



Python으로 배우는 소프트웨어 원리

Appendix 06. 토끼, 거북이, 달팽이 경주 Graphic
-Turtle모듈 활용-

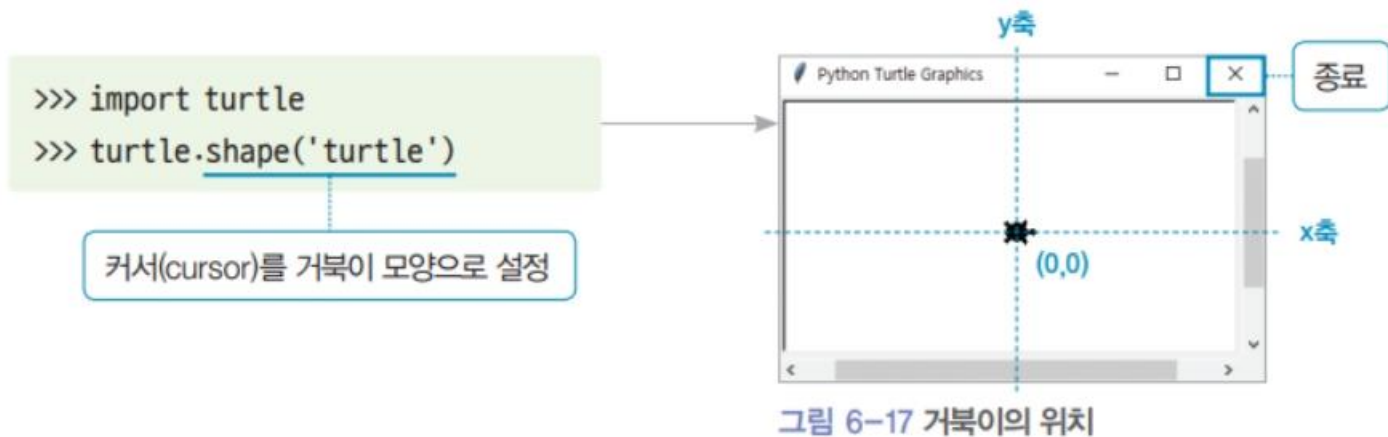
01. 터틀 그래픽스 모듈의 활용

I. 터틀 그래픽스 모듈

- 파이썬을 설치할 때 기본으로 제공되는 터틀 그래픽스라는 모듈을 이용하면 캔버스에 원하는 모양을 그리는 그래픽 프로그램을 간단하게 작성 가능

II. 터틀의 기본 사용법

- 좌표를 이용한 위치 이동뿐 아니라 픽셀pixel 단위로 거리를 지정하여 거북이를 움직일 수도 있음



02. Turtle Graphics Method (1)

[Method 활용]

표 6-3 그리기 동작과 회전 관련 메소드

메소드명	동작	예시	인수 설명
forward(), fd()	앞으로 이동	<code>turtle.forward(100)</code> <code>turtle.fd(100)</code>	픽셀 단위의 거리
backward(), back()	뒤로 이동	<code>turtle.back(100)</code>	픽셀 단위의 거리
left(), lt()	왼쪽으로 회전	<code>turtle.left(90)</code>	회전 각도
right(), rt()	오른쪽으로 회전	<code>turtle.rt(90)</code>	회전 각도
circle()	원 그리기	<code>turtle.circle(50)</code> <code>turtle.circle(50, 180)</code>	반지름 반지름과 각도
speed()	그리기 속도 설정	<code>turtle.speed(10)</code>	0에서 10 사이의 수

02. Turtle Graphics Method (2)

[Method 활용]

표 6-4 펜과 색상 설정 관련 메소드

메소드명	동작	예시	인수 설명
pendown(), pd(), down()	펜을 내려 그리기 준비	<code>turtle.pd()</code>	—
penup(), pu(), up()	펜을 올려 그리지 않기	<code>turtle.pu()</code>	—
shape()	펜 모양 설정	<code>turtle.shape('turtle')</code>	'arrow', 'circle', 'classic', 'square', 'triangle', 'turtle',
pensize(), width()	펜 두께 설정	<code>turtle.pensize(5)</code>	정수, 값이 클수록 두꺼워짐
pencolor()	펜 색상 설정	<code>turtle.pencolor('red')</code>	색상 이름
color()	펜 색상과 채우기 색상 설정	<code>turtle.color('white', 'yellow')</code>	(펜 색, 채우기 색) 순서로 색상 이름 입력
begin_fill() ~ end_fill()	도형의 색 채우기	<code>turtle.begin_fill()</code> <code>turtle.circle(100)</code> <code>#원 그리기</code> <code>turtle.end_fill()</code>	—

