파이썬 프로그래밍

29차시

모듈의 이해와 활용 2

⚠ 학습개요

… 모듈에서 내장 변수 __name__ 사용
… 모듈을 계층 폴더인 패키지로 구성
… 표준 모듈 turtle 활용

⚠ 학습목표

… 모듈 구현 시 내장 변수 __name__을 활용할 수 있다.
… 코드에서 if __name__ == ' __main__ ' 을 활용할 수 있다.
… 모듈을 계층인 여러 폴더에 구성해 패키지로 구현할 수 있다.
… 표준 모듈 turtle을 활용해 구현할 수 있다.

Chapter 1.

모듈에서 내장 변수 __name__ 사용

PYTHON PROGRAMMING

⚠ 내장 변수 __name__

+ 실행되는 모듈의 이름이 저장되는 변수

- 모듈 자체에서 시작되어 실행되면 __main__이 저장됨
- 다른 곳에서 호출(import 구문)이 된다면 자신의 모듈 이름이 저장

+ 소스 파일 코드에 다음이 있는 의미

- if __name__ == '__main__'
  - 소스 파일 자체를 실행한다면 True
  - 소스 파일을 모듈로 이용해 import 한다면 False

⚠ 내장 변수 __name__

+ 소스 code.py 실행 예

- python code.py
  - 소스 code.py가 직접 실행되는 경우
    - __name__ 내장 변수에는 __main__으로 설정
- import code
  - code.py가 다른 스크립트에 의해 import 되었을 경우
    - __name__은 모듈 이름인 code가 저장

```
PS D:\(1 Drive)\hskang\OneDrive - 동양미래대학\2021 파이썬 e-learning ppt 작성\연습 code\ch09> type code.py
print('Hello, code.py')
print(__name__)
PS D:\(1 Drive)\hskang\OneDrive - 동양미래대학\2021 파이썬 e-learning ppt 작성\연습 code\ch09> python code.py
Hello, code.py
__main__
PS D:\(1 Drive)\hskang\OneDrive - 동양미래대학\2021 파이썬 e-learning ppt 작성\연습 code\ch09>
```

⚠ 자신이 만든 모듈에서 직접 실행 점검

+ import hello2.py

- 그 실행 결과가 보임

```
>>> import hello2
Hi, Python!
파이썬, 재미있네요!
```

hello2.py

```
def hi():
    print('Hi, Python!')

msg = '파이썬, 재미있네요!'

hi()
print(msg)
```

⚠ 자신이 만든 모듈에서 직접 실행 점검

+ 조건 if __name__ == '__main__'

- if 블록 문장은 실행되지 않음
- __name__
  - 자신 모듈 이름인 'hello3'가 저장돼 있기 때문
- 시스템 변수 __name__
  - 실행 중인 모듈의 이름이 저장
- hello3.py가 직접 실행
  - __name__이 '__main__'으로 지정

hello3.py

```
def hi():
    print('Hi, Python!')
    print(__name__)

msg = '파이썬, 재미있네요!'

if __name__== '__main__':
    hi()
    print(msg)
```

```
>>> import hello3
>>> print(__name__)
__main__

>>> hello3.hi()
Hi, Python!
hello3
```

⚠ 모듈 random을 함수를 활용해 모듈을 생성하고 불러오기

[코딩실습] 모듈 random을 활용한 난수 발생 함수 구현

난이도 응용

```
1. import random
2.
3. def nrandom(start, end, n, duplicated = False):
4.     ...
5.     start와 end 사이의 정수 난수를 n개 생성해 반환
6.     인자
7.         start: 시작 정수
8.         end: 마지막 정수
9.         n: 난수 개수
10.        duplicated: 중복 허용 여부, 기본은 중복하지 않음
11.    ...
12.    lst = [] # 반환할 난수 리스트
```

⚠ 모듈 random을 함수를 활용해 모듈을 생성하고 불러오기

[코딩실습] 모듈 random을 활용한 난수 발생 함수 구현 난이도 응용

```
13.     if duplicated:
14.         for _ in range(n)
15.             1st.append(random.randint(start, end))
16.     else:
17.       1st = list(random.sample(range(start,end+1),n))
18.    # 모두 정렬해 반환
19.    return sorted(1st)
20.
21. if __name__ == '__main__':
22.    print('로또 복권:', nrandom(1, 45, 6))
23.    print('로또 복권:', nrandom(1, 6, 3, True))
```

결과

```
로또 복권: [3, 12, 24, 25, 30, 34]
주사위 3번: [2, 2, 3]
```

⚠ 모듈 random을 함수를 활용해 모듈을 생성하고 불러오기

[코딩실습] 직접 만든 모듈 kutil의 함수 nrandom 불러 활용 난이도 응용

```
1. from kutil import nrandom
2.
3. for i in range(5):
4.     print('로또 복권 %d:' % (i+1), nrandom(1, 45, 6))
5. print()
6. print('주사위 4번:', nrandom(1, 6, 4, True))
```

결과

```
로또 복권 1 : [3, 7, 12, 27, 38, 43]
로또 복권 2 : [7, 9, 15, 18, 19, 27]
로또 복권 3 : [15, 20, 26, 32, 35, 45]
로또 복권 4 : [9, 10, 19, 32, 35, 37]
로또 복권 5 : [6, 7, 8, 11, 14, 36]

주사위 4번: [2, 4, 5, 6]
```

끝

Chapter 2.

