

# ch0

2019년 7월 20일

## 1 다각형 그리기

```
In [7]: import turtle as t
```

```
n = 30 # n각형을 그립니다 (다른 값을 입력하면 다른 도형을 그립니다).
t.color("purple")
t.begin_fill() # 색칠할 영역을 시작합니다.
for x in range(n): # n번 반복합니다.
    t.forward(20) # 거북이가 50만큼 앞으로 이동합니다.
    t.left(360 / n) # 거북이가 360/n만큼 왼쪽으로 회전합니다.
t.end_fill() # 색칠할 영역을 마무리합니다.
```

## 2 원 반복해서 그리기

```
In [8]: import turtle as t
```

```
n = 50 # 원을 50개 그립니다.
t.bgcolor("black") # 배경색을 검은색으로 지정합니다.
t.color("green") # 펜 색을 녹색으로 지정합니다.
t.speed(0) # 거북이 속도를 가장 빠르게 지정합니다.
for x in range(n): # n번 반복합니다.
    t.circle(80) # 현재 위치에서 반지름이 80인 원을 그립니다.
    t.left(360/n) # 거북이가 360/n만큼 왼쪽으로 회전합니다.
```

### 3 선 반복해서 그리기

```
In [9]: import turtle as t
```

```
angle = 89          # 거북이가 왼쪽으로 회전할 각도를 지정합니다 (값을 바꿀 수 있음).
t.bgcolor("black")  # 배경색을 검은색으로 지정합니다.
t.color("yellow")    # 펜 색을 노란색으로 지정합니다.
t.speed(0)           # 거북이 속도를 가장 빠르게 지정합니다.
for x in range(200): # x 값을 0에서 199까지 바꾸면서 200번 실행합니다.
    t.forward(x)     # x만큼 앞으로 이동합니다 (실행을 반복하면서 선이 길어짐).
    t.left(angle)     # 거북이가 왼쪽으로 89도 회전합니다.
```

### 4 키보드로 거북이 조종하기

```
In [ ]: import turtle as t
```

```
def turn_right():    # 오른쪽으로 이동하는 함수
    t.setheading(0)  # t.seth(0)으로 입력해도 됩니다.
    t.forward(10)     # t.fd(10)으로 입력해도 됩니다.

def turn_up():       # 위로 이동하는 함수
    t.setheading(90)
    t.forward(10)

def turn_left():     # 왼쪽으로 이동하는 함수
    t.setheading(180)
    t.forward(10)

def turn_down():     # 아래로 이동하는 함수
    t.setheading(270)
    t.forward(10)

def blank():         # 화면을 지우는 함수
    t.clear()

t.shape("turtle")    # 거북이 모양을 사용합니다.
```

```

t.speed(0) # 거북이 속도를 가장 빠르게 지정합니다.
t.onkeypress(turn_right, "Right") # →를 누르면 turn_right 함수를 실행합니다.
t.onkeypress(turn_up, "Up")
t.onkeypress(turn_left, "Left")
t.onkeypress(turn_down, "Down")
t.onkeypress(blank, "Escape") # ESC를 누르면 blank 함수를 실행합니다.
t.listen() # 거북이 그래픽 창이 키보드 입력을 받습니다.
t.mainloop()

```

## 5 마우스로 거북이 조종하기

In [4]: `import turtle as t`

```

t.speed(0) # 거북이의 속도를 가장 빠르게 지정합니다.
t.pensize(2) # 펜 굵기를 2로 지정합니다.
t.hideturtle() # 거북이를 화면에서 숨깁니다.
t.onscreenclick(t.goto) # 마우스 버튼을 누르면 t.goto 함수를 호출합니다.
# 그 위치로 거북이가 움직이면서 선을 그립니다.
t.mainloop() # jupyter, pycharm 등에서 실행 시 사용합니다.

```

## 6 turtle 모듈의 주요 함수 알아보기

### 6.1 forward(거리), fd(거리)

거북이가 앞으로 이동합니다.

### 6.2 backward(거리), back(거리)

거북이가 뒤로 이동합니다.

### 6.3 left(각도) / lt(각도)

거북이가 왼쪽으로 회전합니다.

### 6.4 right(각도) / rt(각도)

거북이가 오른쪽으로 회전합니다.

## 6.5 circle(반지름)

현재 위치에서 원을 그립니다.

## 6.6 down() / pendown()

펜을 내려서 선을 그립니다.

## 6.7 up() / penup()

펜을 올려서 선을 그리지 않습니다.

