

mi-3. 状態空間表現での総当たり 人工知能を演習で学ぶシリーズ (3)

<https://www.kkaneko.jp/dblab/intro/m/index.html>

金子邦彦



3-1 Python の変数と条件分岐

3-2 Python の**配列**と繰り返し

3-3 「2つの水差し」のシミュレーション

3-4 「2つの水差し」を**総当たり**で解く

アクティブラーニングの効果

- 面白い
 - 集中できる
 - 居眠りしなくなる 自分で工夫する.
 - 質問しやすい
 - 仲間に教えることが楽しい 仲間と教えあう.
 - **仲間に教えているともっと分かる**
 - **友達が増える**
 - **クラスの雰囲気がよくなる**
- など

3-1 Python の変数と条件分岐 (paiza.IOを使用)

実習タイム

資料

<https://www.kkaneko.jp/a/m.html>

ケイ かねこ ドット ジエイ ピー

スラッシュ

エイ

スラッシュ

エム ドット エイチ ティー エム エル

第2回授業

説明資料（短縮版）：[第2回授業資料 \[PDF版\]](#), [\[パワーポイント版\]](#),

- 2-2. コードコンバットで Python の練習を行ってみる

その資料：[コードコンバットで Python の練習を行ってみる](#)

コードコンバットのWebページ <https://codecombat.com/> で、「**Sign up as an individuals**」で、各自サインアップ

※ コードコンバットの使用で、**IE が動かないときは、Google Chrome Webブラウザを用いてください**

パスワードは大事。自分で考えて設定すること。（大学のパスワードなどを使いまわさないこと）。設定したら覚えておくこと。

コードコンバットを使っている途中で、
Internet Explorer がうまく動かなくなった場合
には、
Google Chrome を使ってください

実習タイム

第3回授業

- 3-1 配列と繰り返し

paiza.IO のウェブページ: <https://paiza.io/>

ここで「Python 3」を選ぶ。表示が日本語になっている場合には、英語に変えることができます

paiza.IO の参考資料: 「[paiza.IO を使ってみる](#)」

- 実習プログラム

```
import sys
x, y = 0, 0
answer = [1, 2, 3, 4, 5, 6, 7, 8]
for i in answer:
    if i == 1 and x < 4:
        x, y = 4, y
    elif i == 2 and y < 3:
        x, y = x, 3
    elif i == 3 and x > 0:
```

paiza.IO

- プログラミングを行える Web サービス

<https://paiza.io>

- Web ブラウザを使う

- たくさんの言語を扱うことができる

Java, Ruby, **Python**, PHP, Perl など

Python

- **プログラミング言語**の1つ
- 無料で使うことができる言語である
- Python には, バージョン 2, バージョン 3 などがある.
 - ※ この授業では, **バージョン 3** を使用

実習準備

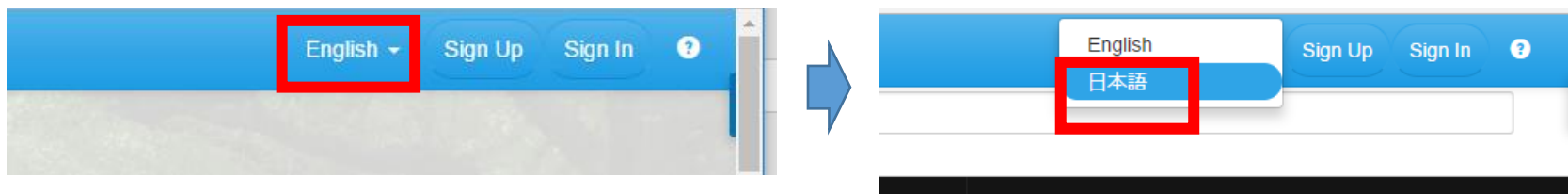
① ウェブブラウザを起動する

② 次の URL を開く

<https://paiza.io/>



③ もし、表示が英語になっていたら、**日本語**に切り替える



④ 「コード作成を試してみる」をクリック



⑤ 「Python 3」を選ぶ

(左上のボタンをクリックするとメニューが出る)



paiza.io

Python3

Enter a title here

Main.py

```
1 # coding: utf-8
2 # Your code here!
3
4
```

実行 (Ctrl-Enter)

プログラムの
編集画面

プログラムを
書き換えること
ができる

実行ボタン

実習 1. 計算

① 次のように入れて

print(100*200)

すべて 半角文字

「*」は掛け算の記号

```
1 print(100*200)
2
```

```
1 # coding: utf-8
2 # Your code here!
3
```

元からあるこの2行は消しても良い

② 実行ボタンをクリック

結果を確認

```
1 print(100*200)
2
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

20000

実習 2. 変数

変数は、値を覚えておくためのメモリ

① 次の 2 行を入れる

※ 前のものは消す

x = 100

print(x)

② 実行ボタンをクリック

x の値が表示される

```
1 x = 100
2 print(x)
3
```



実習 3. 変数と計算

次のように入れて、実行ボタンをクリック。

結果を確認

a = 10

b = 20

print(a + b)

print(a * b)

print((a + 10) * b)

★ 全員、この結果が
得られるのを待ちま
す。

仲間同士で助け
合ってください

```
1 a = 10
2 b = 20
3 print(a + b)
4 print(a * b)
5 print((a + 10) * b)
6
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

