

PyPEN の進化

中西渉

watayan@meigaku.ac.jp
名古屋高等学校

2025年12月26日
令和7年度実践事例報告会

1 はじめに

2 PyPEN の概要

3 PyPEN の進化

4 おわりに

1 はじめに

- 概要
- 発表者紹介

2 PyPEN の概要

3 PyPEN の進化

4 おわりに

概要

PyPEN : プログラミング学習環境 (Web ブラウザで稼働)

筆者の一人開発

共通テスト「情報」プログラミング問題の言語に近い言語

最近の大きい変更点についてまとめてみる

「改良」と思っているが…

発表者紹介

氏名 中西渉（なかにし わたる）
所属 名古屋高等学校（1989-）
俗称 わたやん



Web <https://watayan.net/>
X @watayan

Facebook wataru.nakanishi

GitHub watayan

注力分野 プログラミング, データ分析

同人誌：技術書典, 技書博で頒布

● PyPEN 関係



● Vivliostyle 関係



1 はじめに

2 PyPEN の概要

- PyPEN とは
- PyPEN の特徴
- 自分の環境に設置
- カスタマイズ

3 PyPEN の進化

4 おわりに

PyPEN とは

Web ブラウザ上で動作するプログラミング学習環境

新規 実行 ステップ実行 変数確認 ステップ+変数確認 リセット *

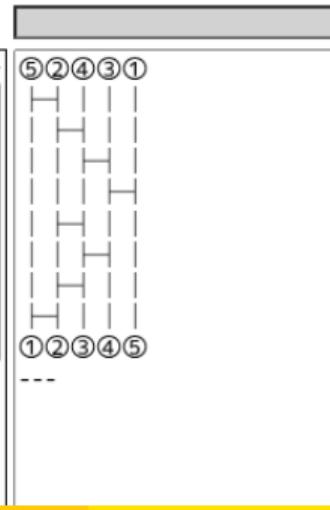
フローチャート コード→Python URL生成

Load Save ファイル名:

問題選択 探点

+ 0 -

```
1 手続き amida(tate,Yokosen,yoko):
2   y を 1 から yokosu まで 1 ずつ増やしながら:
3     x = 1
4     x <= tate の間:
5      もし Yokosen[y] == x ならば:
6         改行なしで表示する("ト")
7         改行なしで表示する("一")
8         x = x + 2
9       そうでなければ:
10      改行なしで表示する("|")
11      x = x + 1
12      改行する
```



PyPEN の特徴

- 共通テスト「情報」プログラミング問題の言語に近い言語
- インストール不要、Web ブラウザで動作
- 自分の環境に設置可能
- 改造は自由（MIT ライセンス）
- 入力支援ボタンなど

自分の環境に設置

- ① GitHub から

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

- ② 筆者のサイトからダウンロード

<https://watayan.net/prog/pypen.html>

おすすめは GitHub

カスタマイズ

自分の環境に置くことのメリット：カスタマイズ可能

- 各機能部分の表示 ON/OFF
- 自動採点問題・サンプルプログラムの差し替え
<https://watayan.net/prog/wapentools.html>
- デザインの変更

1 はじめに

2 PyPEN の概要

3 PyPEN の進化

- 参照の値渡し
- BigInt 対応
- 関数・手続きの一本化
- 関数の充実
- range っぽいやつ
- Python のキーワードへの対応

4 おわりに

参照の値渡し

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: []

[+] [-]

```
1 a = [1, 2, 3]
2 b = a
3 c = copy(a)
4 a[0] = 100
5 b[1] = 200
6 変数を確認する
```

*** 変数確認 ***
a:[100,200,3]
b:[100,200,3]
c:[1,2,3]

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: []

[+] [-]

```
1 a = [1, 2, 3]
2 b = a
3 c = a
4 a[0] = 100
5 b[1] = 200
6 変数を確認する
```

*** 変数確認 ***
a:[100,2,3]
b:[1,200,3]
c:[1,2,3]

[マニュアル](#)