모듈을 계층 폴더인 패키지로 구성

PYTHON PROGRAMMING

⚠ 여러 계층 구조의 패키지 생성과 실행

✚ 여러 모듈을 패키지라는 폴더에 저장해 관리

- 즉, 패키지는 모듈이 저장된 폴더
  - 하부에도 여러 폴더와 모듈이 구성

상위 폴더: D:\Python Code\ch09		
패키지(폴더)	**모듈**	**소스코드**
myai	kai.py	#%% 모듈 파일 myai\kai.py def getAI(): print('저는 인공 지능 모듈이다.')
myai/ml	machine.py	#%% 모듈 파일 myai\ml\machine.py def getML(): print('저는 머신 러닝 모듈이다.')
myai/nn	neural.py	#%% 모듈 파일 myai\nn\neural.py def getANN(): print('저는 뉴럴네트웍 모듈이다.')

⚠ 여러 계층 구조의 패키지 생성과 실행

✚ 여러 모듈을 패키지라는 폴더에 저장해 관리

- 즉, 패키지는 모듈이 저장된 폴더
  - 하부에도 여러 폴더와 모듈이 구성

```
>>> import sys
>>> sys.path.append('D:\\Python Code\ch09')

>>> import myai.kai
>>> myai.kai.getAI()
저는 인공 지능 모듈이다.

>>> from myai.ml import machine
>>> machine.getML()
저는 머신 러닝 모듈이다.

>>> from myai.nn.neural import getANN
>>> getANN()
저는 뉴럴네트웍 모듈이다.
```

Chapter 3.

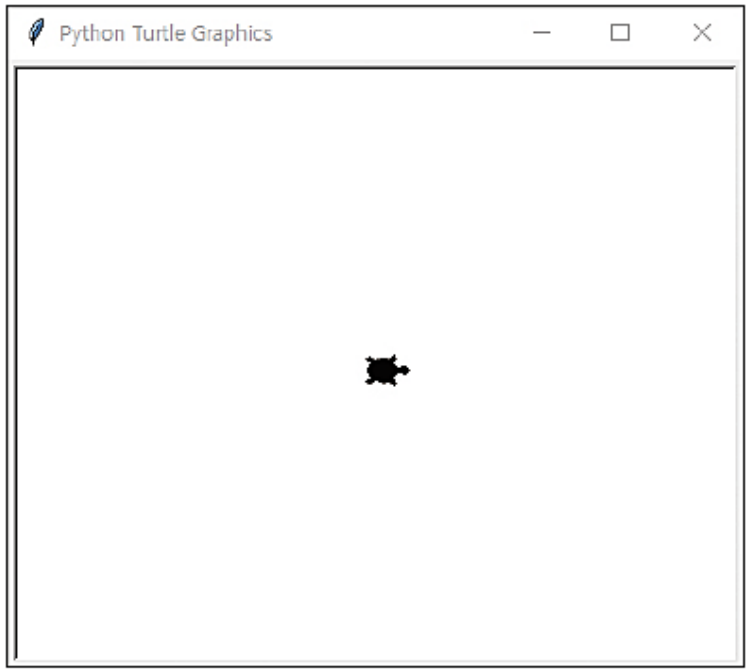
표준 모듈
turtle 활용

PYTHON PROGRAMMING

⚠ 모듈 turtle 개요와 기본 명령

✚ 1967년 아동 교육용으로 개발된 로고(logo)라는 프로그래밍 언어의 일부

```
>>> import turtle as t
>>> t.shape('turtle')
```

