

mi-5. プロダクションシステム

人工知能を演習で学ぶシリーズ (5)

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

金子邦彦



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

アウトライン

5-1 繰り返し処理と終了条件

5-2 属性名と属性値によるデータ表現

5-3 プロダクションシステム

5-1 繰り返し処理と終了条件

繰り返し処理と終了条件

- 繰り返し処理

複数回繰り返される可能性のある処理

(例) **1引く**

- 終了条件

繰り返し処理の終わりを定めるもの

(例) **0未満になったらやめる**

(0以上なら, 繰り返し処理を続ける)

最初 5 とすると : $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$ (やめ)

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

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

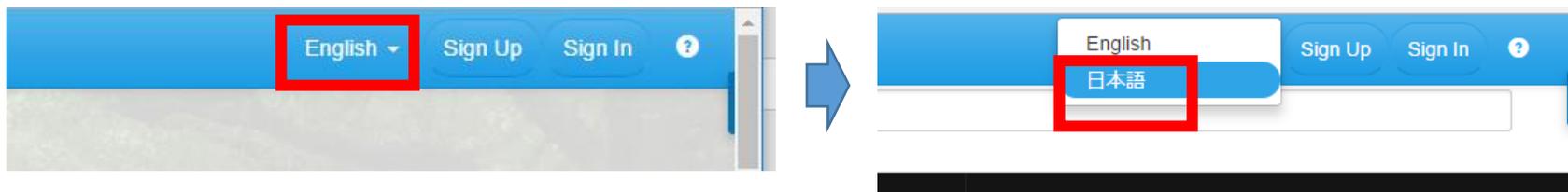
① ウェブブラウザを起動する

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)



パソコン演習

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

① プログラムを丸ごとコピー＆ペースト。 paiza.IO に。

第5回授業

• 5-1 繰り返しと終了処理

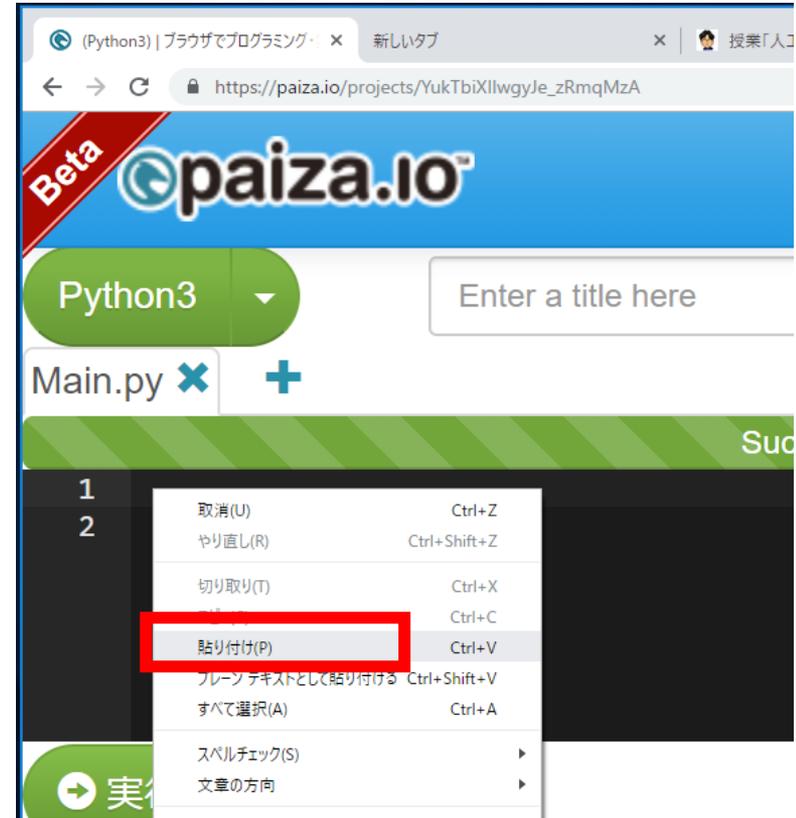
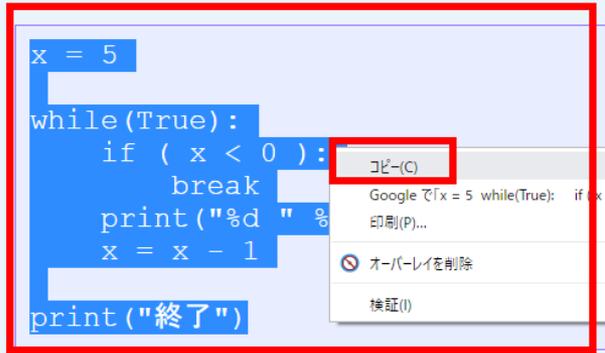
次のWebページを使って、実習を実施。

