

po-5. リスト, 辞書

トピックス: リスト, 辞書 (Python Tutor による
演習)

URL: <https://www.kkaneko.jp/pro/po/index.html>

(Python プログラミングの基本)

金子邦彦



リスト

4 を末尾に挿入

list

0	1	2	3	4
15	8	6	32	23



list

0	1	2	3	4	5
15	8	6	32	23	4

8 の削除



list

0	1	2	3	4
15	6	32	23	4

辞書

4, "Orange" の削除

3 を "Black" に置き換え

1	"Red"
2	"Yellow"
3	"Blue"



1	"Red"
2	"Yellow"
3	"Blue"
4	"Orange"



1	"Red"
2	"Yellow"
3	"Black"
4	"Orange"

全体まとめ①



- リストは、同じ型の要素の並び
- リストの要素には順序がある。0から始まる番号（添字）が付いている
- リストは、要素の挿入，削除により，サイズが増減する

4 を末尾に挿入

list

0	1	2	3	4
15	8	6	32	23



list

0	1	2	3	4	5
15	8	6	32	23	4

8 の削除

list

0	1	2	3	4
15	6	32	23	4

全体まとめ②



- 辞書は、**キーと値（バリュー）のペアの集まり**
- 辞書に、**同じ値のキーは2回以上登場しない**

4, "Orange" の削除

3 を "Black" に置き換え

1	"Red"
2	"Yellow"
3	"Blue"



1	"Red"
2	"Yellow"
3	"Blue"
4	"Orange"



1	"Red"
2	"Yellow"
3	"Black"
4	"Orange"

	項目
	復習
5-1	リスト
5-2	「リスト」を演習できるオンラインサイトの紹介
5-3	辞書

メソッドアクセス, 代入



Python プログラムの例

```
x = 100
a = x + 200
enemy1 = hero.findNearestEnemy()
hero.attack(enemy1)
```

- **代入** : **オブジェクト名** + 「**=**」
+ 式または値またはメソッド呼び出し
- **メソッドアクセス** : **オブジェクト名** + 「**.**」
+ **メソッド名** + 「**()**」 (引数を付けることも)

Python プログラムでは, その他にも, 属性アクセス, 関数呼び出し, 制御, 「*」, 「+」などの演算子, コマンド, 定義など

Python Tutor の起動



① ウェブブラウザを起動する

② Python Tutor を使いたいのので, 次の URL を開く
<http://www.pythontutor.com/>

③ 「Python」 をクリック ⇒ 編集画面が開く

Learn Python, JavaScript, C, C++, and Java

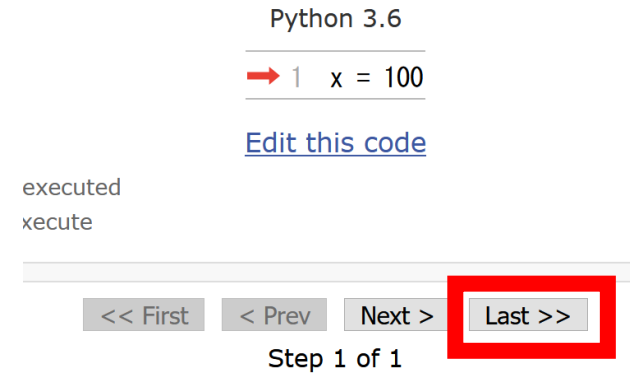
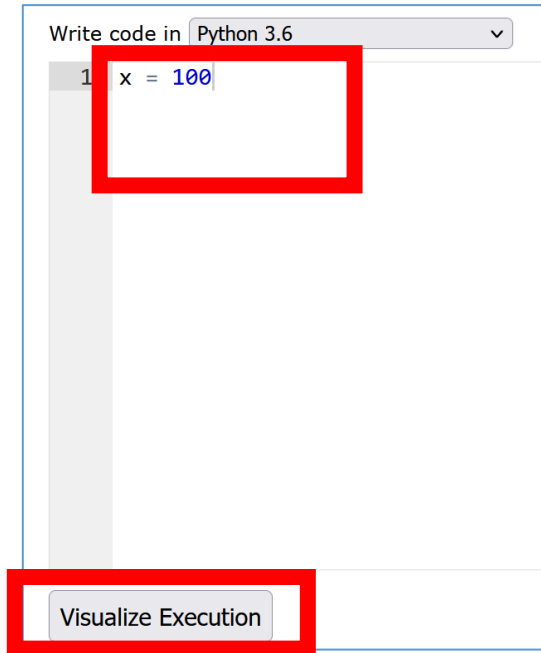
This tool helps you learn Python, JavaScript, C, C++, and Java programming by [visualizing code execution](#). You can use it to debug your homework assignments and as a supplement to online coding tutorials.

