

# Python講座

2016/6/28(火)

※講座無し、次回は6/30(木)

# 今回の内容

- ▶ 演算子と式
- ▶ 複合代入演算子
- ▶ リスト
- ▶ リストに関する文
- ▶ 数字のみのリスト
- ▶ 辞書
- ▶ 比較演算子
- ▶ if文
- ▶ for文①②
- ▶ while文
- ▶ ループ文に関する文
- ▶ 条件式の詳細
- ▶ ユーザー定義関数
- ▶ 注意点
- ▶ turtleに関する文
- ▶ open()①②
- ▶ with文
- ▶ 読み込み・書き込み
- ▶ 練習問題①②③④⑤
- ▶ あとがき

# 演算子と式

- ▶ 加算  $a+b$  減算  $a-b$  乗算  $a*b$  除算  $a/b$
- ▶ べき乗を求める  $10^{**}3 >>> 1000$  (10の3乗)
- ▶ あまりを求める  $14\%3 >>> 2$  (4あまり2)
- ▶ 少数以下の切り捨て  $10//4 >>> 2$  (2.5の切り捨て)

※計算順序は数学と同じ、べき乗は乗算より先

# 複合代入演算子

- ▶ 変数に計算結果を代入する際、以下のように省略しながら書くことが可能

変数=変数 演算子 値 → 変数演算子=値

power=power+3 → power+=3

powerに+3した値をpowerに代入する

※変数に値が代入されている状態でのみ可能である

# リスト

- ▶ リスト名=[要素,要素,...]

list=[1 , 2 , 3 , 4]    A=["文字" , "moji"]

mix=[1 , "moji" , 3]

- ▶ リストの要素を変数として扱う

変数=リスト名[要素の番号]

※要素の番号の最初は0である

# リストに関する文

▶ `len(リスト名)`

リストの要素数を出力する

▶ `eval("[要素,要素,...]")`

" "の中でリストを書くことで、リストとして使用することができる

▶ `A = eval("[1,12,123,1234]")`

`print (len(A))`

`>>> 4`

# 数字のみのリスト

▶ `sum(リスト名)`   `min(リスト名)`  
`max(リスト名)`

▶ `s = [12,3,456,7,89]`  
`print ("{} , {} , {}".format(sum(s),min(s),max(s)))`  
`>>> 567 , 3 , 456`

# 辞書

- ▶ リストの要素に対して名前(数字でも可)を付けることで、番号ではなく名前呼び出す
- ▶ 辞書名={名前:要素, 名前:要素,...}
- ▶ MIX={"RPG":"FF", "卯月":4, 334:"阪神", 33:2}

```
print(MIX[334])           >>> 阪神
```

```
print(MIX["卯月"] * 2)    >>> 8
```

# 比較演算子

▶ 以降のページに出てくる条件式に対して使用する

| Pythonでの書き方            | 数学での書き方    | 意味               |
|------------------------|------------|------------------|
| <code>a == b</code>    | $a = b$    | aはbと等しい          |
| <code>a != b</code>    | $a \neq b$ | aはbと等しくない        |
| <code>a &lt; b</code>  | $a < b$    | aはbより小さい         |
| <code>a &gt; b</code>  | $a > b$    | aはbより大きい         |
| <code>a &lt;= b</code> | $a \leq b$ | aはbより小さいか等しい(以下) |
| <code>a &gt;= b</code> | $a \geq b$ | aはbより大きい等しい(以上)  |

# if文

- ▶ if 条件式: 内容

もし条件式が満たされたとき、内容を実行

- ▶ elif 条件式: 内容

以前のif,elifの条件式が満たされなかった場合かつ  
もし条件式が満たされたとき、内容を実行

- ▶ else: 内容

以前のif,elifの条件式が満たされなかった場合  
内容を実行

# for文①

- ▶ `for 変数a in range(変数b):` 内容  
変数b(整数)の数だけ、内容を実行する  
変数aの初期値は0、内容が実行されるたびに+1  
されていく →最終的には、変数a==変数b - 1
- ▶ `for x in range(5):`  
    `print(x)`  
>>>0 1 2 3 4