30
200
400

実習 4. 三角形の面積

底辺が **2.5** で, 高さが **5** のとき,

三角形の面積は, 面積 : **6.25**

次のように入れて, 実行ボタンをクリック. 結果を確認

```
teihen = 2.5
```

```
takasa = 5
```

```
print(teihen * takasa / 2)
```

```
1 teihen = 2.5
2 takasa = 5
3 print(teihen * takasa / 2)
4
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

6.25

条件分岐

変数の値によって, **結果が変わる!**

というようなこと (判断)

age の値が **12**以下 → **500** yen

13以上 → **1800** yen

実習

次のように入れて，実行ボタンをクリック．結果を確認

```
age = 18
```

```
if (age <= 12):
```

```
    print(500)
```

```
else:
```

```
    print(1800)
```

if (age <= 12)の直後に「:」
else の直後に「:」
(どちらも，コロン)

字下げも正確に！

print の前に，「タブ (Tab)」を1つだけ

```
1 age = 18
2 if (age <= 12):
3     print(500)
4 else:
5     print(1800)
```

正しくない字下げ

```
1 age = 18
2 if (age <= 12):
3     print(500)
4 else:
5     print(1800)
```

正しい字下げ

結果を確認

```
1 age = 18
2 if (age <= 12):
3     print(500)
4 else:
5     print(1800)
6
```

▶ 実行 (Ctrl-Enter)

出力 入力 コメント 0

1800

実習

18 を 7 に書き換えて、実行ボタンをクリック。結果を確認



```
1 age = 7
2 if (age <= 12):
3     print(500)
4 else:
5     print(1800)
6
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

500

The image shows a code editor window with a dark background. The code is written in Python. The first line is 'age = 7', which is highlighted with a red box. The code is an if-else statement: 'if (age <= 12): print(500) else: print(1800)'. Below the code is a green button with a white play icon and the text '実行 (Ctrl-Enter)'. Below the button is a tabbed interface with '出力' (Output) selected, '入力' (Input), and 'コメント' (Comments) with a '0' in a blue circle. The output area shows '500', which is also highlighted with a red box.

3-2 Python の配列と繰り返し (paiza.IOを使用)

配列

0	1	2	3	4
8	6	4	2	3

データの並びで、0から始まる番号
(添字) が付いている

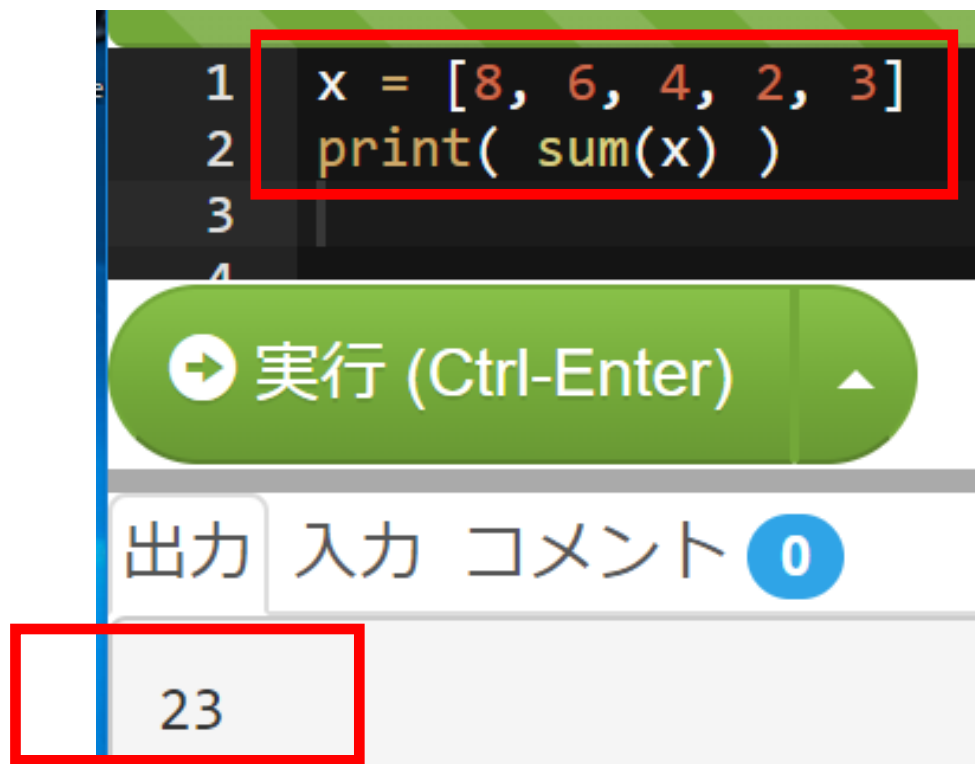
実習 1. 合計

8, 6, 4, 2, 3 というデータについて, **合計**を求めたい

次のように入れて, 実行ボタンをクリック. 結果 **23** を確認

```
x = [8, 6, 4, 2, 3]
```

```
print( sum(x) )
```



The screenshot shows a code editor with the following Python code:

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

Below the code is a green button labeled "実行 (Ctrl-Enter)".

Underneath the button is a tabbed interface with "出力" (Output) selected. The output area shows the result "23".

