



一般向け・プログラミング講座（８）

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



８．アプリ開発（時計）

(1) 本日の内容 cat 12_contents_#09

1. 時計の文字盤の表示

(1) mode("logo")

時計回りの角度を使用するために mode("logo") を用いる。タートルを上向きとし、時計回りとする。

(2) cface.py (clockface)

文字盤は、1 秒刻みで一周（360 度）に 60 個の印を描画する。すなわち、 $360/60=6$ 度毎に描画する。ただし、5 秒毎に太めの短い直線を描く。

g cface.py &

```
# cface.py (clockface)
from turtle import *

mode("logo") # 上向き・角度を時計回りとする
def goto(distance, angle):
    pu();rt(angle);fd(distance);lt(angle);pd()
def clockface(radius): # radius:半径
    reset()
    pensize(7)
    for i in range(60):
        goto(radius,0)
        if i % 5 == 0: # 5 秒毎の場合
            fd(25)
            goto(-radius-25,0)
        else:
            dot(3)
            goto(-radius,0)
        rt(6)
    #tracer(False) # アニメーション停止、高速
    clockface(160)
    exitonclick()
```

動作確認 p cface.py



2. 秒針の表示(cs.py: clock second)

・秒針を hand 関数で作成し、make_hand_shape 関数で、タートルを秒針の形とし、"second_hand" という名前で登録する。

・そして、その second_hand のタートルの向きを $6(=360/60)$ 度毎に変える (setheading(6*sec) メソッド) ことによって、秒針の動きを実現する。

・時間に同期させて動かすために、tick 関数を作り、その中でタートルのメソッド ontimer(tick, 1000) を記述して 1 秒毎の処理を実現する（再帰呼び出し）。

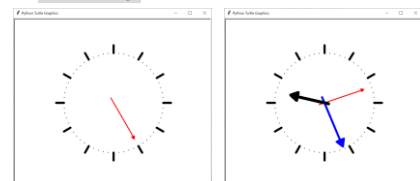
g cs.py &

```
# cs.py (clock second)
from turtle import *
from datetime import datetime
```

```
mode("logo") # 上向き・角度を時計回りとする
def goto(distance, angle):
    pu();rt(angle);fd(distance);lt(angle);pd()
def hand(length, tip): # tip:先端部分
    fd(length*1.15)
    rt(90);fd(tip/2.0)
    lt(120);fd(tip)
    lt(120);fd(tip)
    lt(120);fd(tip/2.0)
def make_hand_shape(name, length, tip):
    reset()
    goto(-length*0.15,0)
    begin_poly()
    hand(length, tip)
    end_poly()
    hand_form = get_poly()
    register_shape(name, hand_form)
def clockface(radius):
    reset()
    pensize(7)
    for i in range(60):
        goto(radius,0)
        if i % 5 == 0:
            fd(25)
            goto(-radius-25,0)
        else:
            dot(3)
            goto(-radius,0)
        rt(6)
def setup():
    global second_hand
    make_hand_shape("second_hand", 125, 10)
    clockface(160)
    second_hand = Turtle()
    second_hand.shape("second_hand")
    second_hand.color("red", "red")
    second_hand.resizemode("user")
    second_hand.shapesize(1, 1, 3)
    second_hand.speed(0)
    ht()
def tick():
    t = datetime.today()
    sec = t.second
    second_hand.setheading(6*sec)
    ontimer(tick, 1000) # 1 sec

tracer(False)
setup()
tracer(True)
tick()
exitonclick()
```

動作確認 p cs.py



3. 分針の追加(cm.py: clock minute)

・秒針の表示と同様にして、分針の動きを実装する。

・タートルの形を "minute_hand" とし、minute=t.minute+sec/60.0 を計算し、1 秒毎に向きを 6*minute 度変えて実現する。

g cm &

```
def goto ...
def hand ...
def make_hand_shape ...

def setup():
    global second_hand, minute_hand
    make_hand_shape("second_hand", 125, 10)
    make_hand_shape("minute_hand", 130, 25)
```

```

clockface(160)
second_hand = Turtle()
...
minute_hand = Turtle()
minute_hand.shape("minute_hand")
minute_hand.color("blue", "blue")
minute_hand.resizemode("user")
minute_hand.shapesize(1,1,7)
minute_hand.speed(0)
def tick():
    t = datetime.today()
    sec = t.second
    second_hand.setheading(6*sec)
    minute = t.minute + sec/60.0
    minute_hand.setheading(6*minute)
    tracer(True)
    ontimer(tick, 1000) # 1 sec
...