## 6.8 shape("모양")

거북이 모양을 바꿉니다.

turtle, arrow, circle, square, triangle 등

## 6.9 speed(속도)

거북이 속도를 바꿉니다.

1부터 10까지 점점 커집니다.

단, 최고 속도는 0입니다.

## 6.10 pensize(굵기) / width

펜 사이즈를 바꿉니다.

## 6.11 color("색 이름")

펜 색을 바꿉니다.

## 6.12 bgcolor("색 이름")

배경 색을 바꿉니다.

## 6.13 fillcolor("색 이름")

도형 내부를 칠하는 색을 바꿉니다.

## 6.14 begin\_fill()

도형 내부 색칠을 시작합니다.

### **6.15 end\_fill()**

도형 내부 색칠을 끝냅니다.

### **6.16 showturtle() / st()**

거북이를 표시합니다.

### **6.17 hideturtle() / ht()**

거북이를 숨깁니다.

### **6.18 clear()**

거북이를 그대로 둔 채 화면을 지웁니다.

### **6.19 reset()**

거북이를 원래 자리로 돌리고 화면을 지웁니다.

### **6.20 pos()**

거북이의 현재 위치(x,y 좌표 모두)를 구합니다.

### **6.21 xcor(), ycor()**

거북이의 현재 위치 (x,y 좌표 중 하나)를 구합니다.

### **6.22 goto(x, y), setpos(x, y)**

거북이를 특정 위치(x,y 좌표 모두)로 옮깁니다.

### **6.23 setx(x), sety(y)**

거북이를 특정 위치(x,y 좌표 중 하나)로 옮깁니다.

### **6.24 distance(x, y)**

거북이의 현재 위치에서 입력받은 위치 사이 거리를 구합니다.

### **6.25 heading()**

거북이가 현재 바라보고 있는 각도를 구합니다.

### 6.26 towards(x, y)

거북이의 현재 위치에서 특정 위치를 바라보는 각도를 구합니다.

### 6.27 setheading(각도)/ seth(각도)

거북이가 바라보는 각도를 바꿉니다.

### 6.28 home()

거북이의 위치와 방향을 처음 상태로 돌립니다.

### 6.29 onkeypress(함수, “키 이름”)

키보드를 눌렀을 때 실행할 함수를 정합니다.

### 6.30 onclick(함수)

마우스 버튼을 눌렀을 때 실행할 함수를 정합니다.

### 6.31 ontimer(함수, 시간)

일정한 시간이 지났을 때 실행할 함수를 정합니다.

시간의 단위는 1/1000초입니다.

### 6.32 listen()

사용자 입력이 잘 처리되도록 그래픽 창에 포커스를 줍니다.

### 6.33 title(“창 이름”)

거북이 그래픽 창의 이름을 지정합니다.

### 6.34 write(“문자열”)

현재 거북이 위치에 문자를 출력합니다.

## 7 터틀린 만들기

```
In [12]: import turtle as t
         import random
```

```

te = t.Turtle()                                # 악당 거북이 (빨간색)
te.shape("turtle")
te.color("red")
te.speed(0)
te.up()
te.goto(0, 200)

ts = t.Turtle()                                # 먹이 (초록색 동그라미)
ts.shape("circle")

ts.color("green")
ts.speed(0)
ts.up()
ts.goto(0, -200)

def turn_right():                              # 오른쪽으로 방향을 바꿉니다.
    t.setheading(0)

def turn_up():                                  # 위로 방향을 바꿉니다.
    t.setheading(90)

def turn_left():                               # 왼쪽으로 방향을 바꿉니다.
    t.setheading(180)

def turn_down():                               # 아래로 방향을 바꿉니다.
    t.setheading(270)

def play():                                    # 게임을 실제로 플레이하는 함수
    t.forward(10)                              # 주인공 거북이가 10만큼 앞으로 이동합니다.
    ang = te.towards(t.pos())
    te.setheading(ang)                          # 악당 거북이가 주인공 거북이를 바라보게 합니다.
    te.forward(9)                              # 악당 거북이가 9만큼 앞으로 이동합니다.
    if t.distance(ts) < 12:                    # 주인공과 먹이의 거리가 12보다 작으면 (가까우면)
        star_x = random.randint(-230, 230)
        star_y = random.randint(-230, 230)
        ts.goto(star_x, star_y)               # 먹이를 다른 곳으로 옮깁니다.