4つの価格について, 1.08倍を求める

0	1	2	3
180	20	250	40

配列データ

実習

次のように入れて、実行ボタンをクリック。結果を確認

```
p = [180, 20, 250, 40]
```

```
for i in p:
```

```
    print(i * 1.08)
```

★ 全員、この結果が
得られるのを待ちます。
仲間同士で助け
合ってください

```
1 p = [180, 20, 250, 40]
2 for i in p:
3     print(i * 1.08)
4
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

```
194.4
21.6
270.0
43.2
```

3-3. 「2つの水差し」 のシミュレーション

状態空間表現とは

水差し 2つ

① 4ガロン入り → 水の量は変化 (x とする)

② 3ガロン入り → 水の量は変化 (y とする)

変数は2個 x と y

状態空間表現とは

複数の変数で、**状態**を表現すること

ルール

1. **水差し①**を満杯にする
2. **水差し②**を満杯にする
3. **水差し①**を空にする
4. **水差し②**を空にする
5. **水差し①**を使って, **水差し②**を一杯にする
6. **水差し②**を使って, **水差し①**を一杯にする
7. **水差し①**の水をすべて, **水差し②**に入れる
8. **水差し②**の水をすべて, **水差し①**に入れる

ルールは全部で 8 個

実習タイム

① 先ほど用いた Web ページをもう一度開いてください

第3回授業

- 3-1 Python の変数と条件分岐, 3-2 Python の配列と繰り返し

paiza.IO のウェブページ: <https://paiza.io/>

ここで「Python 3」を選ぶ。表示が日本語になっている場合には、英語に変更

paiza.IO の参考資料: 「[paiza.IO を使ってみる](#)」

- 3-3 「2つの水差し」のシミュレーション

実習用のプログラムのソースコード

```
import sys
x, y = 0, 0
answer = [1, 2, 3, 4, 5, 6, 7, 8]
for i in answer:
    if i == 1 and x < 4:
        x, y = 4, y
    elif i == 2 and y < 3:
        x, y = x, 3
    elif i == 3 and x > 0:
        x, y = 0, y
    elif i == 4 and y > 0:
        x, y = x, 0
    elif i == 5 and (x + y) >= 4 and x < 4:
        x, y = 4, x + y - 4
    elif i == 6 and (x + y) >= 3 and y < 3:
        x, y = x + y - 3, 3
    elif i == 7 and (x + y) <= 4 and y > 0:
        x, y = x + y, 0
    elif i == 8 and (x + y) <= 3 and x < 0:
        x, y = 0, x + y
    print("%d %d" % (x, y))
```

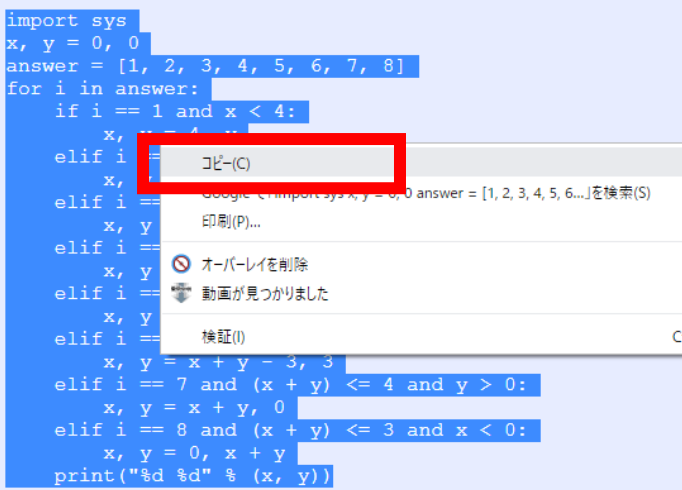
実習タイム

② プログラムを丸ごとコピー & ペースト. paiza.IO に.

第3回授業

- 3-1 配列と繰り返し
paiza.IO のウェブページ: <https://paiza.io/>
ここで「Python 3」を選ぶ。表示が日本語になっている場合には、英語に変える
paiza.IO の参考資料: 「[paiza.IO を使ってみる](#)」
- 実習プログラム