<https://paiza.io>

※ Internet Explorer を使ってうまく動かない場

実習用のプログラムのソースコード

```
x = 5
while(True):
    if ( x < 0 ):
        break
    print("%d " % x)
    x = x - 1
print("終了")
```



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

② 「実行」をクリック。結果を確認

```
1 x = 5
2
3 while(True):
4     if ( x < 0 ):
5         break
6         print("%d " % x, end="")
7         x = x - 1
8
9 print("終了")
10
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

5 4 3 2 1 0 終了

```
x = 5
```

```
while(True):
```

```
    if ( x < 0 ):
```

```
        break
```

```
    print("%d " % x, end="")
```

```
    x = x - 1
```

```
print("終了")
```

終了条件 $x < 0$

繰り返す処理

**x の値を表示,
x を 1 引く**

確認クイズ①

出力	入力	コメント
8 7 6 5 4 3 2 1 0		終了

このような表示を得たい

```
while(True):  
    if ( x < 0 ):  
        break  
    print("%d " % x, end="")  
    x = x - 1  
  
print("終了")
```

どう書き直すか？

次ページに続く

確認クイズ②

出力 入力 コメント 0

8 6 4 2 0 終了

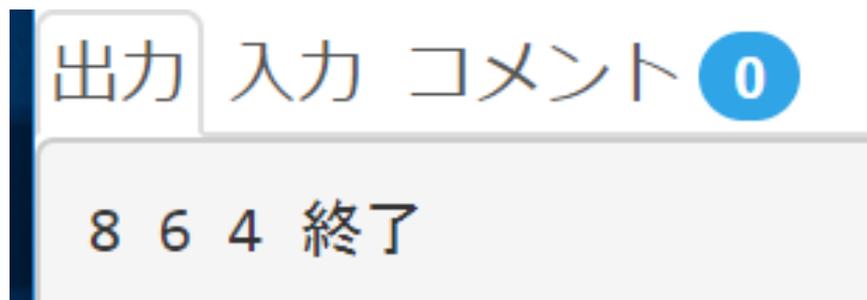
このような表示を得たい

```
while(True):  
    if ( x < 0 ):  
        break  
    print("%d " % x, end="")  
print("終了")
```

どう書き直すか？

次ページに続く

確認クイズ③



このような表示を得たい

```
while(True):  
    if ( ):  
        break  
    print("%d " % x, end="")  
  
