

プログラミング演習

23j1-216

教科書P80-81

この時間の目標

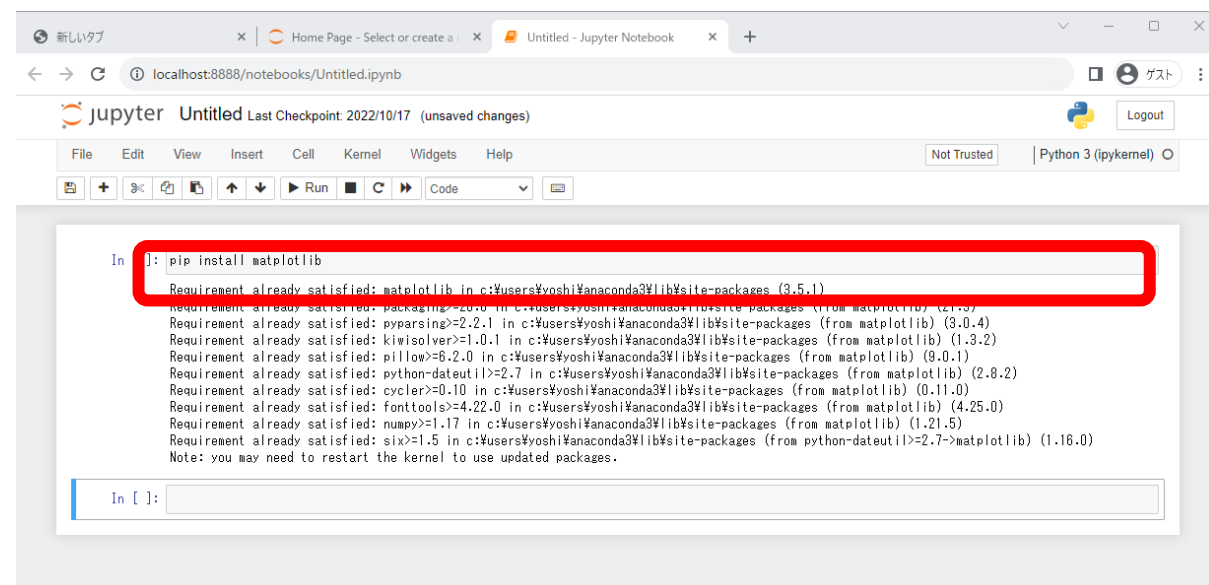
- プログラミング演習
- S プログラム作成の工夫がよく理解でき、手順を工夫して活用しようと思った
- A プログラム作成の工夫がよく理解できた
- B プログラム作成の工夫が理解できた
- C プログラム作成の工夫が理解できなかった

Jupyter Notebookにおまじない

1. Jupyter Notebookのセルに以下の文字列を入力

```
pip install matplotlib
```

2. [Shift]+[Enter]



```
In [ ]: pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\yoshi\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: packaging>=20.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: numpy>=1.17 in c:\users\yoshi\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: six>=1.5 in c:\users\yoshi\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

うまく動かないプログラム

サンプル1

- うまく動きません
 - なぜでしょうか
- 直すべきところはどこ？
 - 少なくとも3か所

```
import random
print('1は20点、2・3は5点、4・5・6は0点')
num=int(input('何回さいころを振りますか'))
for i in range(num):
    def sai():
        me=random.randint(1,6)
        return me
    if sai()==1:
        print(20,'点')
    elif 2<=sai()<=3:
        print(5,'点')
    else:
        print(0,'点')
```

プログラミング演習

ペアでプログラミング

- プログラミングでバトルだ！
- 進め方
 - お題に合うプログラムをペアで作成
 - できたらパワポに貼り付けて[F5]
 - チャットに「できた！〇〇」とパソコンの番号を投稿
 - できたペアはプログラムを改良・改造
 - 数ペアできたら解説

本日のお題

1. スロットマシン
 - 3つの数を生成
2. ジャンケンの勝敗
 - 簡単に判定
3. 確実に数値を入力させるプログラム
 - 違ったら再入力
4. コンピュータが当てる、数当てプログラム
 - 数を当てる手順

1. スロットマシン1

- 0~999の乱数を発生し変数Xに代入
- 100の位を変数x1、10の位をx2、1の位をx3に代入
- x1,x2,x3を表示

- 例 X=0 なら 0 0 0
X=234 なら 2 3 4

2.ジャンケンの勝敗

- 0:グー 1:チョキ 2:パー とする
- 勝敗を判定する式を考える

```
import random
A=int(input('0:グー 1:チョキ 2:パーを入力'))
B=random.randint(0,2)
hantei=['あいこ','勝ち','負け']
te=['グー','チョキ','パー']
```

この続きをつくる

1行でできるよ!

自分+相手		相手		
		0:グー	1:チョキ	2:パー
自分	0:グー	0	1	2
	1:チョキ	1	2	3
	2:パー	2	3	4

相手-自分		相手		
		0:グー	1:チョキ	2:パー
自分	0:グー	0	1	2
	1:チョキ	-1	0	1
	2:パー	-2	-1	0

自分-相手		相手		
		0:グー	1:チョキ	2:パー
自分	0:グー	0	-1	-2
	1:チョキ	1	0	-1
	2:パー	2	1	0

3. 確実に数値を入力させるプログラム

- 数値を入力させたくても文字を入力する人がいる
- それでもエラーを出さずに動かせるプログラムを書く

str.isdigit(文字列)関数
文字列が数値ならTrue,
そうでなければFalse

• 動作イメージ

2桁の数を入力してください

数値ではありません もう一度入力してください

5 ですね 2桁ではありません もう一度入力してください

32 ですね 入力ありがとうございます

4.コンピュータが当てる、数当てプログラム

- 利用者が2桁の数を1つ決める
- 質問をしてこの数を当てるプログラムを書く

- 動作イメージ

あなたが考えた数は20ですね

0:はい 1:大きい 2:小さい で回答してください

では、あなたが考えた数は60ですね

0:はい 1:大きい 2:小さい で回答してください