```
import sys
x, y = 0, 0
answer = [1, 2, 3, 4, 5, 6, 7, 8]
for i in answer:
    if i == 1 and x < 4:
        x, y = x + y - 3, 3
    elif i == 2 and (x + y) <= 4 and y > 0:
        x, y = x + y, 0
    elif i == 3 and (x + y) <= 3 and x < 0:
        x, y = 0, x + y
    print("%d %d" % (x, y))
```

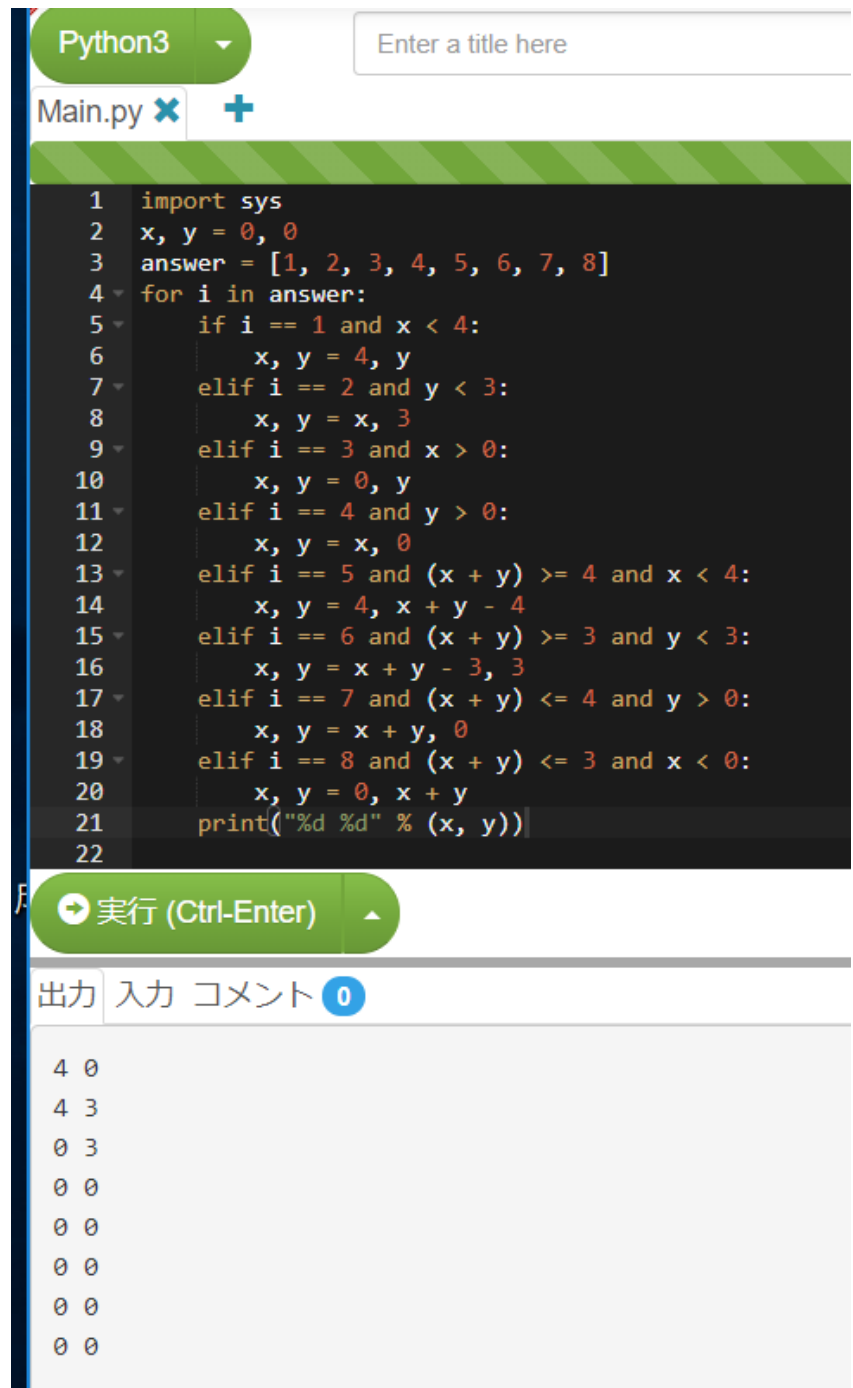


コピーしたい範囲をマウスで選び、
マウスの右クリックメニュー
で「コピー」が便利

マウスの右クリックメニュー
で「貼り付け」が便利

実習タイム

③ 「実行」をクリック



The screenshot shows a Python IDE interface. At the top, there is a dropdown menu set to 'Python3' and a text input field with the placeholder 'Enter a title here'. Below this, a tab labeled 'Main.py' is active. The main area contains a Python script with the following code:

```
1 import sys
2 x, y = 0, 0
3 answer = [1, 2, 3, 4, 5, 6, 7, 8]
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
8         x, y = x, 3
9     elif i == 3 and x > 0:
10        x, y = 0, y
11        elif i == 4 and y > 0:
12            x, y = x, 0
13        elif i == 5 and (x + y) >= 4 and x < 4:
14            x, y = 4, x + y - 4
15        elif i == 6 and (x + y) >= 3 and y < 3:
16            x, y = x + y - 3, 3
17        elif i == 7 and (x + y) <= 4 and y > 0:
18            x, y = x + y, 0
19        elif i == 8 and (x + y) <= 3 and x < 0:
20            x, y = 0, x + y
21        print("%d %d" % (x, y))
22
```

Below the code editor, there is a green button labeled '実行 (Ctrl-Enter)'. Underneath the button, there are three tabs: '出力' (Output), '入力' (Input), and 'コメント' (Comments), with a counter '0' next to 'コメント'. The '出力' tab is selected, showing the following output:

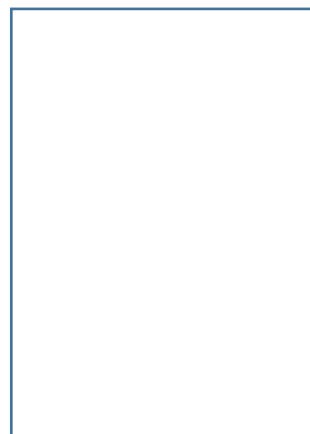
```
4 0
4 3
0 3
0 0
0 0
0 0
0 0
0 0
```


「ルール」とプログラムのソースコード

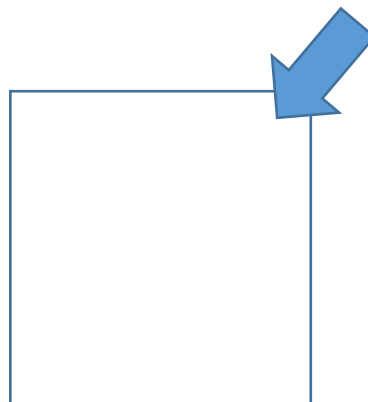
1. **水差し①**を満杯にする
2. **水差し②**を満杯にする
3. **水差し①**を空にする
4. **水差し②**を空にする
5. **水差し①**を使って、**水差し②**を一杯にする
6. **水差し②**を使って、**水差し①**を一杯にする
7. **水差し①**の水をすべて、**水差し②**に入れる
8. **水差し②**の水をすべて、**水差し①**に入れる

