



一般向け・プログラミング講座（3）

益田プログラミング研究会

Masuda Programming Research Society (MPRS)

<https://maro-v.jp/~nagata-iics/>

永田 武



#3 基本命令（座標・移動・回転・色）

(1) 本日の内容 cat 06_contents_#03

1. 座標

```
# grid.py
from turtle import *

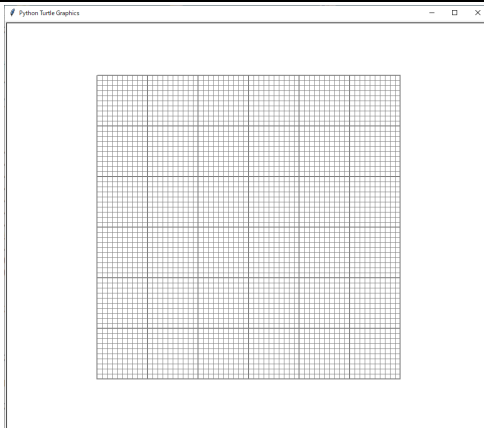
x0,y0,w,h = -300,300,600,600

t = Turtle()
t.ht()
t.speed('fastest')
t.pencolor('gray')
#tracer(20)
t.ht()

t.seth(-90)
for i in range(int(w/10)+1):
    x = x0 + 10*i
    if x%100 == 0: t.pensize(2)
    else: t.pensize(1)
    t.pu();t.goto(x,y0);t.pd()
    t.fd(h)

t.seth(0)
for i in range(int(h/10)+1):
    y = y0 - 10*i
    if y%100 == 0: t.pensize(2)
    else: t.pensize(1)
    t.pu();t.goto(x0,y);t.pd()
    t.fd(w)

exitonclick()
```



2. 移動

```
# 2.py
from turtle import * # コメント行 # turtle モジュールの読み込み
from Tools import Tools # Tools モジュールの読み込み

# 空白行
s = Screen() # スクリーンに s という名前を付ける
s.setup(600,600) # スクリーンのサイズを 600x600 とする

# 空白行
g = Tools() # Tools に g という名前を付ける
g.grid(-300,300,600,600) # グリッド (格子) を描く

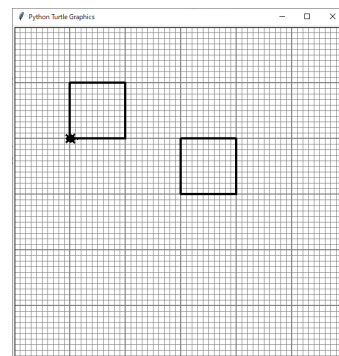
# 空白行
t = Turtle() # turtle に t という名前を付ける
t.shape('turtle') # turtle の形をカメラにする
#t.ht() # タートル t を隠す
t.pensize(4) # ペンの太さを 4 とする

t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度

# 空白行
t.pu();t.goto(-200,100);t.pd() # ペンアップ、(-200,100)に移動、ペンドアウン

t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度
t.fd(100);t.lt(90) # 100 ステップ前進、左 90 度

# 空白行
s.exitonclick() # スクリーン内でクリックする終了する
```



3. 回転

```
# 3.py
from turtle import * # コメント行 # turtle モジュールの読み込み
from Tools import Tools # Tools モジュールの読み込み

# 空白行
s = Screen() # スクリーンに s という名前を付ける
s.setup(600,600) # スクリーンのサイズを 600x600 とする
t = Turtle() # スクリーンのサイズを 600x600 とする
t.shape('turtle') # turtle の形をカメラにする
#g = Tools() # Tools に g という名前を付ける
#g.grid(-300,300,600,600) # グリッド (格子) を表示

# 空白行
t.home() # ホームに移動 ((0,0) 右向き)
t.seth(90) # 方向を 90 度とする
t.pensize(2) # ペンサイズを 2 とする

# 空白行
t.pu();t.goto(-100,100);t.pd() # タートルの移動
t.pensize(4) # ペンサイズを 4 とする
k = 10 # 円の個数
for i in range(k): # k 回のループ
    t.circle(50) # 半径 50 の円の描画
    t.lt(int(360/k)) # 左 360/k 度

# 空白行
s.exitonclick() # スクリーン内でクリックする終了する
```

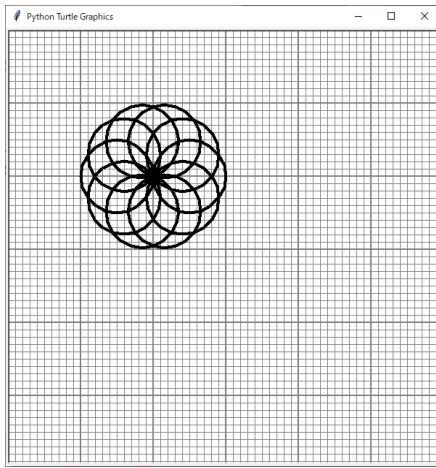
```
# 1.py
from turtle import * # コメント行 # turtle モジュールの読み込み
from Tools import Tools # Tools モジュールの読み込み

# 空白行
s = Screen() # スクリーンに s という名前を付ける
s.setup(600,600) # スクリーンのサイズを 600x600 とする

# 空白行
t = Turtle() # turtle に t という名前を付ける
t.shape('turtle') # turtle の形をカメラにする
#t.shape('circle') # turtle の形を円 (●) にする
#t.shape('square') # turtle の形を四角 (■) にする
#t.shapesize(2,2,2) # タートルのサイズを 2 倍に変更

# 空白行
tracer(0) # 描画速度を速くする
g = Tools() # Tools g という名前を付ける
g.grid(-300,300,600,600) # スクリーンに座標を描く

s.exitonclick() # スクリーン内でクリックする終了する
```



