

2024 永田 武

t.nagata.wp@gmail.com

#2 基本的なアルゴリズム (連接・反復・判断)

(1)本日の内容 cat 05_contents_#02

1. Turtle プログラミング

(1)../04/t1.py をコピーする cp ../04/t1.py . (cp コマンド) ls

cat t1.py

t1.py
from turtle import *
t = Turtle()
t.shape('turtle')
t.fd(100) # forward(100)
t.rt(90) # right(90)
t.fd(100)
t.rt(90)
t.fd(100)

t.rt(90) t.fd(100)

t.rt(90)
exitonclick()

動作確認 python t1.py

(2) 反復 (ループ、繰り返し)

t.fd(100)

t.rt(90)

の部分を4回繰り返すと同じ結果となる

for i in range(4): t.fd(100) t.rt(90)

・t1.py をコピーして t2.py を作る

cp t1.py t2.py (cp コマンド) **gedit t2.py &** (エディタでプログラムを作る)

t2.py
from turtle import *
t = Turtle()
t.shape('turtle')
for i in range(4):
 t.fd(100) # forward(100)
 t.rt(90) # right(90)
exitonclick()

動作確認 python t2.py

(3) 三角形を描く アルゴリズム

fd(100) for 文を使うと、コードが短くなる rt(120) for in range(3): fd(100) rt(120) rt(120) fd(100)

t2.py をコピーして t3.py を作る

cp t2.py t3.py (cp コマンド)

gedit t3.py & (エディタでt3.pyを作る)

t3.py
from turtle import *

t = Turtle()
t.shape('turtle')
for i in range(3):
 t.fd(100)
 t.rt(120)
exitonclick()

動作確認 python t3.py

(4) 五角形を描く

t.rt(360/5)として5回ループ

★三角形の外角に和は360度

(5) 六角形を描く

t3.py に追記する

t.rt(360/6)として6回ループ

(6) n 角形を描く

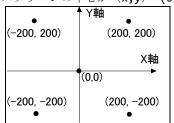
t3.py に追記する

t.rt(360/n)としてn回ループ、nは予めセット

n = 8 #8角形 for in range(n): t.fd(100) t.rt(360/n)

(7) Turtle の座標系

スクリーンの中心が (x,y) =(0,0)



(8) while ループ

while True: とすると無限ループ タートルを 1 歩ずつ無限に動かす

gedit t4.py &

t4.py
from turtle import *
t = Turtle()
t.shape('turtle')
while True:
 t.fd(1)
exitonclick()

動作確認 python t4.py

(9) 判断

t.fd(1)を繰り返すと、タートルが画面内から出て行く x 座標が(200)になったら、引き返す(後ろを向く) 後ろを向くには、**t.rt(180)**

タートルのx座標を知るには、t.xcor()

t4.py を改造

t4.py
from turtle import *
t = Turtle()
t.shape('turtle')
while True:

```
t.fd(1)
  if t.xcor() >= 200:
    t.rt(180)
exitonclick()
```

動作確認 python t4.py

```
更に、x 座標が(-200)になったら、引き返す(後ろを向く)
  if t.xcor() <= -200:
    t.rt(180)
```

複合条件 or(または)を用いると if t.xcor() <= -200 or t.xcor() >= 200: 更に、簡単に

```
x = t.xcor()
if x <= -200 \text{ or } x >= 200:
```

(10) ループからの脱出

break を用いる

```
count = 0
while True:
 t.fd(1)
 x = t.xcor()
 if x < -200 \text{ or } x >= 200:
  t.rt(180)
  if count >=5:
  break
```

```
参考) """で囲むと、コメントとなる
```

```
# t4.py
from turtle import *
t = Turtle()
t.shape('turtle')
count = 0
while True:
 t.fd(1)
 if t.xcor() >= 200: t.rt(180)
 if t.xcor() <= -200: t.rt(180)
 if t.xcor() >= 200 or t.xcor() <= -200:
   t.rt(180)
 ....
 x = t.xcor()
 if x < -200 \text{ or } x >= 200:
   t.rt(180)
   count = count + 1
 if count >= 3: break
exitonclick()
```

(11) プログラムの例

・整数の和(1+2+3+4+5+6+7+8+9+10=55)

```
# intsum for.py
n = 10
sum = 0
for i in range(1,n+1):
 print(i)
 sum = sum + i
print('sum = ',sum)
```

前判定

```
# intsum_while.py
n = 10
```

```
i = 0
sum = 0
while i < n:
 i = i + 1
  print(i)
  sum = sum + i
print('sum = ',sum)
```

· 後判定

```
# intsum dowhile.py
n = 10
i = 0
sum = 0
while True:
 i = i + 1
  print(i)
  sum = sum + i
  if i == n: break
print('sum = ',sum)
```

(12) Turtle のメソッド例

```
• pencolor('red')
                   (ペンの色)
• fillcolor('cyan')
                   (塗りつぶしの色)
• begin_fill()
                   (塗りつぶし開始)
• end_fill()
                   (塗りつぶし終了)
goto(x,y)
                  (移動)
                  (pen up)
· pu()
                  (pen down)
· pd()
                  (ペンサイズ)
pensize(5)
```

https://docs.python.org/ja/3/library/turtle.htm Turtle のメソッド

2. まとめ

- ・プログラムは、「連接」、「反復」、「判断」の3つで 構成できる
- ・ 反復には、「for 文」と「while 文」がある
- ・反復の脱出は、break 文
- ・判断には、「if 文」が用いられる
- ・関係演算子 ==,<,>,!=,>=,<=
- · 論理演算子 and, or, not

アンケート

http://192.168.xxx.yyy/q.php xxx. yyy は指示します おつかれさまでした

次回の内容: 乱数・ファイル入出力・日時・外部プログ ラムの起動・定周期処理

日時:

参考書籍

(1)永田:「Python によるアルゴリズムとデータ構造の基礎」 (コロナ社)

https://www.coronasha.co.jp/np/isbn/9784339029079/ (2) 永田:「Raspberry Pi による IoT システム開発実習」(森

https://www.morikita.co.jp/books/mid/085571 連絡先 t.nagata.wp@gmail.com 永田 武 http://www.maro-v.jp/~nagata-iics/