```
import sys
x, y = 0, 0
answer = [1, 2, 3, 4, 5, 6, 7, 8]
for i in answer:
    if i == 1 and x < 4:
        x, y = 4, y
    elif i == 2 and y < 3:
        x, y = x, 3
    elif i == 3 and x > 0:
        x, y = 0, y
    elif i == 4 and y > 0:
        x, y = x, 0
    elif i == 5 and (x + y) >= 4 and x < 4:
        x, y = 4, x + y - 4
    elif i == 6 and (x + y) >= 3 and y < 3:
        x, y = x + y - 3, 3
    elif i == 7 and (x + y) <= 4 and y > 0:
        x, y = x + y, 0
    elif i == 8 and (x + y) <= 3 and x < 0:
        x, y = 0, x + y
print("%d %d" % (x, y))
```

ルール2：水差し②を満杯にする



4 ガロン入り



3 ガロン入り

満杯にする

$$(x, y \mid \underline{y} < 3) \rightarrow (\underline{x}, \underline{3})$$

x は x のまま
 y は 3 に変化する

「 $y < 3$ 」のときにしか、
このルールは使えないよ

ルール2：水差し②を満杯にする

$(x, y \mid y < 3) \rightarrow (x, 3)$



```
if i == 1 and x < 4:
```

```
    x, y = 4, y
```

```
elif i == 2 and y < 3:
```

```
    x, y = x, 3
```

```
elif i == 3 and x > 0:
```

```
    x, y = 0, y
```

```
elif i == 4 and y > 0:
```

```
    x, y = x, 0
```

```
elif i == 5 and (x + y) >= 4 and x < 4:
```

```
    x, y = 4, x + y - 4
```

```
elif i == 6 and (x + y) >= 3 and y < 3:
```

```
    x, y = x + y - 3, 3
```

```
elif i == 7 and (x + y) <= 4 and y > 0:
```

```
    x, y = x + y, 0
```

```
elif i == 8 and (x + y) <= 3 and x < 0:
```

```
    x, y = 0, x + y
```

.....

コンピュータの
プログラムでは、
変数、条件分岐を使用

```
1 import sys
2 x, y = 0, 0
3 answer = [1, 2, 3, 4, 5, 6, 7, 8]
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
8         x, y = x, 3
9     elif i == 3 and x > 0:
10        x, y = 0, y
11    elif i == 4 and y > 0:
12        x, y = x, 0
13    elif i == 5 and (x + y) >= 4 and x < 4:
14        x, y = 4, x + y - 4
15    elif i == 6 and (x + y) >= 3 and y < 3:
16        x, y = x + y - 3, 3
17    elif i == 7 and (x + y) <= 4 and y > 0:
18        x, y = x + y, 0
19    elif i == 8 and (x + y) <= 3 and x < 0:
20        x, y = 0, x + y
21    print("%d %d" % (x, y))
22
```

ルール 1, 2, 3, 4, 5, 6, 7, 8
の順で行いたい

シミュレーション結果は x, y の変化の様子

```
1 import sys
2 x, y = 0, 0
3 answer = [1, 2, 3, 4, 5, 6, 7, 8]
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
8         x, y = x, 3
9     elif i == 3 and x > 0:
10        x, y = 0, y
11    elif i == 4 and y > 0:
12        x, y = x, 0
13    elif i == 5 and (x + y) >= 4 and x < 4:
14        x, y = 4, x + y - 4
15    elif i == 6 and (x + y) >= 3 and y < 3:
16        x, y = x + y - 3, 3
17    elif i == 7 and (x + y) <= 4 and y > 0:
18        x, y = x + y, 0
19    elif i == 8 and (x + y) <= 3 and x < 0:
20        x, y = 0, x + y
21        print("i: %d, x: %d, y: %d" % (i, x, y))
22
```

実行 (Ctrl-Enter)

実行ボタン



0 0 初期状態

↓ ルール 1

4 0 ↓ ルール 2

4 3 ↓ ルール 3

0 3 ↓ ルール 4

0 0 ↓ ルール 5

0 0 ↓ ルール 6

0 0 ↓ ルール 7

0 0 ↓ ルール 8

x, y の変化の様子

実習 1, 6, 3, 7 に書き換え, 実行しなさい