Start coding now in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Over 15 million people in more than 180 countries have used Python Tutor to visualize over 200 million pieces of code. It is the most widely-used program visualization tool for computing education.

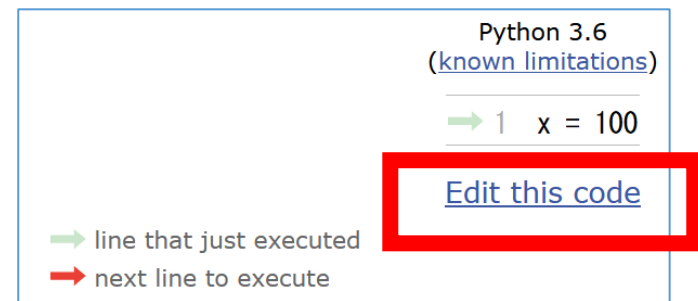
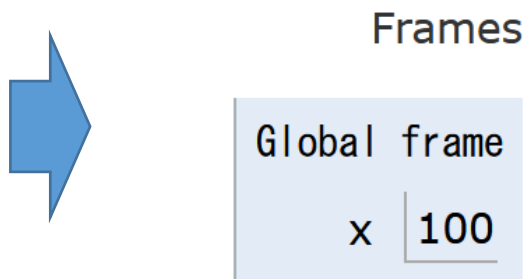
You can also embed these visualizations into any webpage. Here's an example showing recursion in Python:

Python Tutor でのプログラム実行手順



(1) 「**Visualize Execution**」をクリックして**実行画面**に切り替える

(2) 「**Last**」をクリック。



(3) 実行結果を確認する。

(4) 「**Edit this code**」をクリックして**編集画面**に戻る

Python Tutor 使用上の注意点①



- 実行画面で、次のような赤の表示が出ることもある →
無視してよい

過去の文法ミスに関する確認表示
邪魔なときは「Close」

Python Tutor: Visualize code in [Python](#), [JavaScript](#), [C](#), [C++](#), and [Java](#)

Python 3.6
([known limitations](#))

```
→ 1 x = 100
```

[Edit this code](#)

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

Step 1 of 1

[Customize visualization](#)

Frames Objects

You just fixed the following error:

```
1 x = 100!
```

SyntaxError: invalid syntax (<string>, line 1)

Please help us improve this tool with your feedback.
What misunderstanding do you think caused this error?

Submit Close [Hide all of these pop-ups](#)

Python Tutor 使用上の注意点②



「please wait ... executing」のとき，10秒ほど待つ。



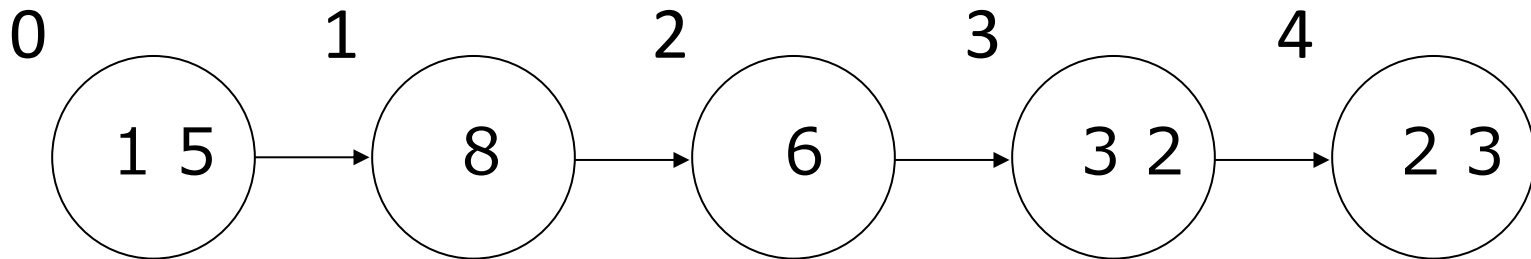
→ 混雑しているときは，「Server Busy・・・」
というメッセージが出ることがある。