02. Turtle Graphics Method (3)

[Method 활용]

표 6-5 캔버스와 윈도우 설정 관련 메소드

메소드명	동작	예시	인수 설명
setup()	캔버스 크기 설정	<code>turtle.setup(800, 600)</code>	폭(가로)과 높이(세로)
title()	터틀 윈도우의 제목 설정	<code>turtle.title('Painter')</code>	-
write()	캔버스에 출력하기	<code>turtle.write("ABCDEFGG")</code>	문자나 숫자 등
clear()	그림 모두 지우기	<code>turtle.clear()</code>	-
reset()	그림을 지우고, 위치와 색상 설정 초기화	<code>turtle.reset()</code>	-
done()	터틀 윈도우를 닫지 않고 그리기 종료	<code>turtle.done()</code>	-
exitonclick()	마우스를 클릭하면 터틀 윈도우 닫기	<code>turtle.exitonclick()</code>	-
bye()	터틀 윈도우 닫기	<code>turtle.bye()</code>	

03. 실습 : Turtle Module 활용

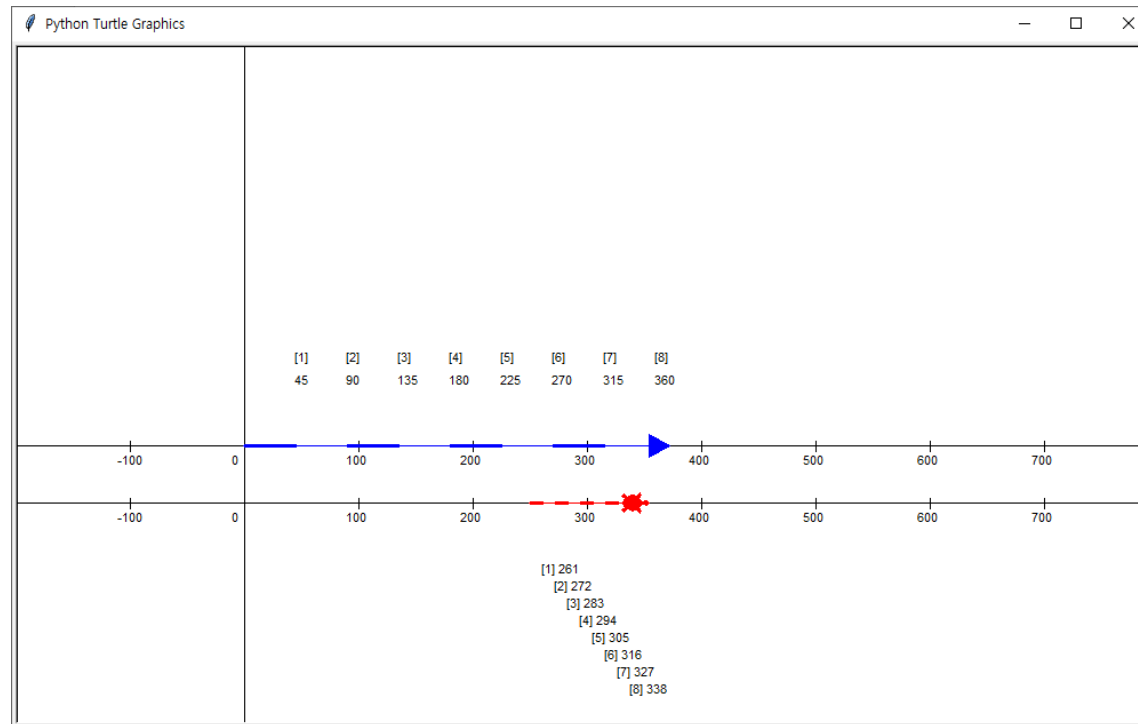
[실습] Turtle Module로 구현: 토끼와 거북이 경주

- ◆ 토끼와 거북이가 달리기 시합을 한다.