## for文②

▶ **for 変数 in リスト名:** 内容

**リストの要素数**だけ、内容を実行する

変数にはリストの要素が入り、内容が実行される  
たびに変数が更新される

▶ `list=[12,3,"moji",5]`

`for x in list:`

`print(x)`

`>>> 12 3 moji 5`

# while文

- ▶ while 条件式: 内容

条件式が満たされている間だけ、回数の限りなくずっと内容を実行し続ける

※条件式を間違えると、無限ループが起こる

- ▶ a=1

```
while a<5 :
```

```
    print(a)
```

```
    a+=2
```

```
>>>1 3
```

# ループ文に関する文

## ▶ break

ループをその場で終了させる

## ▶ continue

continue以降の内容を実行せず、ループを引き続き実行する

# 条件式の詳細

- ▶ 条件式から得られるものは、TrueかFalse
- ▶ 条件式が満たされている場合はTrue(真)
- ▶ 条件式が満たされていない場合はFalse(偽)
- ▶ 条件式から得られたものによって、if文やwhile文の処理の有無が変わる

# ユーザー定義(自作)関数

- ▶ 以下の作業を「関数を定義する」という
- ▶ `def` 関数名(変数1):      ▶ `def` 関数名(変数1):  
    内容                                      内容  
                                                `return` 変数2
- ▶ 関数名はわかりやすいものが好ましい
- ▶ 変数1は引数(ひきすう)と言い、内容で使用できる
- ▶ `return`でこの関数で返却するもの(変数2)を決める

# 自作関数(注意点)

- ▶ 関数内にreturnは複数書けるが、返却できるものは**1つのみ**
  - if文などで指定する場合が多い
- ▶ **関数外で使用している変数を関数内**では使用不可
  - 使う場合は**引数**で関数内に引っ張ってくる
- ▶ **関数内で使用している変数を関数外**では使用不可
  - 使う場合は**return**で関数外に返却
  - 逆に内外で変数名が同じでも問題なし

# turtleに関する文

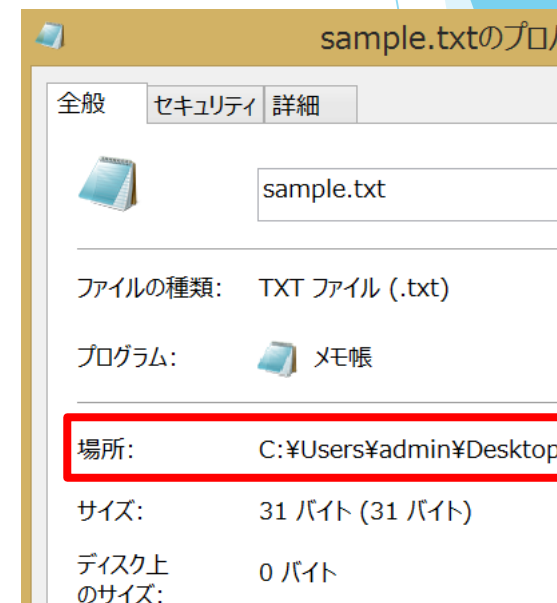
▶ <http://docs.python.jp/3.3/library/turtle.html>

|               |              |
|---------------|--------------|
| t.forward(x)  | まっすぐx進む      |
| t.left(x)     | 反時計回りにx度回転   |
| t.right(x)    | 時計回りにx度回転    |
| t.goto(x,y)   | 座標(x,y)に進む   |
| t.circle(r,x) | 半径rを本体左にx度描画 |
| t.up()        | 描画をしない       |
| t.down()      | 描画をする        |
| t.reset       | 白紙にする        |