```
1 import sys
2 x, y = 0, 0
3 answer = [1, 6, 3, 7]
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
8         x, y = x, 3
9     elif i == 3 and x > 0:
10        x, y = 0, y
11    elif i == 4 and y > 0:
12        x, y = x, 0
13    elif i == 5 and (x + y) >= 4 and x < 4:
14        x, y = 4, x + y - 4
15    elif i == 6 and (x + y) >= 3 and y < 3:
16        x, y = x + y - 3, 3
17    elif i == 7 and (x + y) <= 4 and y > 0:
18        x, y = x + y, 0
19    elif i == 8 and (x + y) <= 3 and x < 0:
20        x, y = 0, x + y
21    print("%d %d" % (x, y))
22
```

実行 (Ctrl-Enter)



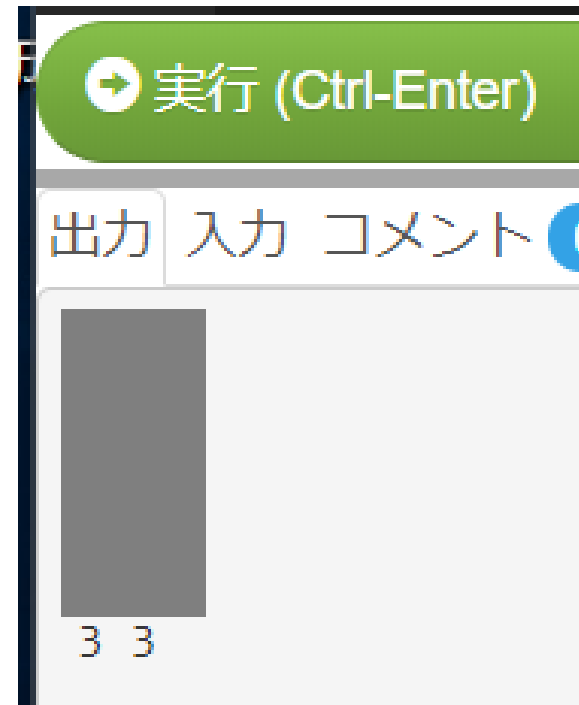
x y の変化の様子

実行ボタン

実習. 最後を [3 3] にできますか？

1. 水差し①を満杯にする
2. 水差し②を満杯にする
3. 水差し①を空にする
4. 水差し②を空にする
5. 水差し①を使って, 水差し②を一杯にする
6. 水差し②を使って, 水差し①を一杯にする
7. 水差し①の水をすべて, 水差し②に入れる
8. 水差し②の水をすべて, 水差し①に入れる

```
1 import sys
2 x, y = 0, 0
3 answer = 
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
```



x y の変化の様子

実習.

いろいろ試してみよう

- 最後を $[4\ 2]$ にできますか
- 最後を $[1\ 0]$ にできますか

まとめ

```
1 import sys
2 x, y = 0, 0
3 answer = [1, 2, 3, 4, 5, 6, 7, 8]
4 for i in answer:
5     if i == 1 and x < 4:
6         x, y = 4, y
7     elif i == 2 and y < 3:
8         x, y = x, 3
9     elif i == 3 and x > 0:
10        x, y = 0, y
11    elif i == 4 and y > 0:
12        x, y = x, 0
13    elif i == 5 and (x + y) >= 4 and x < 4:
14        x, y = 4, x + y - 4
15    elif i == 6 and (x + y) >= 3 and y < 3:
16        x, y = x + y - 3, 3
17    elif i == 7 and (x + y) <= 4 and y > 0:
18        x, y = x + y, 0
19    elif i == 8 and (x + y) <= 3 and x < 0:
20        x, y = 0, x + y
21    print("%d %d" % (x, y))
22
```



実行したいルールの列
(配列)

ルール

結果表示

「2つの水差し」
のシミュレーター

2つの変数 x, y を使用

3-4. 「2つの水差し」を 総当たりで解く

実習タイム

① 先ほど用いた Web ページをもう一度開いてください

```
x, y = x + y, 0
elif i == 8 and (x + y) <= 3 and x < 0:
    x, y = 0, x + y
print("%d %d" % (x, y))
```

- 3-4 「2つの水差し」を総当りで解く

実習用のプログラムのソースコード

```
import itertools
import sys
x, y = 0, 0
nsteps = 3
seq=[1,2,3,4,5,6,7,8]
for j in list(itertools.product(seq,repeat=nsteps)):
    for i in j:
        if i == 1 and x < 4:
            x, y = 4, y
        elif i == 2 and y < 3:
            x, y = x, 3
        elif i == 3 and x > 0:
            x, y = 0, y
        elif i == 4 and y > 0:
            x, y = x, 0
        elif i == 5 and (x + y) >= 4 and x < 4:
            x, y = 4, x + y - 4
        elif i == 6 and (x + y) >= 3 and y < 3:
            x, y = x + y - 3, 3
        elif i == 7 and (x + y) <= 4 and y > 0:
            x, y = x + y, 0
        elif i == 8 and (x + y) <= 3 and x < 0:
            x, y = 0, x + y
    print("%s %d %d" % (str(j), x, y))
```

3-4 . . .
の真下のもの

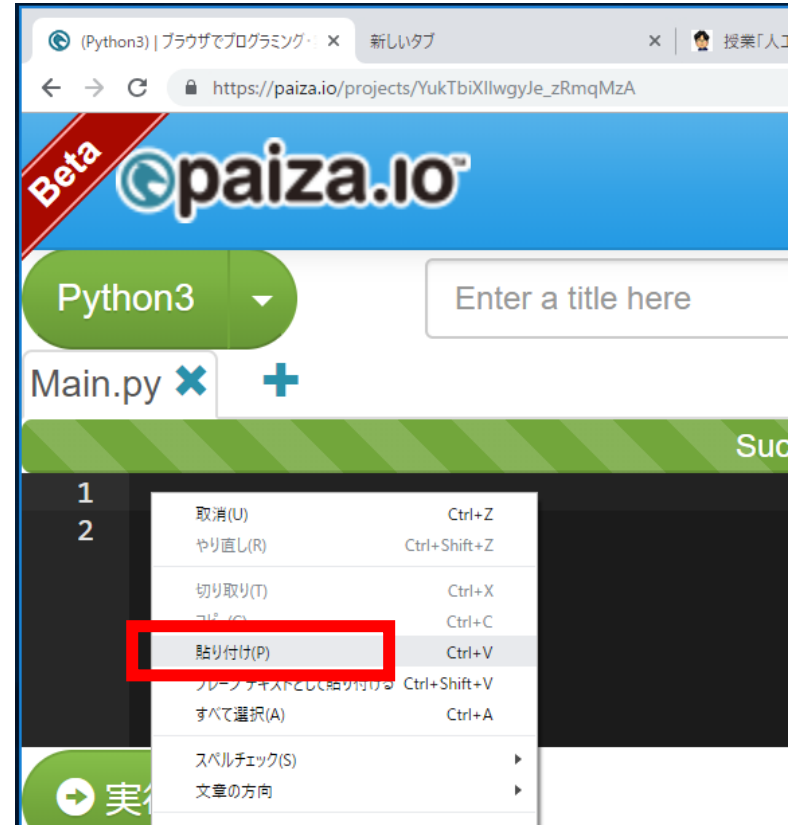
実習タイム

② プログラムを丸ごとコピー & ペースト. paiza.IO に.