print("終了")
```

どう書き直すか？

以上, 各自パソコンで試すこと.

5-2 属性名と属性値によるデータ表現

属性名と属性値による「知識のデータ」の表現

name	has
ichiro	ball

属性名

属性値

属性名と属性値による「知識のデータ」の表現

name	has
ichiro	ball

属性名

属性値

+

name	has
saburo	ball

属性名

属性値



属性名が同じときは合体

name	has
ichiro	ball
saburo	ball

nil

nil は「**何もない**」ことを示すキーワード

name	has
ichiro	ball
jiro	nil
saburo	ball
shiro	nil

確認クイズ

次の3つのデータを，1つにまとめなさい

name	has
A	pen

name	has
B	nil

name	has
C	apple



name	has

空欄に書き込み
みなさい

パソコン演習

1 つめ.

次の**データ**を, 変数 m に覚えさせるプログラム
確認表示まで

name	has
ichiro	ball
jiro	nil
saburo	ball
shiro	nil

2 つめ. ichiro の行を取り出す

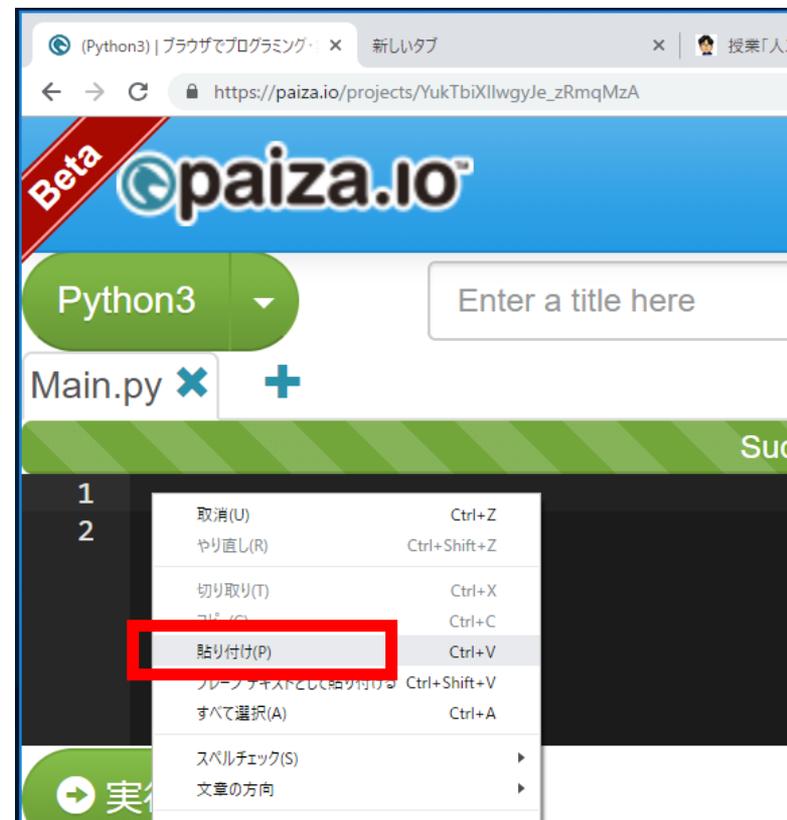
3 つめ. ball の行を取り出す

- ① 「5-2」 のところにある「1つめ」のプログラムを丸ごとコピー&ペースト。 paiza.IO に。

• 5-2 属性名と属性値による知識表現 (20 分)

1つめ

```
m = [{'name': 'ichiro', 'has': 'ball'},  
      {'name': 'jiro', 'has': 'nil'},  
      {'name': 'saburo', 'has': 'ball'},  
      {'name': 'shiro', 'has': 'nil'}]  
print(m)
```



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

② 「実行」をクリック。結果を確認

データを，変数 m に覚えさせるプログラム
確認表示まで



Success ツイート Share 0

```
1 m = [{'name': 'ichiro', 'has': 'ball'},
2       {'name': 'jiro', 'has': 'nil'},
3       {'name': 'saburo', 'has': 'ball'},
4       {'name': 'shiro', 'has': 'nil'}]
5 print(m)
6
```

実行 (Ctrl-Enter)

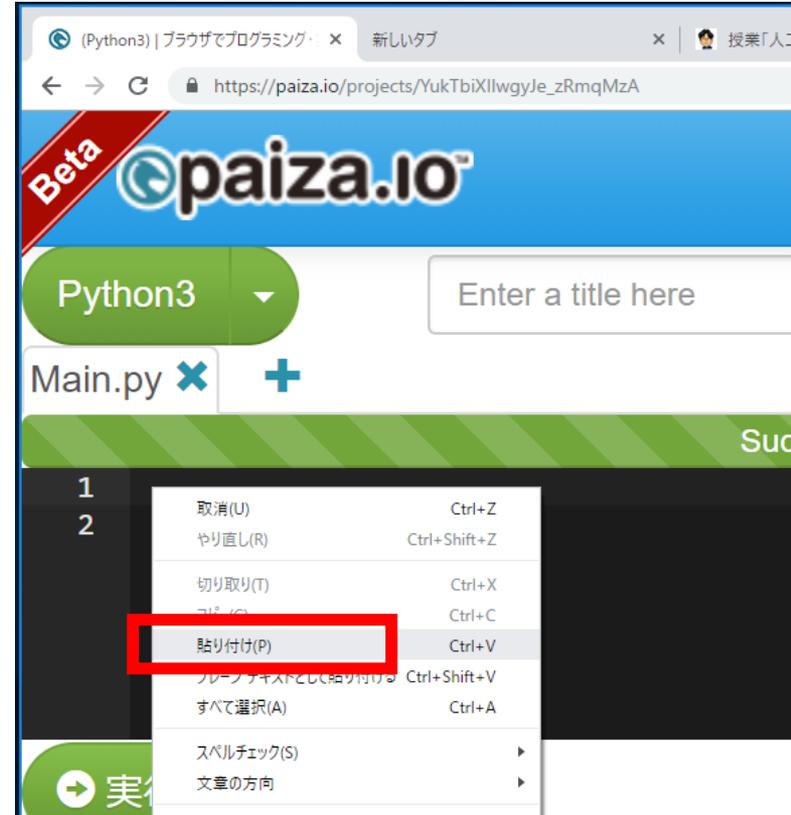
出力 入力 コメント 0 (0.04 sec)

[{'name': 'ichiro', 'has': 'ball'}, {'name': 'jiro', 'has': 'nil'},
{ 'name': 'saburo', 'has': 'ball'}, {'name': 'shiro', 'has': 'nil'}]

③ 「5-2」 のところにある「2つめ」の
プログラムを丸ごとコピー&ペースト。 paiza.IO に。

2つめ