混雑している。少し（数秒から数十秒）待つと自動で表示が変わる（変わらない場合には，操作をもう一度行ってみる）

5-1. リスト

リスト

- リストは、同じ型の要素の並び
- リストの要素には順序がある。0から始まる番号（添字）が付いている
- 要素の削除，挿入によりサイズが増減する



演習

資料 : 14 ~ 15

【トピックス】
・リスト

① Python Tutor のエディタで次のプログラムを入れ、実行し、結果を確認する

8, 6, 4, 2, 3 のリストを作る。
その後、合計を求める。

```
x = [8, 6, 4, 2, 3]
print(sum(x))
```



The screenshot shows the Python Tutor interface. On the left, the code editor displays the following code:

```
Python 3.6
1 x = [8, 6, 4, 2, 3]
2 print(sum(x))
```

Below the code is a button labeled "Edit this code". At the bottom of the editor, there are navigation buttons: "<< First", "< Prev", "Next >", and "Last >>". The status bar indicates "Done running (2 steps)".

On the right, the "Print output" window shows the result "23". Below it, the "Frames" and "Objects" panels are visible. The "Frames" panel shows a "Global frame" with a variable "x". The "Objects" panel shows a "list" object with the following values:

0	1	2	3	4
8	6	4	2	3

結果の
「23」を確認

「Visual Execution」をクリック。そして「Last」をクリック。結果を確認。
「Edit this code」をクリックすると、エディタの画面に戻る

② Python Tutor のエディタで次のプログラムを入れ、実行し、結果を確認する



6 月は 30 日までである。7 月は 31 日までである。

月の日数についてのリストを作る

※ うるう年のことは考えないことにする

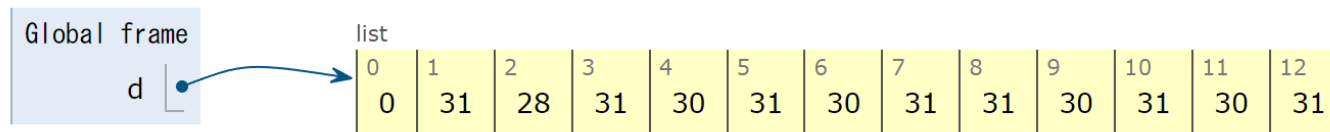
```
d = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
print(d[6])
print(d[7])
```

Print output (drag lower right corner to resize)

```
30
31
```

Frames

Objects



結果を確認

「**Visual Execution**」をクリック、そして「**Last**」をクリック、結果を確認。
「**Edit this code**」をクリックすると、エディタの画面に戻る