# open()①

- ▶ 変数 = `open(場所, モード)`
- ▶ 場所：ファイルの場所とファイル名を指定  
使いたいファイルのプロパティから参照
- ▶ 例：`C:¥Users¥admin¥Desktop`
- ▶ 書き方は二通りある

```
file = open(r"C:\Users\admin\Desktop\sample.txt")  
file = open("C:\\Users\\admin\\Desktop\\sample.txt")
```



# open()②

- ▶ 変数 = open(場所, モード)
- ▶ モード：指定したファイルをどのように使用するかを決める、無記入の場合は読み取り("r")

|   |                                           |
|---|-------------------------------------------|
| r | 読み込み専用。書き込みはできない。<br>ファイルが存在しないときはエラーになる。 |
| w | 書き込み専用。読み込みはできない。<br>ファイルが存在していない場合は新規作成。 |
| a | 追加書き込み専用。読み込みはできない。<br>ファイルがない場合は新規作成。    |

# with文

- ▶ with open(場所, モード) as 変数:
- ▶ open()とclose()の処理をまとめたもの
- ▶ ブロックでファイルの開閉を管理できるため閉じ忘れが無い、基本的にこちらを使う

```
file = open("C:\\Users\\admin\\Desktop\\sample.txt", "r")  
print(file.read())  
file.close()
```



```
with open (r"C:\Users\admin\Desktop\sample.txt") as file:  
    print(file.read())
```

# 読み込み・書き込み

- ▶ 変数.read()

ファイルを全て読み込み、それを一つの文字列として扱う

- ▶ 変数.readline()

ファイルの1行を読み込み、それを文字列として扱う

- ▶ 変数.write("文字列")

文字列を引数に取り、ファイルに書き込む

# 練習問題①

▶ 何が表示されるか答えよ(※解答)

```
value=[1,2,3,4,5]
for x in range(len(value)):
    if x%2 == 0:
        value[x] += 3
    elif x%2 == 1:
        value[x] *= 2
    else:
        value[x] = 0
print(value)
```

※

```
value = [12,4,9,2,10,6]
j = sum(value)/len(value)
for x in range(len(value)):
    i = value[x]
    if i < j:
        value[x] += 2
    elif i > j:
        value[x] *= 2
    else:
        value[x] = 0
print(value)
```

※

# 練習問題②

▶ 何が表示されるか答えよ

```
def count(value):  
    count = 0  
    for x in value:  
        if x >= 5:  
            count += 2  
        if x >= 3:  
            count += 1  
        else:  
            break  
    return count
```

```
value=[]  
for x in range(10):  
    value.append(10 - x)
```

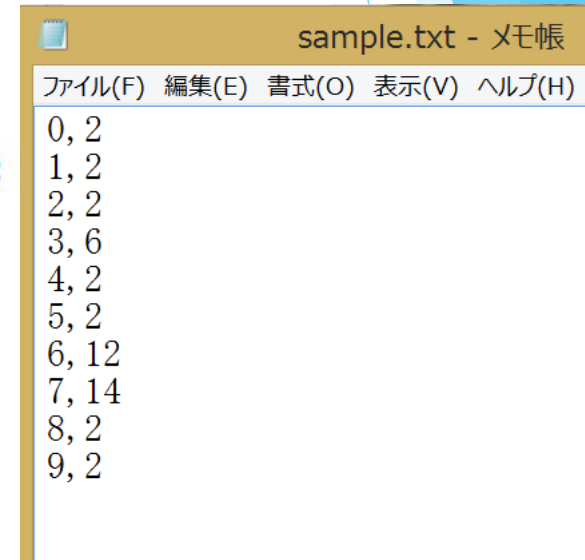
```
print(count(value))
```

※

# 練習問題③

▶ 何が表示されるか答えよ

```
count1 = 0
count2 = 0
with open (r"C:\Users\admin\Desktop\sample.txt","r") as file:
    for x in file:
        value = x.rstrip("\n").split(",")
        if int(value[0])%2 == 1:
            count1 += int(value[1])
        else:
            count2 += int(value[1])
print(count1)
print(count2)
```



