

Python講座

2016/5/19(木)

今回の内容

- ▶ スコープ①
- ▶ 例①
- ▶ 例②
- ▶ 例③
- ▶ 例④
- ▶ スコープ②
- ▶ グローバル変数①
- ▶ グローバル変数②
- ▶ グローバル変数③
- ▶ turtle 訂正版
- ▶ 演習問題

スコープ①

- ▶ Pythonをはじめとする多くのプログラミング言語には、**変数のスコープ**（有効範囲）という考え方がある
- ▶ 変数を宣言した場所によって、その変数を使える範囲が異なる
- ▶ 関数内で宣言された変数は**ローカル変数**と呼ばれる

例①

```
def f():  
    a=0
```

```
f()  
print(a)
```

▶ 関数fでaが宣言



▶ 関数外ではaが宣言されていない



```
print(a)  
NameError: name 'a' is not defined
```

例②

```
def f1(a):  
    print(a)  
def f2(b):  
    print(a + b)
```

```
f1(10)  
f2(20)
```

- ▶ 関数f1でローカル変数aが宣言
- ▶ 関数f2でローカル変数bが宣言

↓

- ▶ 関数f2で使用できる変数aがない

↓

```
print(a + b)  
NameError: name 'a' is not defined
```

例③

```
def f1(a):  
    print(a)  
def f2(a, b):  
    print(a + b)
```

```
a = 0  
f1(10)  
f2(20, 30)
```

- ▶ 関数f1でローカル変数aが宣言
- ▶ 関数f2でローカル変数aとbが宣言

↓

- ▶ 関数外で変数aが宣言

↓

```
>>>  
10  
50
```

例④

```
def f1(a):  
    print(a)  
def f2(b):  
    print(a + b)
```

```
a = 20  
f1(10)  
f2(30)
```

- ▶ 関数f1でローカル変数aが宣言
- ▶ 関数f2でローカル変数bが宣言

↓

- ▶ 関数外でaが宣言

↓

```
>>>  
10  
50
```

スコープ②

- ▶ 関数ごとに異なるスコープを持っている
- ▶ 関数の中で定義したローカル変数は、関数の外で利用することができない(例①)
- ▶ ある関数で定義されたローカル変数は別の関数で利用できない(例②)

グローバル変数①

```
def f1(a):  
    print(a)  
def f2(b):  
    print(a + b)
```

```
a = 20
```

```
f1(10)
```

```
f2(30)
```

- ▶ aは**グローバル変数**と呼ばれる
- ▶ 値を**参照する場合**は使用できる



- ▶ 今回はa+bの結果を求める



```
>>>
```

```
10
```

```
50
```

グローバル変数②

```
def f1(a):  
    print(a)  
def f2(b):  
    a = a + b  
    print(a)
```

```
a = 20  
f1(10)  
f2(30)
```

- ▶ a+bの値をaに代入する
- ▶ 値を参照する場合は使用できる



- ▶ aに新たな値を代入させることができない



```
a = a + b  
UnboundLocalError: local variable 'a'  
referenced before assignment
```

グローバル変数③

```
def f2(b):  
    global a  
    a = a + b  
    print(a)  
  
def f3():  
    print(a)
```

```
a = 20  
f2(30)  
f3()
```

- ▶ 変数aがglobalであることを宣言
- ▶ その関数内で値の代入ができるようになる



- ▶ aに新たな値を代入させることができる

```
>>>
```



```
50
```

```
50
```

turtle 訂正版

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

演習問題

- ▶ テーマ: turtleで穴埋めゲーム
- ▶ 今回は段階的にゲームを作る
 1. 正方形を描画し、中身を塗りつぶす
 2. 9×9マスの正方形の集合体を作る
 3. 行と列の値を入力するとその中身を塗りつぶす
 4. 塗りつぶせる回数を制限し、塗りつぶした場所によってポイントが手に入るようにする(運ゲー)

演習問題 補足

- ▶ import **turtle,random**
2つ目のimportを省略可
- ▶ random.**choice**(リスト名)
そのリストの**要素の一つ**を**ランダム**に返す
- ▶ リスト名.**append**(値)
そのリストに値を**要素として末尾に加える**
- ▶ リスト名.**remove**(値)
そのリストから値を**要素から削除**