```

動作確認 **p cm.py**

4. 時針の追加(ch.py: clock hour)

- ・分針の表示と同様にして、時針の動きを実装する。
- ・タートルの形を "hour_hand" とし、**hour=t.hour+minute/60.0** を計算し、**1 秒毎に向きを 30*hour 度変えて実現**する(12 時間で 360 度)。

g ch &

```

def goto ...
def hand ...
def make_hand_shape ...

def setup():
    global second_hand, minute_hand, hour_hand
    make_hand_shape("second_hand", 125, 10)
    make_hand_shape("minute_hand", 130, 25)
    make_hand_shape("hour_hand", 90, 25)
    clockface(160)
    second_hand = Turtle()
    ...
    minute_hand = Turtle()
    ...
    hour_hand = Turtle()
    hour_hand.shape("hour_hand")
    hour_hand.color("black", "black")
    for hand in second_hand, minute_hand, hour_hand:
        hand.resizemode("user")
        if(hand == hour_hand):
            hand.shapesize(1,1,10)
        elif(hand == minute_hand):
            hand.shapesize(1,1,7)
        else: hand.shapesize(1, 1, 3)
        hand.speed(0)
def tick():
    t = datetime.today()
    sec = t.second
    second_hand.setheading(6*sec)
    minute = t.minute + sec/60.0
    hour = t.hour + minute/60.0
    tracer(True)
    ontimer(tick, 1000) # 1 sec
...

```

動作確認 **p ch.py**

5. 時計の完成(clock.py)

- ・日時のデータを表示させて、時計を完成させる。

g clock.py &

```

# clock.py
from turtle import *
from datetime import datetime

mode("logo") # 上向き・角度を時計回りとする

def goto(distance, angle):

```

```

...
def hand(length, tip):
...
def make_hand_shape(name, length, tip):
...
def clockface(radius):
...
def setup():
    global second_hand, minute_hand, hour_hand, writer
    ...
    writer = Turtle()
    writer.ht()
    writer.pu()
    def day_of_week(t): # day of week
        week_name = ["Monday", "Tuesday", "Wednesday",
                     "Thursday", "Friday", "Saturday", "Sunday"]
        return week_name[t.weekday()]
    def month_name(today): # month name
        mname = ["Jan.", "Feb.", "Mar.", "Apr.", "May", "June",
                 "July", "Aug.", "Sep.", "Oct.", "Nov.", "Dec."]
        y = today.year
        m = mname[today.month - 1]
        d = today.day
        return "%s %d %d" % (m, d, y)

def tick():
    ...
    tracer(False)
    writer.clear()
    writer.home()
    writer.fd(65)
    writer.write(day_of_week(t),
                 align="center", font=("Courier", 14, "bold"))
    #writer.back(150) # 65-150=-85
    writer.home()
    writer.bk(85)
    writer.write(month_name(t),
                 align="center", font=("Courier", 14, "bold"))
    tracer(True)
    try:
        second_hand.setheading(6*sec)
        minute_hand.setheading(6*minute)
        hour_hand.setheading(30*hour)
    except:
        writer.clear()
        raise
    tracer(True)
    ontimer(tick, 100)
...

```

動作確認 **p clock.py**



6. まとめ

- ・タートルを時計の針に形を変えて実装し、そのタートル（オブジェクト）を回転させて、時計の針を実現するという発想は、面白いと思いませんか。
- ・このような柔軟な考え方ができるのが、プログラミングの醍醐味だと思いますが、いかがでしたか。

アンケート (url.py)

<http://192.168.xxx.yyy/q.php>

xxx.yyy は指示します

連絡先

t.nagata.wp@gmail.com 永田 武

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