토끼는 1분에 45m를 달리고, 거북이는 1분에 11m를 달린다.

- 거북이와 토끼의 최초 위치 값을 키보드로 입력 받는다.
- 매 분별로 토끼와 거북이의 위치를 표시(경과 분수 포함)

❖ [Ch06-xyShift좌표.py 코드 참고](#)



03. Turtle Graphics Module 활용 예제

[실습] 축 이동된 xy 좌표 그리기 (1)

Ch06-xyShift좌표.py

```
import turtle as t
w_ysize = 1600 #window 가로 크기 pixel
w_hsize = 900 #window 세로 크기 pixel
win = t.getscreen() #Window screen 객체 생성
win.setup(w_ysize, w_hsize)
```

```
l_shift = 400 #x축 left로 shift pixel 값
d_shift = 100 #y축 down으로 shift pixel 값
```

##Shift된 new x-좌표 구하기

```
def newx(x=0):
    global l_shift
    return (x - l_shift)
```

##Shift된 new y-좌표 구하기

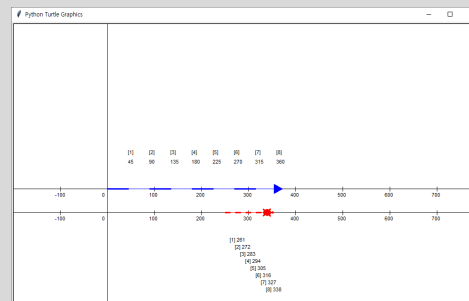
```
def newy(y=0):
    global d_shift
    return (y - d_shift)
```

##[함수] (x1, y1) - (x2, y2) 직선 그리기

```
def line(x1, y1, x2, y2):
    t.up()
    t.goto(x1, y1)
    t.down()
    t.goto(x2, y2)
```

##[함수] (x,y)에 텍스트 쓰기

```
def txtwrite(x, y, text):
    t.up()
    t.goto(x, y)
    t.down()
    t.write(text)
```



03. Turtle Graphics Module 활용 예제

[실습] 축 이동된 xy 좌표 그리기 (2-1)

Ch06-xyShift좌표.py

```
##[함수] x, y축 좌표 그리기
def draw_xy(wsize, step) :
    line(-wsize, 0, wsize, 0)    # x축 라인
    line(0, -wsize, 0, wsize)    # y축 라인

    for i in range(-wsize, wsize, step) :
        line(i, -5, i, 5)        # x축 눈금
        if i != 0:
            txtwrite(i-10, -20, i)
    for i in range(-wsize, wsize, step) :
        line(-5, i, 5, i)        # y축 눈금
        if i != 0:
            txtwrite(10, i-5, i)
```

```
##.....메인 시작.....##
##변수 초기화
wsize = 500    #x축 길이
step = 100     #눈금 간격 pixel

## x, y축 좌표 그리기
t.hideturtle()
t.speed(0)

draw_xy(wsize, step)

t.exitonclick() #실행 창을 닫지 않도록
##.....메인 끝.....##
```


03. Turtle Graphics Module 활용 예제

[실습] 축 이동된 xy 좌표 그리기 (2-2)