```
elif i == 8 and (x + y) <= 3 and x < 0:  
    x, y = 0, x + y  
    print("%d %d" % (x, y))
```

• 3-4 「2つの水差し」を総当りで解く
実習用のプログラムのソースコード

```
import itertools  
import sys  
x, y = 0, 0  
nsteps = 3  
seq=[1,2,3,4,5,6,7,8]  
for j in list(itertools.product(seq, repeat=nsteps)):  
    for i in j:  
        if i == 1 and x < 4:  
            x, y = x + 1, y  
        elif i == 2 and x < 4 and y < 4:  
            x, y = x + 1, y + 1  
        elif i == 3 and x < 4 and y < 4:  
            x, y = x + 1, y + 1  
        elif i == 4 and x < 4 and y < 4:  
            x, y = x + 1, y + 1  
        elif i == 5 and x < 4 and y < 4:  
            x, y = x + 1, y + 1  
        elif i == 6 and (x + y) >= 3 and y < 3:  
            x, y = x + y - 3, 3  
        elif i == 7 and (x + y) <= 4 and y > 0:  
            x, y = x + y, 0  
        elif i == 8 and (x + y) <= 3 and x < 0:  
            x, y = 0, x + y  
    print("%s %d %d" % (str(j), x, y))
```



コピーしたい範囲をマウスで選び、マウスの右クリックメニューで「コピー」が便利

マウスの右クリックメニューで「貼り付け」が便利

実習タイム

③ 「実行」をクリック

```
3 x, y = 0, 0
4 nsteps = 3
5 seq=[1,2,3,4,5,6,7,8]
6 for j in list(itertools.product(seq,repeat=nsteps)):
7     for i in j:
8         if i == 1 and x < 4:
9             x, y = 4, y
10        elif i == 2 and y < 3:
11            x, y = x, 3
12        elif i == 3 and x > 0:
13            x, y = 0, y
14        elif i == 4 and y > 0:
15            x, y = x, 0
16        elif i == 5 and (x + y) >= 4 and x < 4:
17            x, y = 4, x + y - 4
18        elif i == 6 and (x + y) >= 3 and y < 3:
19            x, y = x + y - 3, 3
20        elif i == 7 and (x + y) <= 4 and y > 0:
21            x, y = x + y, 0
22        elif i == 8 and (x + y) <= 3 and x < 0:
23            x, y = 0, x + y
24 print("%s %d %d" % (str(j), x, y))
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

```
(1, 1, 1) 4 0
(1, 1, 2) 4 3
(1, 1, 3) 0 3
(1, 1, 4) 4 0
(1, 1, 5) 4 0
(1, 1, 6) 1 3
(1, 1, 7) 4 3
```

④ 結果を確認

(1, 1, 1) 4 0

(1, 1, 2) 4 3

(1, 1, 3) 0 3

(1, 1, 4) 4 0

1, 1, 1 の順だと 4 0

1, 1, 2 の順だと 4 3

1, 1, 3 の順だと 0 3

1, 1, 4 の順だと 4 0

⑤ 「nsteps = 3」

を「nsteps = 2」に書き換えて実行し，結果を確認
しなさい



```
1 import itertools
2 import sys
3 x, y = 0, 0
4 nsteps = 2
5 seq=[1,2,3,4,5,6,7,8]
6 for j in list(itertools.product(seq,repeat=nsteps)):
7     for i in j:
8         if i == 1 and x < 4:
9             x, y = 4, y
10        elif i == 2 and y < 3:
```

実行 (Ctrl-Enter)

出力	入力	コメント
(6, 6)	0 0	
(6, 7)	0 0	
(6, 8)	0 0	

全部を試しているので
総当たり

```
1 import itertools
2 import sys
3 x, y = 0, 0
4 nsteps = 2
5 seq=[1,2,3,4,5,6,7,8]
6 for j in list(itertools.product(seq,repeat=nsteps)):
7     for i in j:
8         if i == 1 and x < 4:
9             x, y = 4, y
10        elif i == 2 and y < 3:
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