4. 色

https://www.rapidtables.com/web/color/RGB_Color.html

```
# 4.py                                # コメント行
from turtle import *                  # turtle モジュールの読み込み
from Tools import Tools               # Tools モジュールの読み込み

# 空白行
s = Screen()                          # スクリーンに s という名前を付ける
s.setup(600,600)                      # スクリーンのサイズを 600×600 とする
t = Turtle()                          # turtle に t という名前を付ける
t.pensize(4)                          # ペンサイズを 2 とする

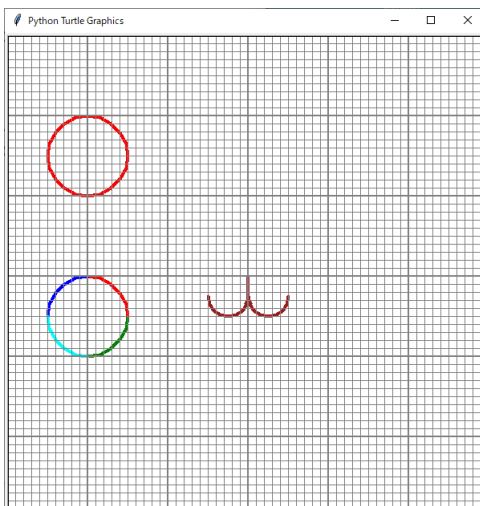
# 空白行
t.pu(); t.goto(-200,100); t.pd()      # タートルの移動
t.color('red','red')                  # ペンの色を赤に指定 (ペン色、塗りつぶし色)
"""
circle(radius, extent=None)
radius -- 半径
extent -- 角度 (または None)
"""
t.circle(50)
t.pu(); t.goto(-200,-100); t.pd()     # タートルの移動
t.color('green','green')
t.circle(50,90)

t.color('red','red'); t.circle(50,90)
t.color('blue','blue'); t.circle(50,90)
t.color('cyan','cyan'); t.circle(50,90)

s.colormode(255)
t.color((155,28,28)) # r,b,b
t.pu(); t.goto(0,-100); t.pd()
t.pu(); t.home(); t.pd()
t.rt(90); t.fd(25)
t.circle(25,180)
t.pu(); t.goto(-50,-25); t.pd()
t.rt(180); t.circle(25,180)

# 空白行
tracer(0)                            # 描画速度を速くする
g = Tools()                          # Tools に g という名前を付ける
g.grid(-300,300,600,600)              # グリッド (格子) を描く

# 空白行
s.exitonclick()                      # スクリーン内でクリックする終了する
```



4. おまけ

(1) 座標による猫の顔

```
# neko.py

from turtle import *
from Tools import Tools

p =
[(0,0),(50,10),(70,20),(75,35),(70,50),(50,90),(60,105),(
60,120),(50,120),(40,115),(20,100),(0,100),
(-20,100),(-40,115),(-50,120),(-60,120),(-60,105),(-50,90),
(-70,50),(-75,35),(-70,20),(-50,10),(0,0) ]

t = Turtle()
t.ht()
t.pensize(4)
for i in range(len(p)):
    t.goto(p[i])

t.pu(); t.goto(0,30); t.pd(); t.circle(7)
t.rt(90)
t.fd(5)
t.circle(7,180)

t.pu(); t.goto(-10-5,30-5); t.pd();
t.rt(180)
t.circle(7,180)

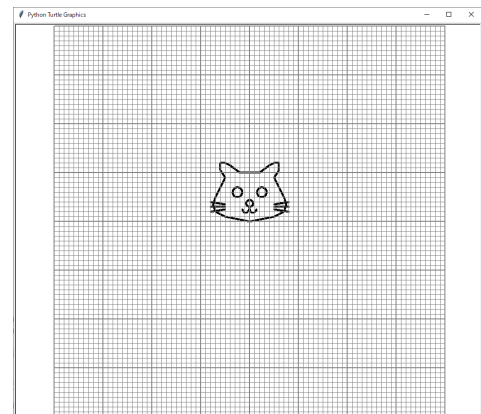
t.pu(); t.home()
#t.pu(); t.goto(0,0); t.pd();
t.pu(); t.goto(-25,50); t.pd();
t.circle(10)
t.pu(); t.goto( 25,50); t.pd();
t.circle(10)

t.pu(); t.goto(50,35); t.pd(); t.goto(80,40)
t.pu(); t.goto(50,30); t.pd(); t.goto(80,30)
t.pu(); t.goto(50,25); t.pd(); t.goto(80,20)

t.pu(); t.goto(-50,35); t.pd(); t.goto(-80,40)
t.pu(); t.goto(-50,30); t.pd(); t.goto(-80,30)
t.pu(); t.goto(-50,25); t.pd(); t.goto(-80,20)

tracer(2)
g = Tools()
g.grid(-400,400,800,800)

exitonclick()
```



参考資料

(1) Python によるアルゴリズムとデータ構造の基礎
<https://www.coronasha.co.jp/np/isbn/9784339029079/>

(2) Raspberry Pi による IoT システム開発実習
<https://www.morikita.co.jp/books/mid/085571>

アンケート

<http://192.168.xxx.yyyy/q.php> (url.py)

xxx.yyy は指示します

おつかれさまでした

連絡先 t.nagata.wp@gmail.com 永田 武
<http://www.maro-v.jp/~nagata-iics/>