Ch06-xyShift좌표.py

➤ x축 라인을 그릴 때 축 이동된 y(newy(0))에 그림

```
##[함수] x, y축 좌표 그리기(축 이동된)
def draw_xy(wsize, hsize, step):
    line(-wsize, newy(0), wsize, newy(0)) # x축 라인
    line(newx(0), -hsize, newx(0), hsize) # y축 라인

    for i in range(0, -wsize, -step):
        line(i, newy(-5), i, newy(5))      #x(-)축 눈금
        txtwrite(i-10, newy(0)-20, i-newx(0))

    for i in range(step, wsize, step):
        line(i, newy(-5), i, newy(5))      #x(+)축 눈금
        txtwrite(i-10, newy(0)-20, i-newx(0))

    for i in range(0, -hsize, -step):
        line(newx(-5), i, newx(5), i)      #y(-)축 눈금
        txtwrite(newx(0)+10, i-10, i-newy(0))

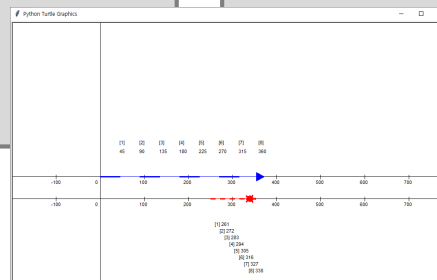
    for i in range(step, hsize, step):
        line(newx(-5), i, newx(5), i)      #y(+)축 눈금
        txtwrite(newx(0)+10, i-10, i-newy(0))
```

```
##.....메인 시작.....##
##변수 초기화
wsize = int(w_wsize / 2) #x(+)축 길이
hsize = int(w_hsize / 2) #y(+)축 길이
step = 100 #눈금 간격 pixel

## x, y축 좌표 그리기
t.hideturtle()
t.speed(0)

draw_xy(wsize, hsize, step)

t.exitonclick() #실행 창을 닫지 않도록
##.....메인 끝.....##
```



03. 실습 : Turtle Module 활용

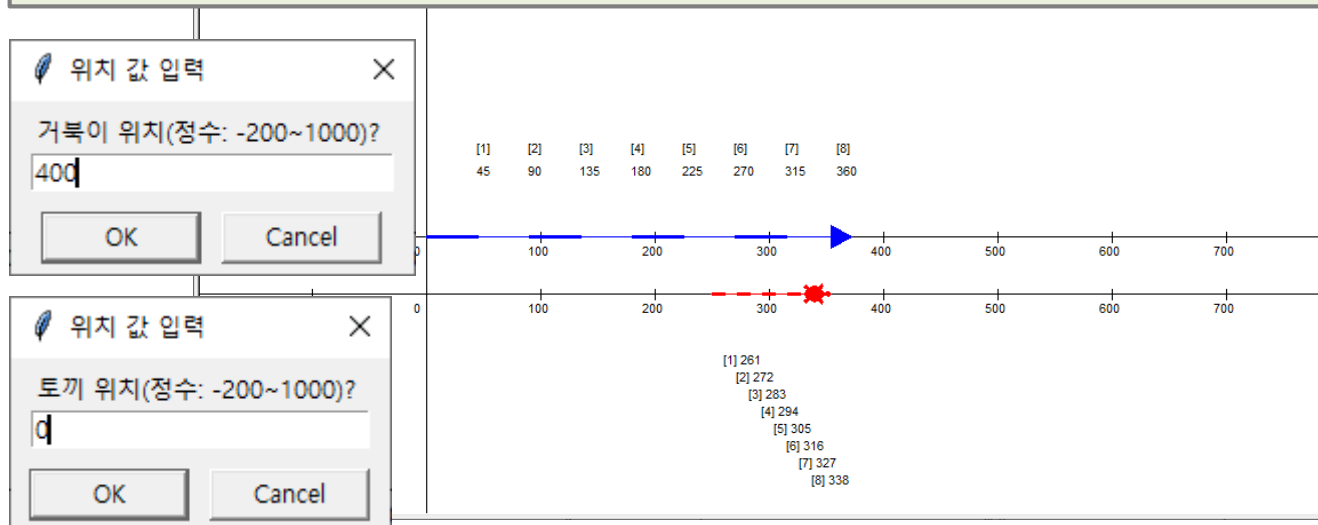
[실습] Turtle Module로 구현: 토끼와 거북이 경주

- ◆ 토끼는 1분에 45m를 달리고, 거북이는 1분에 11m를 달린다.
 - 거북이와 토끼의 최초 위치 값을 키보드로 입력 받는다.
 - 매 분별로 토끼와 거북이의 위치를 표시(경과 분수 포함)
 - 토끼와 거북이를 독자적 행동이 가능한 객체를 생성하여 해결
- ❖ Ch06-xyShift좌표.py 코드 참고