リストの性質

- リストは、要素の**削除**、**挿入**により**サイズが増減**する

list

0	1	2	3	4
15	8	6	32	23



4 を末尾に挿入

list

0	1	2	3	4	5
15	8	6	32	23	4



8 の削除

list

0	1	2	3	4
15	6	32	23	4

Python でのリストの組み立て, 要素の挿入, 要素の削除



list

0	1	2	3	4
15	8	6	32	23

リストの組み立て

```
x = [15, 8, 6, 32, 23];
```



要素の挿入

```
x.append(4)
```

0	1	2	3	4	5
15	8	6	32	23	4



要素の削除

```
x.remove(8)
```

0	1	2	3	4
15	6	32	23	4

演習

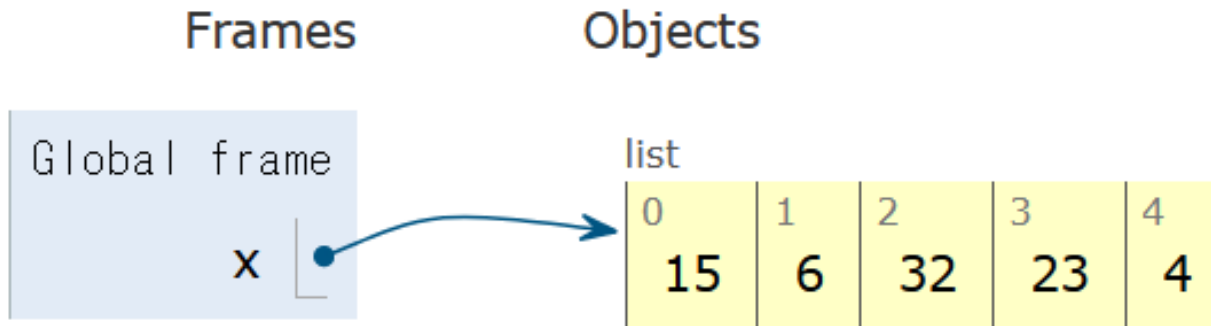
資料 : 19

【トピックス】

- ・ リストでの要素の挿入, 削除

① Python Tutor のエディタで次のプログラムを入れ, 実行し, 結果を確認する

```
x = [15, 8, 6, 32, 23]
x.append(4)
x.remove(8)
```

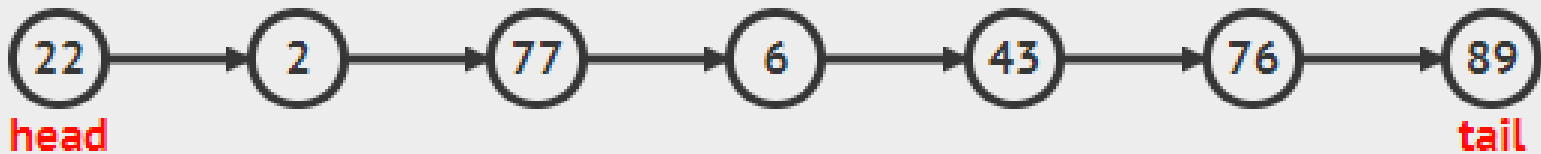


表示を確認

「**Visual Execution**」をクリック. そして「**Last**」をクリック. 結果を確認.
「**Edit this code**」をクリックすると, エディタの画面に戻る

5-2. 「リスト」を演習できる オンラインサイトの紹介

「リスト」を演習できる オンラインサイトの紹介



リストは、同じ型の要素の並び

① ウェブブラウザを起動する

② 次の URL を開く

<https://visualgo.net/ja>

③ 「**Linked List** (連結リスト)」をクリック



The screenshot shows the Visualgo website homepage. The main header includes the logo "VISUALGO .NET/EN" and the tagline "visualising data structures and algorithms through animation". A search bar is present, along with a "LOGIN" button. Below the header, there are promotional banners for "Competitive Programming 4" and "Lulu Buy Now!". The "Do You Know?" section contains a message about new features. At the bottom, there are three category tiles: "Sorting", "Bitmask", and "Linked List". The "Linked List" tile is highlighted with a red border and shows a diagram of a linked list with nodes containing 'H' and 'T'.

VISUALGO .NET/EN
visualising data structures and algorithms through animation

Search...

NUS Computing
National University of Singapore
Featured story: [Visualizing Algorithms with a Click](#)