```
m = [{"name": "ichiro", "has": "ball"},  
      {"name": "jiro", "has": "nil"},  
      {"name": "saburo", "has": "ball"},  
      {"name": "shiro", "has": "nil"}]  
for i in m:  
    if i['name'] == 'ichiro':  
        print(i)
```



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

④ 「実行」をクリック。結果を確認

データの中から「ichiro の行」を取り出すプログラム

The screenshot shows a code editor interface. At the top, there is a green bar with the text "Success" and social media sharing icons for Twitter and Facebook. Below this, the code is displayed on a dark background with line numbers 1 through 8. The code is as follows:

```
1 m = [{"name": "ichiro", "has": "ball"},
2       {"name": "jiro", "has": "nil"},
3       {"name": "saburo", "has": "ball"},
4       {"name": "shiro", "has": "nil"}]
5 for i in m:
6     if i['name'] == 'ichiro':
7         print(i)
8
```

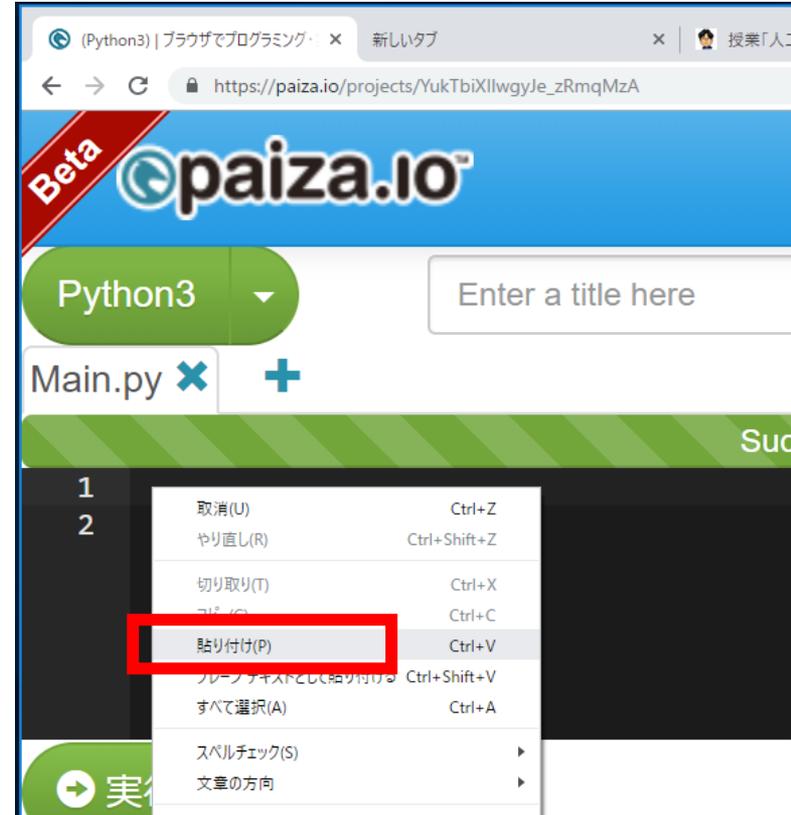
Below the code editor, there is a green button with a play icon and the text "実行 (Ctrl-Enter)". Underneath the button is a toolbar with icons for a user profile, information, settings, a clock, and a refresh icon. Below the toolbar, there are tabs for "出力" (Output), "入力" (Input), and "コメント" (Comments), with a blue circle containing the number "0" next to "コメント". The "出力" tab is selected, and the output is displayed in a light gray box:

```
{'name': 'ichiro', 'has': 'ball'}
```

- ⑤ 「5-2」のところにある「3つめ」のプログラムを丸ごとコピー&ペースト。 paiza.IO に。

3つめ

```
m = [{"name": "ichiro", "has": "ball"},  
      {"name": "jiro", "has": "nil"},  
      {"name": "saburo", "has": "ball"},  
      {"name": "shiro", "has": "nil"}]  
for i in m:  
    if i['has'] == 'ball':  
        print(i)
```



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

⑥ 「実行」をクリック。結果を確認

データの中から「ball の行」を取り出すプログラム



The screenshot shows a Python IDE interface. At the top, there is a dropdown menu set to "Python3" and a text input field labeled "Enter a title here". Below this, the file name "Main.py" is displayed with a close button and a plus sign. A green banner indicates "Success" with social media sharing options for "ツイート" (Twitter) and "Share". The main area contains a Python script:

```
1 m = [{"name": "ichiro", "has": "ball"},
2       {"name": "jiro", "has": "nil"},
3       {"name": "saburo", "has": "ball"},
4       {"name": "shiro", "has": "nil"}]
5 for i in m:
6     if i['has'] == 'ball':
7         print(i)
8
```

Below the code, a green button labeled "実行 (Ctrl-Enter)" is visible. A tooltip above it reads "スケジュール実行 / この機能は無料プランでご利用ください" (Schedule execution / This feature is available for free users). At the bottom, there are tabs for "出力" (Output), "入力" (Input), and "コメント" (Comments). The "出力" tab is active, showing the following output:

```
{'name': 'ichiro', 'has': 'ball'}
{'name': 'saburo', 'has': 'ball'}
```

確認クイズ

出力 | 入力 | コメント **0**

```
{'name': 'jiro', 'has': 'nil'}
```

属性名 `name` の
属性値が `jiro` の行を
取り出したい

```
m = [{"name": "ichiro", "has": "ball"},  
      {"name": "jiro", "has": "nil"},  
      {"name": "saburo", "has": "ball"},  
      {"name": "shiro", "has": "nil"}]  
for i in m:  
    if i['name'] == '':  
        print(i)
```

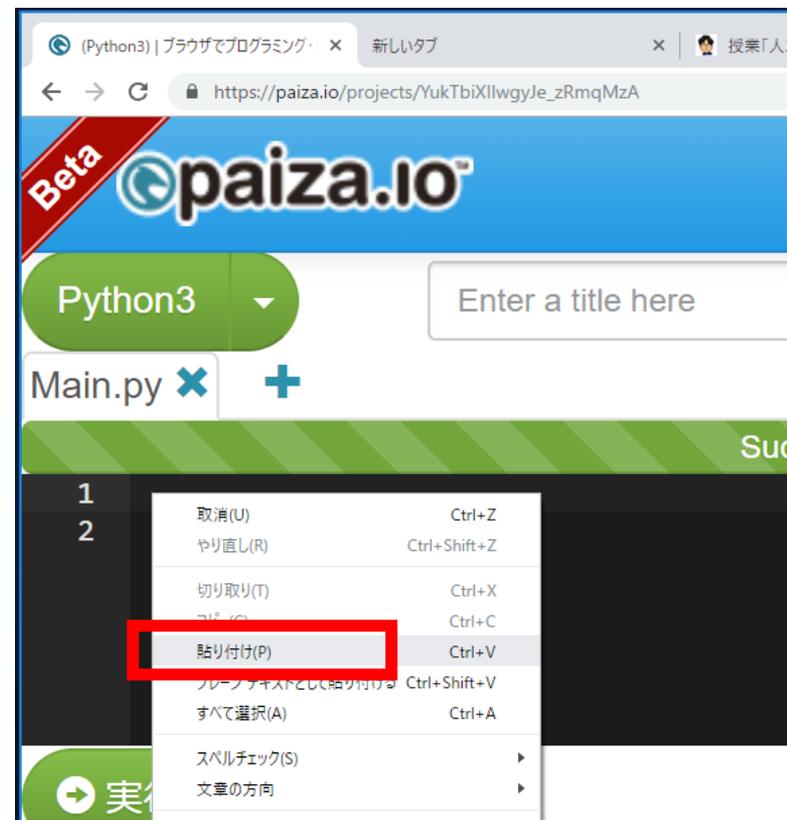
どう書き直すか？

- ① 「5-2」 のところにある「1つめ」のプログラムを丸ごとコピー&ペースト。 paiza.IO に。

• 5-2 属性名と属性値による知識表現 (20 分)

1つめ