##다이얼로그 윈도우로 값 입력

```
p_ttl = int(t.numinput("위치 값 입력", "거북이 위치(정수: -200~1000)? "))    #turtle 최초 위치
```

```
p_rbt = int(t.numinput("위치 값 입력", "토끼 위치(정수: -200~1000)? "))    #rabbit 최초 위치
```



03. Turtle Graphics Module 활용 예제

[실습] 토끼와 거북이 경주 (1)

Ch06-xy거북이.py

```
##.....메인 시작.....##  
##변수 초기화  
l_shift = 400  #x축 left로 shift pixel 값  
d_shift = 100  #y축 down으로 shift pixel 값  
  
wsize = int(w_ysize / 2)  #x(+)축 길이  
hsize = int(w_hsize / 2)  #y(+)축 길이  
step = 100  
  
## x, y축 좌표 그리기  
t.hideturtle()  
t.speed(0)  
draw_xy(wsize, hsize, step)
```

03. Turtle Graphics Module 활용 예제

[실습] 토끼와 거북이 경주 (2)

Ch06-xy거북이.py

```
##토끼, 거북이 객체 생성하기
ttl = t.Turtle() #turtle 객체 생성
ttl.hideturtle()
ttl.penup()
ttl.shape('turtle')
ttl.color('red')

rbt = t.Turtle() #rabbit 객체 생성
rbt.hideturtle()
rbt.penup()
rbt.shape('triangle')
rbt.color('blue')

##게임 시작
p_ttl = int(t.numinput("위치 값 입력", "거북이 위치(정수: -200~1000)? ")) #turtle 최초 위치
p_rbt = int(t.numinput("위치 값 입력", "토끼 위치(정수: -200~1000)? ")) #rabbit 최초 위치

run_game(ttl, rbt, p_ttl, p_rbt)

t.exitonclick() # 실행 창을 닫지 않도록
##.....메인 끝.....##
```

❖ [turtle.numinput\(title, prompt, default=None, minval=None, maxval=None\)](#)

03. Turtle Graphics Module 활용 예제

[실습] 토끼와 거북이 경주 (3)

Ch06-xy거북이.py

```
##[함수] turtle-rabbit running game
def run_game(ttl, rbt, p_ttl, p_rbt) :
    #turtle 상태 설정
    ttl.speed('fastest')
    ttl.goto(newx(p_ttl), newy(0))
    ttl.showturtle()
    ttl.pendown()
    ttl.speed('slowest')

    #rabbit 상태 설정
    rbt.speed('fastest')
    rbt.goto(newx(p_rbt), newy(50))
    rbt.pendown()
    rbt.showturtle()
    rbt.speed('slowest')
```

03. Turtle Graphics Module 활용 예제

[도전-1] 토끼와 거북이 경주 (완성)

Ch06-xy거북이.py

```
min = 0

while p_rbt <= p_ttl :
    min += 1
    p_rbt += 45
    p_ttl += 11

    #두 객체 이동 처리
    rbt.pensize(          ) #펜사이즈 번갈아 굵기 조절
    ttl.pensize(          ) #펜사이즈 번갈아 굵기 조절

    rbt.forward(45)
    ttl.forward(11)

    #위치 표시하기
    txtwrite(newx(p_rbt), newy(120), '['+str(min)+']') #rabbit 경과 시간(min)
    txtwrite(newx(p_rbt), newy(100), p_rbt)           #rabbit 위치(m)
    txtwrite(newx(p_ttl), newy(-50-(15*min)), '['+str(min)+' '+str(p_ttl)) #turtle 위치(m)
```

