

遊んでみよう(修正)

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
```

← 画面に表示させる
おまじない

プロットするデータをaに作る

```
a = pd.DataFrame(index=[1, 2, 3, 4, 5], columns=['身長', '体重'])
for i in [1, 2, 3, 4, 5]:
    a.at[i, '身長'] = i
    a.at[i, '体重'] = i**2
print(a)
```

⇒ a.loc[i, '身長'] = i
a.loc[i, '体重'] = i**2

```
a.plot.scatter(x='身長', y='体重') ← 散布図を描く
plt.title('身長と体重')
plt.xlabel('身長')
plt.ylabel('体重')
plt.show()
```

aのカラム名に直す

この辺は適宜書き直す

試してみよう

(時間があれば) 遊んでみよう(修正)

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
```

← 画面に表示させる
おまじない

プロットするデータをaに作る

```
ix = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] range(10)
a = pd.DataFrame(index=ix, columns=['x', 'y'])
for i in ix:
    a.at[i, 'x'] = i
    a.at[i, 'y'] = i**2
print(a)
```

⇒ a.loc[i, 'x'] = i
a.loc[i, 'y'] = i**2

```
a.plot.scatter(x='身長', y='体重') ← 散布図を描く
plt.title('身長と体重')
plt.xlabel('身長')
plt.ylabel('体重')
plt.show()
```

aのカラム名に直す

この辺は適宜書き直す

試してみよう

(時間があれば) 遊んでみよう

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
```

← 画面に表示させる
おまじない

```
import math
```

```
# プロットするデータをaに作る
```

```
ix = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] range(10)
a = pd.DataFrame(index=ix, columns=['x', 'y'])
for i in ix:
    a.at[i, 'x'] = i
    a.at[i, 'y'] = i**2
```

⇒ a.loc[i, 'x'] = i
a.loc[i, 'y'] = i**2
math.exp(i)

```
print(a)
a.plot.scatter(x='身長', y='体重')
plt.title('身長と体重')
plt.xlabel('身長')
plt.ylabel('体重')
plt.show()
```

散点図を描く

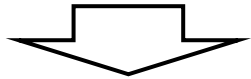
aのカラム名に直す

時間があれば
試してみよう

この辺は適宜書き直す

プログラムで遊ぶ1 ①

プログラムは（ほとんど）
代入・if・forと関数呼出し
で作れる



いろいろと
やってみよう

簡単なじゃんけんゲーム？

プレイヤーとコンピュータが
じゃんけんをする
プレイヤーは手を入力する
コンピュータは手を
ランダムに生成する
勝ち負けを判定し表示

外枠の構造

ループで繰り返す

```
y=コンピュータの手を生成  
（後出しじゃないよ！）  
x=プレイヤーの手を入力
```

もしxとyが引分けなら:

```
print('あいこ')
```

もしxがyに勝てば:

```
print('私の勝ち')
```

もしxがyに負ければ:

```
print('あなたの勝ち')
```

プログラムで遊ぶ1 ②

プレイヤーの手を入力？

input文を使おう (第2回)

```
y = int(input('促進文字列'))
```



少し試してみよう

```
y = int(input('入力せよ'))
```

```
print(y)
```

コンピュータの手を生成

乱数を発生させよう

手(グー・チョキ・パー)は

パー=1、チョキ=2、グー=3

で表そう



```
x = randint(1,3)    1~3の乱数
```

```
print(x)
```

ループで繰り返す

```
y=コンピュータの手を生成  
(後出しじゃないよ！)
```

```
x=プレイヤーの手を入力
```

もしxとyが引分けなら:

```
print('あいこ')
```

もしxがyに勝てば:

```
print('私の勝ち')
```

もしxがyに負ければ:

```
print('あなたの勝ち')
```

プログラムで遊ぶ1 ③

勝ち負けの判定？

xの値とyの値の組合せを
考えればいいだろう

パー=1、チョキ=2、グー=3

組合せを考えてみよう

x==1 and y==1なら あいこ

x==1 and y==2なら yの勝ち

... ..

ループで繰り返す

for文で書こう

繰り返し回数は適当に？

どうやって止める？

「9を入力したら終わり」とする
(入力を見て判定する)

ループで繰り返す

y=コンピュータの手を生成
(後出しじゃないよ！)

x=プレイヤーの手を入力

もしxとyが引分けなら:

print('あいこ')

もしxがyに勝てば:

print('私の勝ち')

もしxがyに負ければ:

print('あなたの勝ち')

プログラムを紙と
鉛筆で考えてみよう

プログラムで遊ぶ1 ③-2