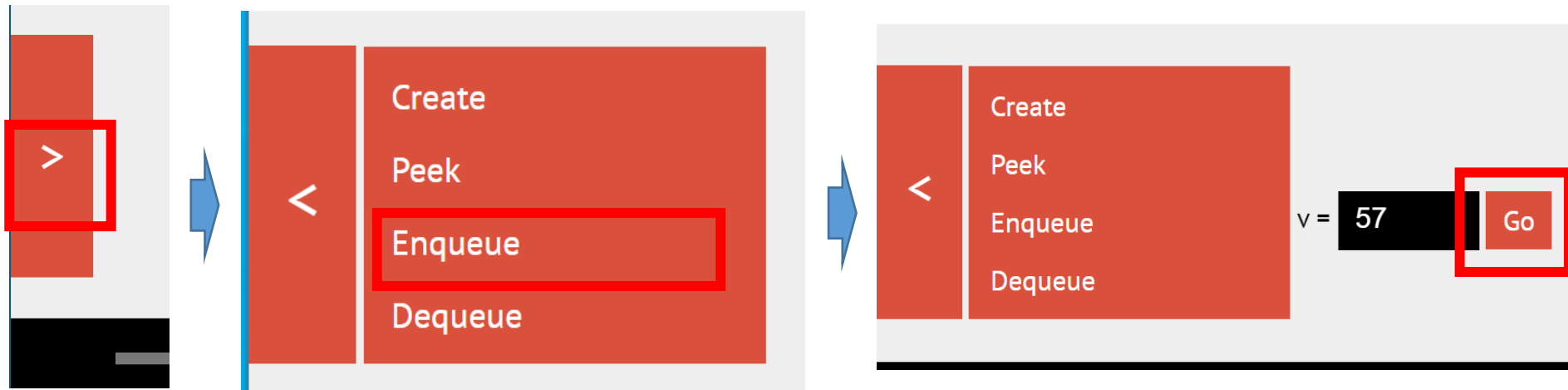
Do You Know? Next Random Tip
Please check the newest features of VisuAlgo: 1. User accounts system for NUS students and verified CS lecturers worldwide (and also read the latest Privacy Policy popup at the bottom right corner), 2. More mobile-friendly setup, 3. More polished e-Lecture notes to reach 'NUS standard', and 4. Trilingual capability (/en, /zh, or /id).

Sorting Training
array algorithm bubble select

Bitmask Training
bit manipulation set cs3233

Linked List Training
stack queue doubly deque

- ④ 説明が出る場合がある. **ESC** キーを押して, 説明を消す
- ⑤ 左下のメニューで「**Enqueue** (入れる)」をクリックし, 「**Go**」をクリック



⑥ 末尾に挿入されるので、確認する



5-3. 辞書

辞書



- **辞書**は, **キーと値 (バリュー)** の**ペア**の集まり
- **辞書**に, 同じ値の**キー**は**2回以上登場しない**

キー	値 (バリュー)
1	Red
2	Yellow
3	Blue

繰り返しのプログラム例

辞書の 組み立て



```
1 d = {1: "Red", 2: "Yellow", 3: "Blue"}
2 print(d[1])
3 print(d[2])
4 print(d[3])
```

辞書のキー 1, 2, 3 のそれぞれについて、値（バリュー）を表示

実行結果

```
Red
Yellow
Blue
```

演習

資料 : 29 ~ 31

【トピックス】

・ 辞書

① Python Tutor のエディタで次のプログラムを入れ、実行し、結果を確認する

```
d = {1: "Red", 2: "Yellow", 3: "Blue"}  
print(d[1])  
print(d[2])  
print(d[3])
```

Print output (drag lower right corner to resize)

```
Red  
Yellow  
Blue
```

Frames

Objects

Global frame

d

dict

1	"Red"
2	"Yellow"
3	"Blue"

結果を確認

「Visual Execution」をクリック、そして「Last」をクリック、結果を確認。
「Edit this code」をクリックすると、エディタの画面に戻る

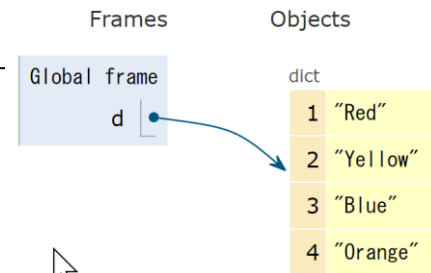
② Python Tutor のエディタで次のプログラムを入れ、実行し、結果を確認する

`d[4] = "Orange"` は、新しいキー、値（バリュー）の追加

```
d = {1: "Red", 2: "Yellow", 3: "Blue"}  
print(d[1])  
print(d[2])  
print(d[3])  
d[4] = "Orange"  
print(d[4])
```

Print output (drag lower right corner to resize)

```
Red  
Yellow  
Blue  
Orange
```



結果を確認

「**Visual Execution**」をクリック、そして「**Last**」をクリック、結果を確認。
「**Edit this code**」をクリックすると、エディタの画面に戻る