03. 실습 : Turtle Module 활용

[실습] 달팽이 우물에서 탈출하기

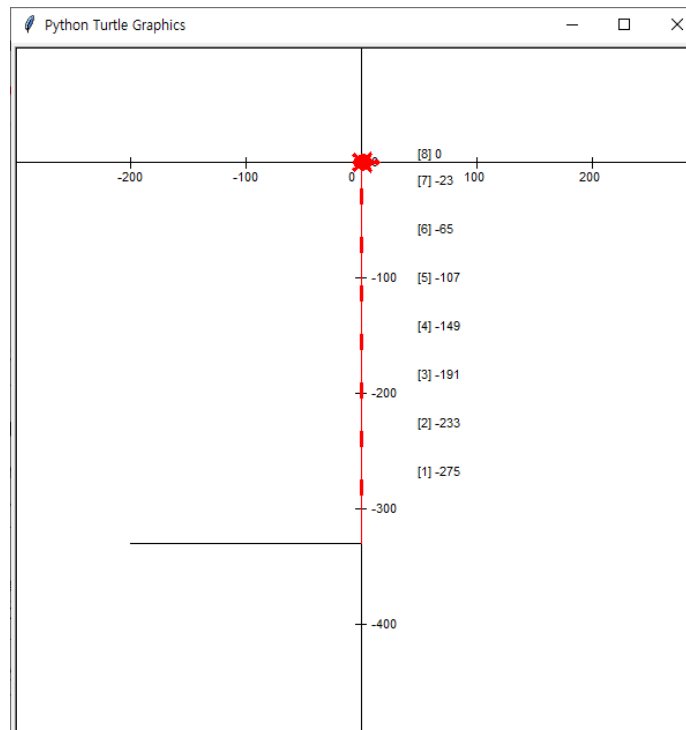
- ◆ 우물에 빠진 달팽이가 우물 밖으로 빠져나오는데 걸리는 시간 출력
 - 우물의 깊이 (달팽이의 최초 위치) 값을 키보드로 입력 받는다.
 - 하루마다 한번씩 달팽이의 위치를 표시(경과 일수 포함)
 - 목표에 근접해서는 남은 거리만 올라가고, 다 올라가서 오른쪽으로 90도 회전
- ❖ [Ch06-xyShift좌표.py 코드 참고](#)

위치 값 입력

우물 깊이 값 입력(cm, 양수)?

330

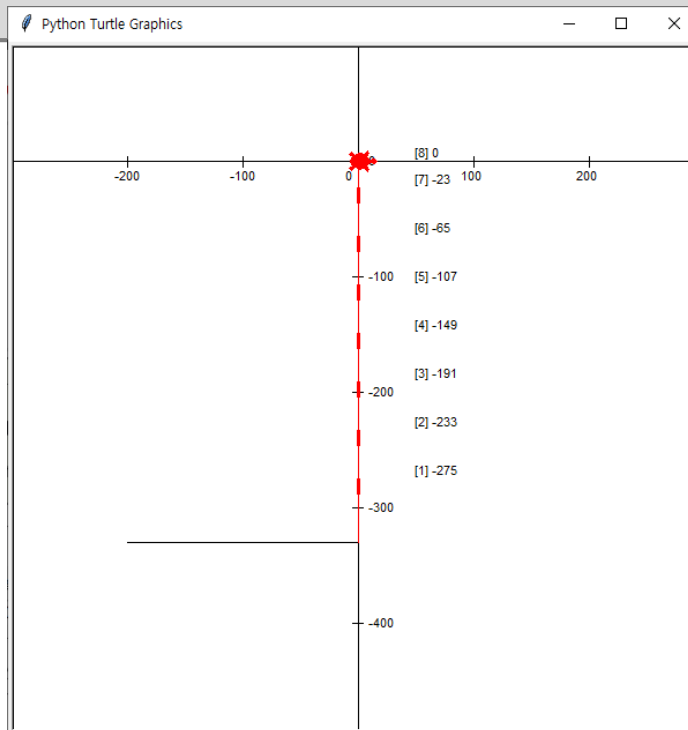
OK Cancel



03. Turtle Graphics Module 활용 예제

[실습] 달팽이 우물에서 탈출하기 (1)

```
import turtle as t
w_ysize = 600    #window 가로 크기 pixel
w_hsize = 600    #window 세로 크기 pixel
win = t.getscreen()  #Window screen 객체 생성
win.setup(w_ysize, w_hsize)
```