```
import random
for i in range(10000):
    y = random.randint(1,3)
    x = int(input('パー=1、チョキ=2、グー=3、終了
    =9>¥n'))
    if x==9:
        print('終わり')
        break
    if (x==1 and y==1) or (x==2 and y==2) or (x==3
    and y==3):
        print('私=', y, 'あいこ')
    elif (x==1 and y==3) or (x==2 and y==1) or
    (x==3 and y==2):
        print('私=', y, 'あなたの勝ち')
    elif (x==1 and y==2) or (x==2 and y==3) or
    (x==3 and y==1):
        print('私=', y, '私の勝ち')
```

ループで繰り返す

y=コンピュータの手を生成
(後出しじゃないよ!)

x=プレイヤーの手を入力

もしxとyが引分けなら:

print('あいこ')

もしxがyに勝てば:

print('私の勝ち')

もしxがyに負ければ:

print('あなたの勝ち')

プログラムをまず紙と
鉛筆で考えてみよう

コンピュータで試そう
⇒ TAチェック

プログラムで遊ぶ 1 ③-3

```
import random
xwin = 0
ywin = 0
even = 0
for i in range(10000):
    y = random.randint(1,3)
    x = input('パー=1、チョキ=2、グー=3、終り=9¥n')
    x = int(x)
    if x==9:
        print('終わり', 'あなたの勝ち=', xwin, '私の勝ち', ywin, 'あいこ', even)
        break
    if (x==1 and y==1) or (x==2 and y==2) or (x==3 and y==3):
        print('私=', y, 'あいこ')
        even = even + 1
    elif (x==1 and y==3) or (x==2 and y==1) or (x==3 and y==2):
        print('私=', y, 'あなたの勝ち')
        xwin = xwin + 1
    elif (x==1 and y==2) or (x==2 and y==3) or (x==3 and y==1):
        print('私=', y, '私の勝ち')
        ywin = ywin + 1
```


プログラムで遊ぶ1 ④

三目並べ

3 × 3 の碁盤

先に3つ並んだほうが勝ち

プレイヤーとコンピュータが
交互に石 (oとx) を打つ

プレイヤーは手を入力する

手は(x, y)座標

コンピュータは手を

ランダムに生成する

勝ち負けを判定し表示

外枠の構造

盤面を初期化

ループで繰り返す

x, y = プレイヤーの手を入力

もし空いてなければやり直し

空いていれば石 o を置く

y = コンピュータの手を生成

もし空いてなければやり直し

空いていれば石 x を置く

盤面を表示

もし o が3つ並んだところがあれば
プレイヤーの勝ち。終了

もし x が3つ並んだところがあれば
コンピュータの勝ち。終了

プログラムで遊ぶ1 ⑤

外枠の構造

プレイヤーの手を入力

inputで2つ(xとy座標)入力

```
x, y = (int(u) for u in input('x y>').split())
```

inputで文字列 '2 1'を入力

.split()で、空白区切りで分割

リスト['2', '1']になる

(int(u) for u in ['2', '1'])で

要素ごとにint化して返す

⇒ x, y = (2, 1)になる

盤上(x,y)が空(' ')なら

```
b.loc[x,y] = 'o'
```

コンピュータの手を生成

1~3の整数乱数を発生⇒xとyに

上と同じに、盤上(x,y)が空(' ')なら

```
b.loc[x,y] = 'x'
```

盤面を初期化

ループで繰り返す

x, y = プレイヤーの手を入力

もし空いてなければやり直し

空いていれば石 o を置く

y = コンピュータの手を生成

もし空いてなければやり直し

空いていれば石 x を置く

盤面を表示

もし o が3つ並んだところがあれば
プレイヤーの勝ち。終り

もし x が3つ並んだところがあれば
コンピュータの勝ち。終り

プログラムで遊ぶ 1⑤-2

外枠の構造

盤面を初期化
ループで繰り返す

x, y = プレイヤーの手を入力
もし空いてなければやり直し
空いていれば石 o を置く
y = コンピュータの手を生成
もし空いてなければやり直し
空いていれば石 x を置く
盤面を表示

もし o が3つ並んだところがあれば
プレイヤーの勝ち。終了

もし x が3つ並んだところがあれば
コンピュータの勝ち。終了

```
if (
(b.loc[1,1]=='o' and b.loc[1,2]=='o' and b.loc[1,3]=='o') or ¥
(b.loc[2,1]=='o' and b.loc[2,2]=='o' and b.loc[2,3]=='o') or ¥
(b.loc[3,1]=='o' and b.loc[3,2]=='o' and b.loc[3,3]=='o') or ¥
(b.loc[1,1]=='o' and b.loc[2,1]=='o' and b.loc[3,1]=='o') or ¥
(b.loc[1,2]=='o' and b.loc[2,2]=='o' and b.loc[3,2]=='o') or ¥
(b.loc[1,3]=='o' and b.loc[2,3]=='o' and b.loc[3,3]=='o') or ¥
(b.loc[1,1]=='o' and b.loc[2,2]=='o' and b.loc[3,3]=='o') or ¥
(b.loc[1,3]=='o' and b.loc[2,2]=='o' and b.loc[3,1]=='o')):
    print(b)
    print('あなたの勝ち')
    break
```