```
m = [{'name': 'ichiro', 'has': 'ball'},  
      {'name': 'jiro', 'has': 'nil'},  
      {'name': 'saburo', 'has': 'ball'},  
      {'name': 'shiro', 'has': 'nil'}]  
print(m)
```



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

② 「実行」をクリック。結果を確認

データを，変数 `m` に覚えさせるプログラム
確認表示まで



```
Success ツイート Share 0
1 m = [{'name': 'ichiro', 'has': 'ball'},
2     {'name': 'jiro', 'has': 'nil'},
3     {'name': 'saburo', 'has': 'ball'},
4     {'name': 'shiro', 'has': 'nil'}]
5 print(m)
6
```

実行 (Ctrl-Enter)

出力 入力 コメント 0 (0.04 sec)

```
[{'name': 'ichiro', 'has': 'ball'}, {'name': 'jiro', 'has': 'nil'},
{'name': 'saburo', 'has': 'ball'}, {'name': 'shiro', 'has': 'nil'}]
```

5-3 プロダクションシステム

プロダクションシステム

プロダクションシステムとは、

ルールの記述とそれに基づく推論を実行するための枠組み



ノートページ

ルールの例：

一郎がボールを持っていれば、一郎は次郎にボールを渡す

ルールによる「知識のデータ」の変化

一郎がボールを持っていれば、一郎は次郎にボールを渡す

name	has
ichiro	ball
jiro	nil
saburo	ball
shiro	nil



name	has
ichiro	nil
jiro	ball
saburo	ball
shiro	nil

ルールが適用されたことによる
「知識のデータ」の変化

確認クイズ

saburo が ball を持っていれば

→ **saburo** は **shiro** に ball を渡す

name	has
ichiro	ball
jiro	nil
saburo	ball
shiro	nil



name	has
ichiro	ball
jiro	nil
saburo	
shiro	

ルールが適用されたことによる
データの変化

ルールのプログラム化

一郎がボールを持っていれば、一郎は次郎にボールを渡す

```
m = [{"name": "ichiro", "has": "ball"},  
      {"name": "jiro", "has": "nil"},  
      {"name": "saburo", "has": "ball"},  
      {"name": "shiro", "has": "nil"}]
```

ルール

```
for i in m:
```

```
    if i['name'] == 'ichiro' and i['has'] == 'ball':
```

```
        for j in m:
```

```
            if j['name'] == 'ichiro':
```

```
                j['has'] = 'nil'
```

```
            if j['name'] == 'jiro':
```

```
                j['has'] = 'ball'
```

データの変化

```
print(m)
```

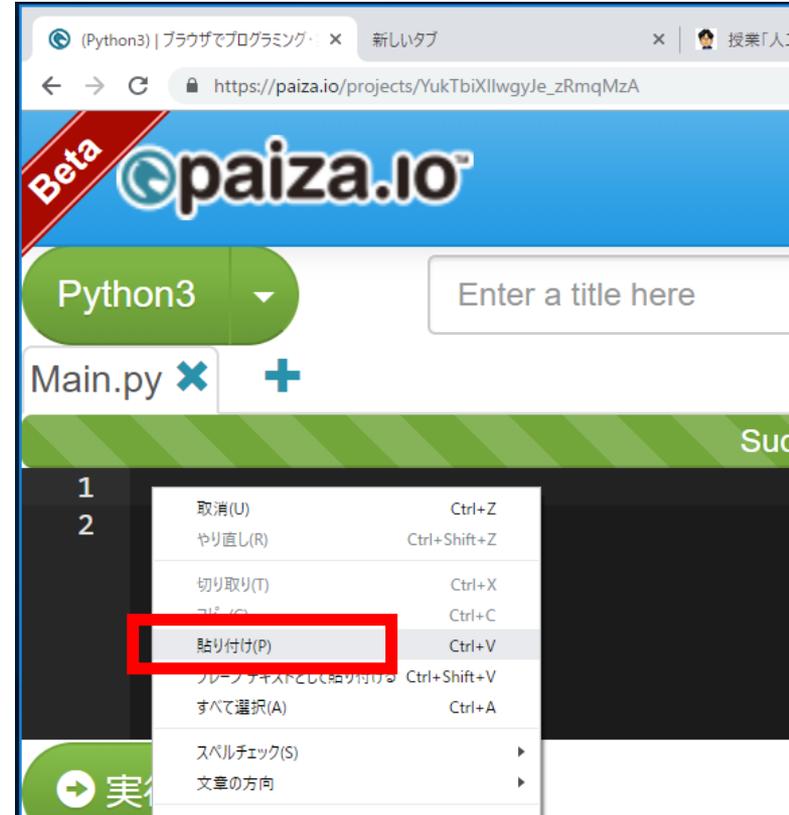
① 「5-3」 のところ

プログラムを丸ごとコピー＆ペースト。 paiza.IO に。

• 5-3 プロダクションシステム

1つのルールを持つプロダクションシステム