##변수 초기화

`l_shift = 0` #x축 left로 shift pixel 값

`d_shift = -200` #y축 down으로 shift pixel 값

`wsizer = int(w_ysize / 2)` #x-축 (+)축 길이

`hsize = int(w_hsize / 2)` #y-축 (+)축 길이

`step = 100` #눈금 간격 pixel

x, y축 좌표 그리기

`t.hideturtle()`

`t.speed(0)`

`draw_xy(wsize, hsize, step)`

03. Turtle Graphics Module 활용 예제

[실습] 달팽이 우물에서 탈출하기 (2)

```
##토끼, 거북이 객체 생성하기
dal = t.Turtle() #turtle 객체 생성
dal.hideturtle()
dal.penup()
dal.shape('turtle')
dal.color('red')

##게임 시작
p_dal = int(t.numinput("위치 값 입력", "우물 깊이 값 입력(cm, 양수)? "))

run_game(dal, p_dal)
```

03. Turtle Graphics Module 활용 예제

[실습] 달팽이 우물에서 탈출하기 (3)

```
##[함수] running game
def run_game(dal, p_dal) :
    p_dal = -p_dal #높이를 깊이로 환산

    dal.speed('fastest')
    dal.goto(newx(0), newy(p_dal))
    dal.left(90)
    dal.showturtle()
    dal.pendown()
    dal.speed('slowest')

    line(newx(-200), newy(p_dal), newx(0), newy(p_dal))    # x축 바닥 라인
```

03. Turtle Graphics Module 활용 예제

[실습] 달팽이 우물에서 탈출하기 (4)

```
# 달팽이 이동 처리
height = 0 # 우물 밖 목표 위치(cm)
days = 0
while p_dal < height:
    days += 1
    dal.pensize(1)
    p_dal += 55
    dal.forward(55)
    txtwrite(newx(50), newy(p_dal), '[' + str(days) + ']' + str(p_dal)) # 달팽이 현재 위치(m)
    print("%2d일째>%5dcm" % (days, p_dal), end=" ")
    if p_dal >= height:
        break
    p_dal -= 13
    dal.pensize(3)
    dal.backward(13)
    print(">%5dcm" % (p_dal))

print("\n>>%d일만에 탈출 성공" % days)
```

```
1일째> -345cm > -358cm
2일째> -303cm > -316cm
3일째> -261cm > -274cm
4일째> -219cm > -232cm
5일째> -177cm > -190cm
6일째> -135cm > -148cm
7일째> -93cm > -106cm
8일째> -51cm > -64cm
9일째> -9cm > -22cm
10일째> 33cm
>>10일만에 탈출 성공
```

03. Turtle Graphics Module 활용 예제

[도전-2] 달팽이 우물에서 탈출하기 (완성)

Ch06-xy달팽이.py

- ◆ 우물에 빠진 달팽이가 우물 밖으로 빠져나오는데 걸리는 기간 출력
 - 우물의 깊이 (달팽이의 최초 위치) 값을 키보드로 입력 받는다.
 - 하루마다 한번씩 달팽이의 위치를 표시(경과 일수 포함)
 - **목표에 근접해서는 남은 거리만 올라가고, 다 올라가서 오른쪽으로 90도 회전**

달팽이 이동 처리

height = 0 # 우물 밖 목표 위치(cm)

days = 0

while p_dal < height:

 days += 1

 dal.pensize(1)

 p_dal += 55

 dal.forward(55)

 txtwrite(newx(50), newy(p_dal), '[' + str(days) + ']' + str(p_dal)) # 달팽이 현재 위치(m)

 print("%2d일 짜>%5dcm" % (days, p_dal), end=" ")

 if p_dal >= height:

 break

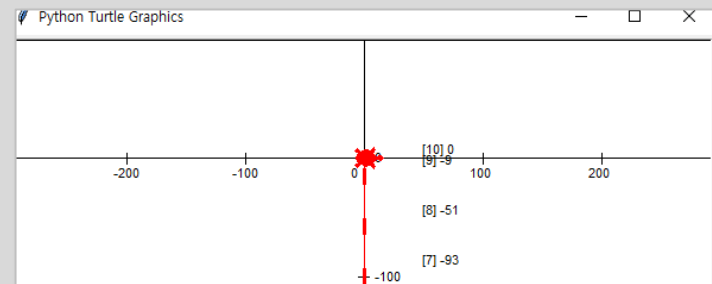
 p_dal -= 13

 dal.pensize(3)

 dal.backward(13)

 print(">%5dcm" % (p_dal))

print("\n n>>%d일만에 탈출 성공" % days)



Thank You !

[Python]