```
elif (
(b.loc[1,1]=='x' and b.loc[1,2]=='x' and b.loc[1,3]=='x') or ¥
(b.loc[2,1]=='x' and b.loc[2,2]=='x' and b.loc[2,3]=='x') or ¥
(b.loc[3,1]=='x' and b.loc[3,2]=='x' and b.loc[3,3]=='x') or ¥
(b.loc[1,1]=='x' and b.loc[2,2]=='x' and b.loc[3,3]=='x') or ¥
(b.loc[1,1]=='x' and b.loc[2,1]=='x' and b.loc[3,1]=='x') or ¥
(b.loc[1,2]=='x' and b.loc[2,2]=='x' and b.loc[3,2]=='x') or ¥
(b.loc[1,3]=='x' and b.loc[2,3]=='x' and b.loc[3,3]=='x') or ¥
(b.loc[1,1]=='x' and b.loc[2,2]=='x' and b.loc[3,3]=='x') or ¥
(b.loc[3,1]=='x' and b.loc[2,2]=='x' and b.loc[1,3]=='x')):
    print(b)
    print('私の勝ち')
    break
```

プログラムで遊ぶ1 ⑥

```
import pandas as pd
import random

b = pd.DataFrame([[[' ',' ',' '],[' ',' ',' '],[' ',' ',' ']], ¥
                 index=[1,2,3], columns=[1,2,3])

for i in range(10000):
    # プレイヤーが打つ
    for j in range(100):
        x, y = (int(u) for u in input('x y>').split())
        if (x<1 or 3<x or y<1 or 3<y):
            print('値が不正です')
        if b.loc[x,y]!=' ':
            print('既に存在します')
        else:
            break
    b.loc[x,y] = 'o'

    # コンピュータが打つ
    for j in range(100):
        x = random.randint(1,3)
        y = random.randint(1,3)
        if b.loc[x,y]==' ':
            break
    b.loc[x,y] = 'x'
    # 画面表示
    print(b, '¥n')
```

```
if ((b.loc[1,1]=='o' and b.loc[1,2]=='o' and b.loc[1,3]=='o') or ¥
    (b.loc[2,1]=='o' and b.loc[2,2]=='o' and b.loc[2,3]=='o') or ¥
    (b.loc[3,1]=='o' and b.loc[3,2]=='o' and b.loc[3,3]=='o') or ¥
    (b.loc[1,1]=='o' and b.loc[2,1]=='o' and b.loc[3,1]=='o') or ¥
    (b.loc[1,2]=='o' and b.loc[2,2]=='o' and b.loc[3,2]=='o') or ¥
    (b.loc[1,3]=='o' and b.loc[2,3]=='o' and b.loc[3,3]=='o') or ¥
    (b.loc[1,1]=='o' and b.loc[2,2]=='o' and b.loc[3,3]=='o') or ¥
    (b.loc[1,3]=='o' and b.loc[2,2]=='o' and b.loc[3,1]=='o')):
    print(b)
    print('あなたの勝ち')
    break
elif ((b.loc[1,1]=='x' and b.loc[1,2]=='x' and b.loc[1,3]=='x') or ¥
      (b.loc[2,1]=='x' and b.loc[2,2]=='x' and b.loc[2,3]=='x') or ¥
      (b.loc[3,1]=='x' and b.loc[3,2]=='x' and b.loc[3,3]=='x') or ¥
      (b.loc[1,1]=='x' and b.loc[2,2]=='x' and b.loc[3,3]=='x') or ¥
      (b.loc[1,1]=='x' and b.loc[2,1]=='x' and b.loc[3,1]=='x') or ¥
      (b.loc[1,2]=='x' and b.loc[2,2]=='x' and b.loc[3,2]=='x') or ¥
      (b.loc[1,3]=='x' and b.loc[2,3]=='x' and b.loc[3,3]=='x') or ¥
      (b.loc[1,1]=='x' and b.loc[2,2]=='x' and b.loc[3,3]=='x') or ¥
      (b.loc[3,1]=='x' and b.loc[2,2]=='x' and b.loc[1,3]=='x')):
    print(b)
    print('私の勝ち')
    break
print('complete')
```

ファイルダウンロード

<http://pepper.is.sci.toho-u.ac.jp/DL/>へ行き、sanmoku2.ipynbを右クリック
「名前を付けて保存」⇒ E:¥WPy-3662¥notebooks¥sanmoku2.ipynb へ保存。
Jupyter Notebookでsanmoku2.ipynbが見えるので、クリックして開く。