sample.txt - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

0, 2  
1, 2  
2, 2  
3, 6  
4, 2  
5, 2  
6, 12  
7, 14  
8, 2  
9, 2

※

# 練習問題④

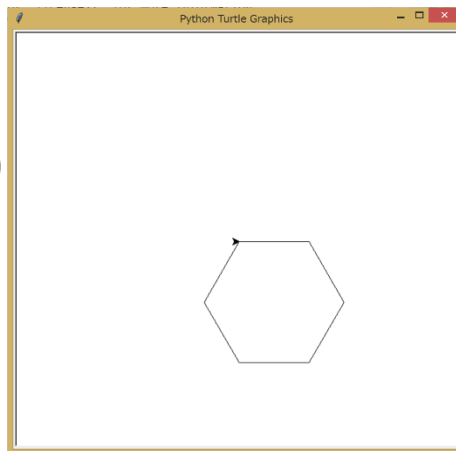
▶ 次のうち、どれか

```
import turtle
t = turtle.Pen()

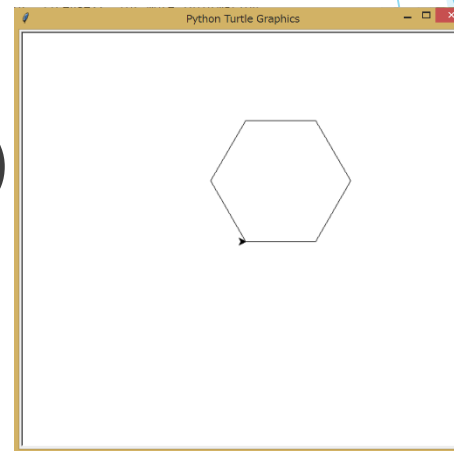
for x in range(6):
    t.left(60)
    t.forward(100)
```

※

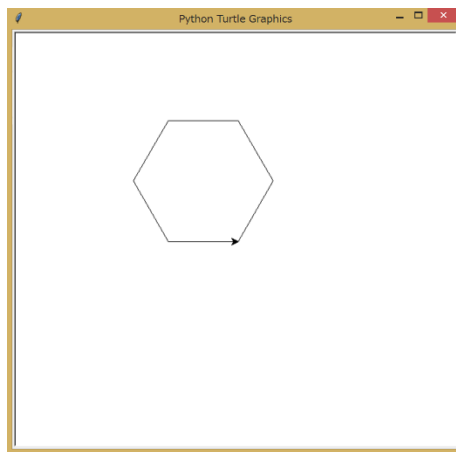
a)



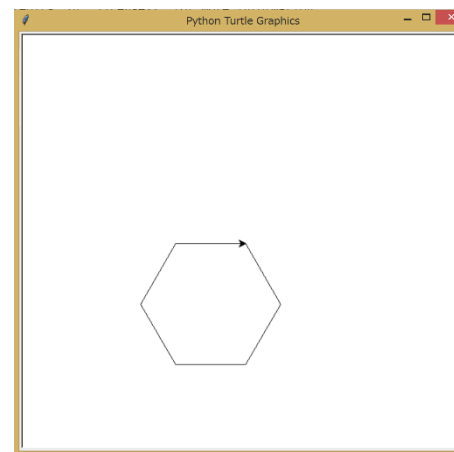
b)



c)



d)



# 練習問題⑤

## ▶ 描画されるものを示せ(図)

```
import turtle
t = turtle.Pen()

word = {"for":100, "left":90, "right":45}
move = ['for', 'right', 'left', 'for', 'left',
        'for', 'left', 'right', 'left', 'for',
        'left', 'for', 'right', 'for', 'left']

for x in move:
    if x == "left":
        t.left(word[x])
    elif x == "right":
        t.right(word[x])
    else:
        t.forward(word[x])
```