③ Python Tutor のエディタで次のプログラムを入れ, 実行し, 結果を確認する

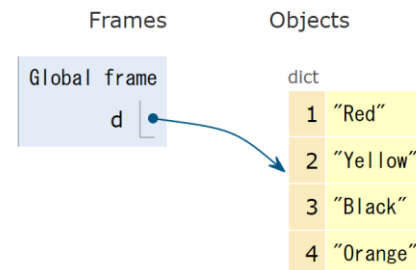


d[3] = "Black" は, 値 (バリュー) の置き換え

```
d = {1: "Red", 2: "Yellow", 3: "Blue"}
print(d[1])
print(d[2])
print(d[3])
d[4] = "Orange"
print(d[4])
d[3] = "Black"
print(d[3])
```

Print output (drag lower right corner to resize)

```
Red
Yellow
Blue
Orange
Black
```



結果を確認

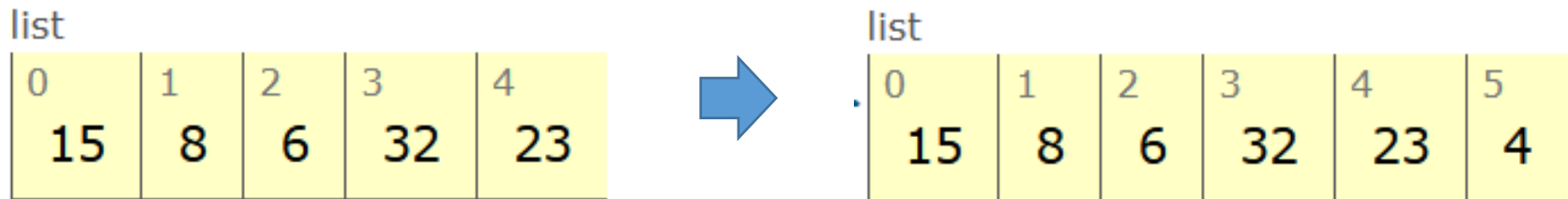
「Visual Execution」をクリック, そして「Last」をクリック, 結果を確認.
「Edit this code」をクリックすると, エディタの画面に戻る

全体まとめ①

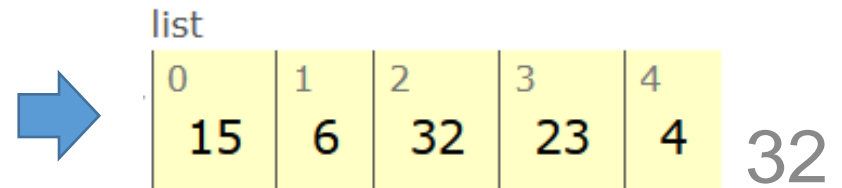


- リストは、同じ型の要素の並び
- リストの要素には順序がある。0から始まる番号（添字）が付いている
- リストは、要素の挿入，削除により，サイズが増減する

4 を末尾に挿入



8 の削除



全体まとめ②



- **辞書**は、**キーと値（バリュー）**の**ペア**の集まり
- **辞書**に、**同じ値のキー**は2回以上登場しない

4, "Orange" の削除

3 を "Black" に置き換え

1	"Red"
2	"Yellow"
3	"Blue"



1	"Red"
2	"Yellow"
3	"Blue"
4	"Orange"



1	"Red"
2	"Yellow"
3	"Black"
4	"Orange"

Python 関連ページ



- Python まとめページ

<https://www.kkaneko.jp/tools/man/python.html>

- Python 入門（スライド資料とプログラム例）

<https://www.kkaneko.jp/pro/pf/index.html>

- Python プログラミングの基本（スライド資料とプログラム例）

<https://www.kkaneko.jp/pro/po/index.html>

- Python プログラム例

<https://www.kkaneko.jp/pro/python/index.html>

- 人工知能の実行（Google Colaboratory を使用）

<https://www.kkaneko.jp/ai/ni/index.html>

- 人工知能の実行（Python を使用）（Windows 上）

<https://www.kkaneko.jp/ai/deepim/index.html>