```

```

        if t.distance(te) >= 12:
            t.ontimer(play, 100)
t.setup(500, 500)
t.bgcolor("orange")
t.shape("turtle")
t.speed(0)
t.up()
t.color("white")
t.onkeypress(turn_right, "Right") # →를 누르면 turn_right 함수를 실행합니다.
t.onkeypress(turn_up, "Up")
t.onkeypress(turn_left, "Left")
t.onkeypress(turn_down, "Down")
t.listen()
play()
t.mainloop()

```

# 주인공과 악당의 거리가 12 이상이면 (떨면)  
# 0.1초 후 play 함수를 실행합니다 (게임 계속).

# 거북이 모양의 커서를 사용합니다.  
# 거북이 속도를 가장 빠르게 지정합니다.

# 거북이 그래픽 창이 키보드 입력을 받도록 합니다.  
# play 함수를 호출해서 게임을 시작합니다.

## 8 터틀런 2 만들기

```

In [16]: import turtle as t
import random

```

```

score = 0
playing = False

te = t.Turtle()
te.shape("turtle")
te.color("red")
te.speed(0)
te.up()
te.goto(0, 200)

ts = t.Turtle()
ts.shape("circle")
ts.color("green")
ts.speed(0)
ts.up()

```

# 점수를 저장하는 변수  
# 현재 게임이 플레이 중인지 확인하는 변수

# 악당 거북이 (빨간색)

# 먹이 (초록색 동그라미)



```

ts.goto(0, -200)

def turn_right():                # 오른쪽으로 방향을 바꿉니다.
    t.setheading(0)

def turn_up():                   # 위로 방향을 바꿉니다.
    t.setheading(90)

def turn_left():                 # 왼쪽으로 방향을 바꿉니다.
    t.setheading(180)

def turn_down():                # 아래로 방향을 바꿉니다.
    t.setheading(270)

def start():                     # 게임을 시작하는 함수
    global playing
    if playing == False:
        playing = True
        t.clear()                # 메시지를 지웁니다.
        play()

def play():                      # 게임을 실제로 플레이하는 함수
    global score
    global playing
    t.forward(10)                # 주인공 거북이 10만큼 앞으로 이동합니다.
    if random.randint(1, 5) == 3: # 1~5 사이에서 뽑은 수가 3이면 (20% 확률)
        ang = te.towards(t.pos())
        te.setheading(ang)       # 악당 거북이가 주인공 거북이를 바라봅니다.
    speed = score + 5            # 점수에 5를 더해서 속도를 올립니다
    # 점수가 올라가면 빨라집니다.
    if speed > 15:               # 속도가 15를 넘지는 않도록 합니다.
        speed = 15
    te.forward(speed)
    if t.distance(te) < 12:      # 주인공과 악당의 거리가 12보다 작으면
        # 게임을 종료합니다.
        text = "Score : " + str(score)
        message("Game Over", text)

```

```

    playing = False
    score = 0
    if t.distance(ts) < 12:          # 주인공과 먹이의 거리가 12보다 작으면 (가까우면)
        score = score + 1           # 점수를 올립니다.
        t.write(score)              # 점수를 화면에 표시합니다.
        star_x = random.randint(-230, 230)
        star_y = random.randint(-230, 230)
        ts.goto(star_x, star_y)     # 먹이를 다른 곳으로 옮깁니다.
    if playing:
        t.ontimer(play, 100)        # 게임 플레이 중이면 0.1초 후
                                    # play 함수를 실행합니다.

def message(m1, m2):               # 메시지를 화면에 표시하는 함수
    t.clear()
    t.goto(0, 100)
    t.write(m1, False, "center", (" ", 20))
    t.goto(0, -100)
    t.write(m2, False, "center", (" ", 15))
    t.home()

t.title("Turtle Run")
t.setup(500, 500)
t.bgcolor("orange")
t.shape("turtle")                 # 거북이 모양의 커서를 사용합니다.
t.speed(0)                        # 거북이 속도를 가장 빠르게 지정합니다.
t.up()
t.color("white")
t.onkeypress(turn_right, "Right") # →를 누르면 turn_right 함수를 실행합니다.
t.onkeypress(turn_up, "Up")
t.onkeypress(turn_left, "Left")
t.onkeypress(turn_down, "Down")
t.onkeypress(start, "space")
t.listen()                        # 거북이 그래픽 창이 키보드 입력을 받도록 합니다.
message("Turtle Run", "[Space]")
t.mainloop()

```