```
(1, 1) 4 0
(1, 2) 4 3
(1, 3) 0 3
(1, 4) 4 0
(1, 5) 4 0
(1, 6) 1 3
(1, 7) 4 3
(1, 8) 4 3
(2, 1) 4 3
(2, 2) 4 3
(2, 3) 0 3
(2, 4) 0 0
(2, 5) 0 3
(2, 6) 0 3
(2, 7) 3 0
(2, 8) 3 3
(3, 1) 4 3
(3, 2) 0 3
(3, 3) 0 3
```


実習タイム

① 先ほど用いた Web ページをもう一度開いてください

- 3-4 「2つの水差し」を総当りで解く

2つ目。実習用のプログラムのソースコード

```
import itertools
import sys
x, y = 0, 0
nsteps = 3
goal = (3, 0)
seq=[1,2,3,4,5,6,7,8]
for j in list(itertools.product(seq,repeat=nsteps)):
    for i in j:
        if i == 1 and x < 4:
            x, y = 4, y
        elif i == 2 and y < 3:
            x, y = x, 3
        elif i == 3 and x > 0:
            x, y = 0, y
        elif i == 4 and y > 0:
            x, y = x, 0
        elif i == 5 and (x + y) >= 4 and x < 4:
            x, y = 4, x + y - 4
        elif i == 6 and (x + y) >= 3 and y < 3:
            x, y = x + y - 3, 3
        elif i == 7 and (x + y) <= 4 and y > 0:
            x, y = x + y, 0
        elif i == 8 and (x + y) <= 3 and x < 0:
            x, y = 0, x + y
    if ( goal == (x, y) ):
        print("%s %d %d" % (str(j), x, y))
```

「2つ目」
のもの

実習タイム

② プログラムを丸ごとコピー & ペースト. paiza.IO に.

• 3-4 「2つの水差し」を総当りで解く

2つ目。実習用のプログラムのソースコード

```
import itertools
import sys
x, y = 0, 0
nsteps = 3
goal = (3, 0)
seq=[1,2,3,4,5,6,7,8]
for j in list(itertools.product(seq, repeat=nsteps)):
    for i in j:
        if i == 1 and x < 4:
            x, y = 4, y
        elif i == 7 and (x + y) <= 4 and y > 0:
            x, y = x + y, 0
        elif i == 8 and (x + y) <= 3 and x < 0:
            x, y = 0, x + y
        if ( goal == (x, y) ):
            print("%s %d %d" % (str(j), x, y))
```

コピー(C)

Google (「import itertools import sys x, y = 0, 0 nsteps = ...」を検索(S))

印刷(P)...

オーバーレイを削除

動画が見つかりました

検証(I)

Ctrl



(Python3) | ブラウザでプログラミング · × 新しいタブ × | 授業「入

https://paiza.io/projects/YukTbiXIlwgyJe_zRmqMzA

Beta paiza.io™

Python3 Enter a title here

Main.py × +

実行

1 2

取消(U) Ctrl+Z

やり直し(R) Ctrl+Shift+Z

切り取り(T) Ctrl+X

コピー(C) Ctrl+C

貼り付け(P) Ctrl+V

プレーンテキストとして貼り付け Ctrl+Shift+V

すべて選択(A) Ctrl+A

スペルチェック(S)

文章の方向

コピーしたい範囲をマウスで選び、
マウスの右クリックメニュー
で「コピー」が便利

マウスの右クリックメニュー
で「貼り付け」が便利

実習タイム

③ 「実行」をクリック

```
1 import itertools
2 import sys
3 x, y = 0, 0
4 nsteps = 3
5 goal = (3, 0)
6 seq=[1,2,3,4,5,6,7,8]
7 for j in list(itertools.product(seq,repeat=nsteps)):
8     for i in j:
9         if i == 1 and x < 4:
10            x, y = 4, y
11        elif i == 2 and y < 3:
12            x, y = x, 3
13        elif i == 3 and x > 0:
14            x, y = 0, y
15        elif i == 4 and y > 0:
16            x, y = x, 0
17        elif i == 5 and (x + y) >= 4 and x < 4:
18            x, y = 4, x + y - 4
19        elif i == 6 and (x + y) >= 3 and y < 3:
20            x, y = x + y - 3, 3
21        elif i == 7 and (x + y) <= 4 and y > 0:
22            x, y = x + y, 0
23        elif i == 8 and (x + y) <= 3 and x < 0:
24            x, y = 0, x + y
25        if ( goal == (x, y) ):
26            print("%s %d %d" % (str(j), x, y))
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

```
(2, 2, 7) 3 0
(2, 3, 7) 3 0
(2, 5, 7) 3 0
(2, 6, 7) 3 0
(2, 7, 4) 3 0
(2, 8, 7) 3 0
```

操作回数は

3

ゴール状態は

(3, 0)

```
1 import itertools
2 import sys
3 x, y = 0, 0
4 nsteps = 3
5 goal = (3, 0)
6 seq=[1,2,3,4,5,6,7,8]
7 for j in list(itertools.product(s
8     for i in j:
9         if i == 1 and x < 4:
10             x, y = 4, y
```

実行 (Ctrl-Enter)		
出力	入力	コメント
(2, 2, 7)	3	0
(2, 3, 7)	3	0
(2, 5, 7)	3	0
(2, 6, 7)	3	0
(2, 7, 4)	3	0

(3, 0) になるような操作手順を
総当たりで発見

- 次のように設定したらどうなるか？

```
nsteps = 5  
goal = (2, 0)  
seq=[1 2 3 4 5 6 7 8]
```

- 次のように設定したらどうなるか？
(答えが見つからないケース)

```
nsteps = 4  
goal = (2, 0)  
seq=[1 2 3 4 5 6 7 8]
```