: 文字列 偶数 (String: Even)
 - 偽 (False) branch: 文字列 奇数 (String: Odd)

The output window displays the following text:

```
入力 12
実行結果
偶数
====プログラム終了====
```

At the bottom, a table shows the variable `x` with the value `12`.

Blockly での実装の構想もあるらしい...

① はじめに

② 各プログラミング学習環境

- テキストによるプログラミング
- ブロックによるプログラミング

③ WaPEN について

- PEN → PenFlowchart → WaPEN
- 機能の違い
- PEN の xDNCL との違い
- 想定している使用環境
- 使用実績
- 余談：PEN とブロックプログラミング

④ おわりに

おわりに

Web 上でプログラミング学習が可能だが...

→インストール版も有用

- ハードウェア，ローカルマシンの資源にアクセス
- ネットワークなしで使える

WaPEN は「素材」

...改造したり，何かに組み入れたり

...どんなライセンスがいいんだろう...