BigInt 対応

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット]

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: _____

```
1 p := 1
2 n を 1 から 100 まで 1 ずつ増やしながら :
3 ... p *= n
4 ... 表示する(n, "!=" , p)
```

```
00000000000000000000000000000000
99 !=
933262154439441526816992388562667
004907159682643816214685929638952
175999932299156089414639761565182
862536979208272237582511852109168
6400000000000000000000000000000000
100 !=
933262154439441526816992388562667
004907159682643816214685929638952
175999932299156089414639761565182
862536979208272237582511852109168
6400000000000000000000000000000000
---
```

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット]

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: _____

```
1 p := 1
2 n を 1 から 100 まで 1 ずつ増やしながら :
3 ... p *= n
4 ... 表示する(n, "!=" , p)
```

```
6 != 720
7 != 5040
8 != 40320
9 != 362880
10 != 3628800
11 != 39916800
12 != 479001600
13 != 6227020800
14 != 87178291200
15 != 1307674368000
16 != 20922789888000
17 != 355687428096000
18 != 6402373705728000
実行時エラーです
3行目:整数で表せない値です
```

[マニュアル](#)

関数・手続きの一本化

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

□ フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: []

[+] [-]

```
1 関数 f(x):
2     ... x ** 2 + 1 を返す
3
4 関数 g(x):
5     ... 表示する(x)
6
7 x = 3
8 g(f(x))
```

10

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

□ フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: []

[+] [-]

```
1 関数 f(x):
2     ... x ** 2 + 1 を返す
3
4 手続き g(x):
5     ... 表示する(x)
6
7 x = 3
8 g(f(x))
```

10

[マニュアル](#)

関数の充実

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット]

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名:

[+] [-]

```
1 a = [32, -13, -45, -72, -12, -4, -96, -35]
2 表示する(sum(a))
3 表示する(mean(a))
4 .
```

```
309
38.625
---
```

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名:

[+] [-]

```
1 関数 sum(a):
2     l = length(a)
3     s = 0
4     nを0からl-1まで1ずつ増やしながら:
5         s += a[n]
6     sを返す
7 関数 mean(a):
8     l = length(a)
9     sum(a) / l を返す
10 a = [32, -13, -45, -72, -12, -4, -96, -35]
11 表示する(sum(a))
12 表示する(mean(a))
```

```
309
38.625
---
```

[マニュアル](#)

range っぽいやつ

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット]

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: range

[+] [-]

1 range(10)の要素nについて:

2 ... 表示する(n)

0
1
2
3
4
5
6
7
8
9

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット]

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: range2

[+] [-]

1 n を 0 から 9 まで 1 ずつ増やしながら:

2 ... 表示する(n)

3 ...

0
1
2
3
4
5
6
7
8
9

[マニュアル](#)

Python のキーワードへの対応

新バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: []

```
1 def f(n):
2     if n < 2:
3         return 1
4     g = f(n-1)
5     return n * g
6
7 for i in range(1, 11):
8     print(f(i))
```

```
1
2
3
4
5
6
7
8
```

[マニュアル](#)

旧バージョン

[新規] [実行] [ステップ実行] [変数確認] [ステップ+変数確認] [リセット] *

フローチャート [コード→Python] [URL生成]

[Load] [Save] ファイル名: range2

```
1 関数 f(n):
2     もし n < 2 ならば:
3         1 を返す
4     g = f(n-1)
5     n * g を返す
6
7 iを1から10まで1ずつ増やしながら:
8     表示する(f(i))
```

```
1
2
3
4
5
6
7
8
```

[マニュアル](#)

1 はじめに

2 PyPEN の概要

3 PyPEN の進化

4 おわりに

おわりに

何のため？

- プログラミングの制約を減らしたい
- 共通テスト向け演習と Python の橋渡し

要望等あればご連絡ください

Facebook に PyPEN グループを作りました

緩募

PyPENに関する実践報告・論文