※解答は  
次のページ

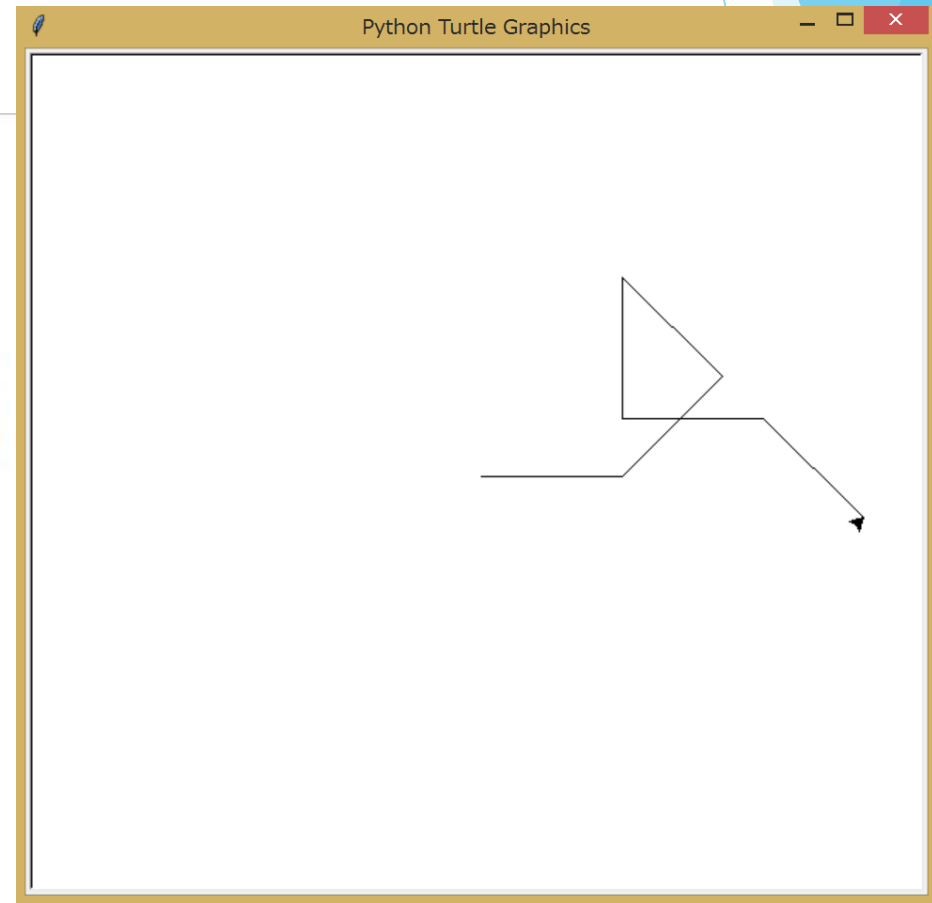
# 練習問題⑤解答

## ▶ 描画されるものを示せ(図)

```
import turtle
t = turtle.Pen()

word = {"for":100, "left":90, "right":45}
move = ['for', 'right', 'left', 'for', 'left',
        'for', 'left', 'right', 'left', 'for',
        'left', 'for', 'right', 'for', 'left']

for x in move:
    if x == "left":
        t.left(word[x])
    elif x == "right":
        t.right(word[x])
    else:
        t.forward(word[x])
```



# あとかき

- ▶ おそらくこのような難易度の問題はでない、  
もっと簡単で単純
- ▶ 問題を解く際は、与えられたプログラムコードを  
丁寧に読むこと
- ▶ turtleが最も正答率が低い(去年)  
→ left や right などの角度調整には注意
- ▶ この講座の演習問題はかなり複雑な部類のため、  
書けない、理解できない、は仕方ない