```
m = [{"name": "ichiro", "has": "ball"},  
      {"name": "jiro", "has": "nil"},  
      {"name": "saburo", "has": "ball"},  
      {"name": "shiro", "has": "nil"}]  
  
for i in m:  
    if i['name'] == 'ichiro' and i['has'] == 'ball':  
        for j in m:  
            if j['name'] == 'ichiro':  
                j['has'] = 'nil'  
            if j['name'] == 'jiro':  
                j['has'] = 'ball'  
  
print(m)
```



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

② 「実行」をクリック. 結果を確認

一郎がボールを持っていれば、一郎は次郎にボールを渡す

```
1 m = [{"name": "ichiro", "has": "ball"},
2     {"name": "jiro", "has": "nil"},
3     {"name": "saburo", "has": "ball"},
4     {"name": "shiro", "has": "nil"}]
5
6 for i in m:
7     if i['name'] == 'ichiro' and i['has'] == 'ball':
8         for j in m:
9             if j['name'] == 'ichiro':
10                j['has'] = 'nil'
11            if j['name'] == 'jiro':
12                j['has'] = 'ball'
13 print(m)
```

実行 (Ctrl-Enter)

出力 入力 コメント 0 (0.03 sec)

```
{'name': 'ichiro', 'has': 'nil'}, {'name': 'jiro', 'has': 'ball'},
{'name': 'saburo', 'has': 'ball'}, {'name': 'shiro', 'has': 'nil'}
```

パソコン演習

saburo が ball を持っていれば

→ saburo は shiro に ball を渡す

今のプログラムを書き換えなさい

開始前に, (次ページ)を確認しなさい

動作を各自で確認

```
1 m = [{"name": "ichiro", "has": "ball"},
2       {"name": "jiro", "has": "nil"},
3       {"name": "saburo", "has": "ball"},
4       {"name": "shiro", "has": "nil"}]
5
6 for i in m:
7     if i['name'] ==  and i['has'] == 'ball':
8         for j in m:
9             if j['name'] == :
10                j['has'] = 'nil'
11                if j['name'] == :
12                    j['has'] = 'ball'
13 print(m)
```

実行 (Ctrl-Enter)



出力 入力 コメント 0

(0.04 sec)

[{'name': 'ichiro', 'has': 'ball'}, {'name': 'jiro', 'has': 'nil'},
{'name': 'saburo', 'has': 'nil'}, {'name': 'shiro', 'has': 'ball'}]

プロダクションシステム

ルールは、あてまめる**条件**と、条件が成立したときのデータの**変化**を定めるもの

- **条件**

例 **ichiro** が **ball** を持っていれば

- **変化**

例 **saburo** は **shiro** に **ball** を渡す

データは、メモリに記憶され、**ルール**の適用により変化

※ ここでの**データ**は、**状態**（誰が何を持っているか）を記述したデータのこと

プロダクションシステム

- **知識のデータ**と**ルール**により, 推論を実行
- **ルール**の適用により, **知識のデータ**は変化する
- **プロダクションシステム**では, **知識のデータ**を変化させて, 所定のゴール状態にたどり着くかを確認することができる。

ルールの例 :

ichiro が ball を持っていれば
→ ichiro は jiro に ball を渡す

プロダクションシステム

- **ルール**は複数設定可能
- **プロダクションシステム**では、**ルール**はすべて適用される。
 - ※ 「どのルールを使うか」という**探索**はない

プロダクションシステムでは、データを変化させて、所定の**ゴール状態**にたどり着くかを確認する

- ※ たどり着くか、つかないかを調べる。
探索ではない