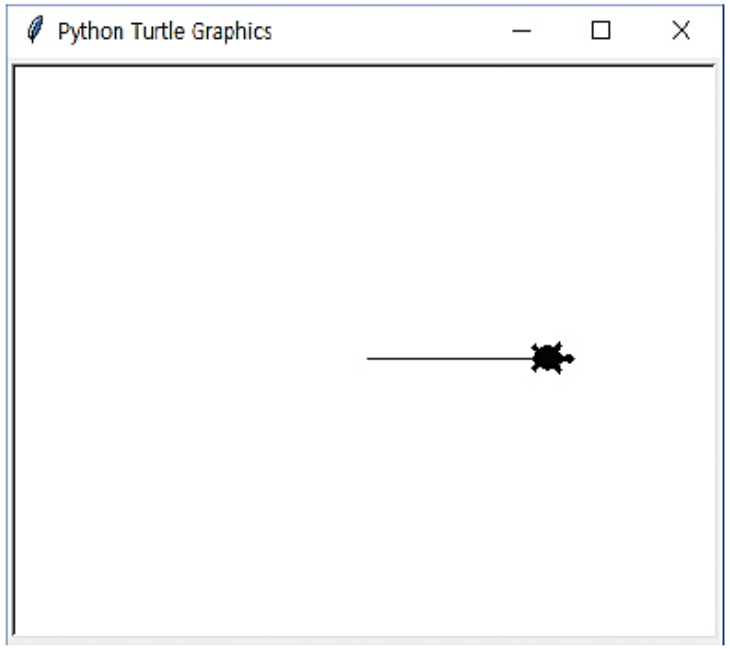


⚠ 모듈 turtle 개요와 기본 명령

✚ 1967년 아동 교육용으로 개발된 로고(logo)라는 프로그래밍 언어의 일부

```
>>> t.forward(100)
```

거북이 머리 반대 방향으로의 이동은 backward(픽셀)로 가능하다.
간단히 back()와 bk()로도 가능하다.



⚠ 거북이로 삼각형과 사각형 그리기

[코딩실습] 거북이로 삼각형과 사각형 그리기

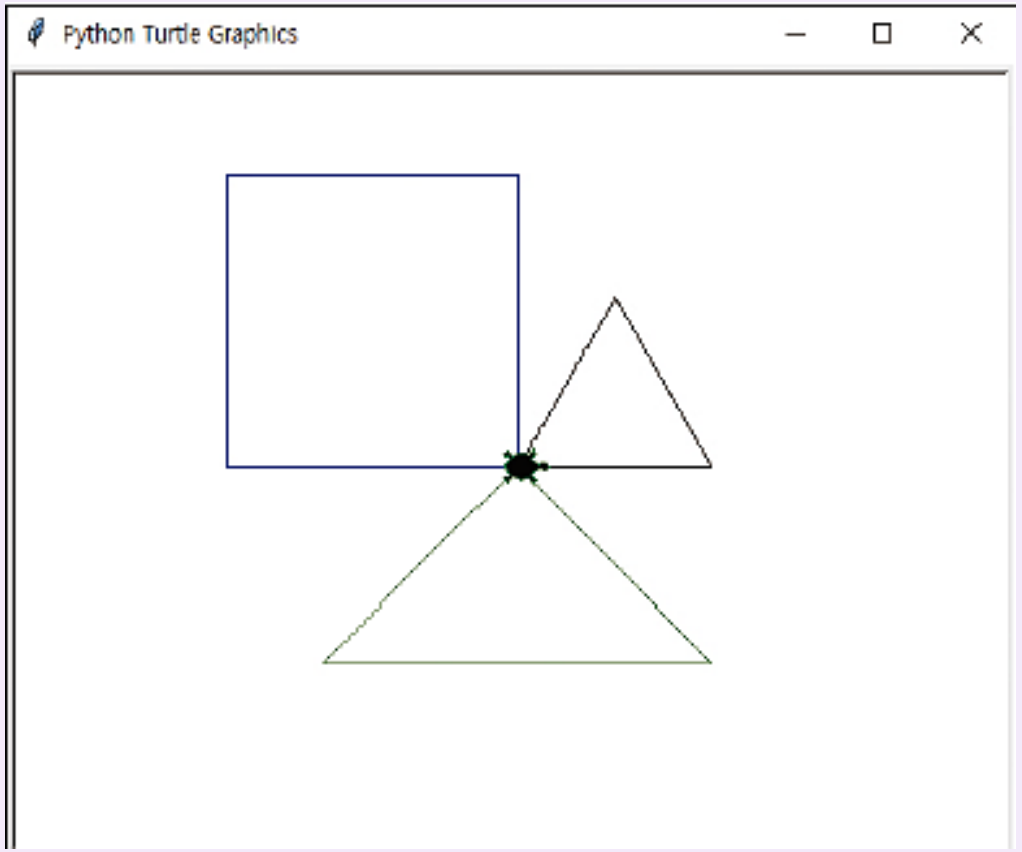
난이도 기본

```
1. import turtle as t
2. t.shape('turtle')
3.
4. #삼각형 그리기
5. t.forward(100)
6. t.left(120)
7. t.forward(100)
8. t.left(120)
9. t.forward(100)
10.t.left(120)
11.
12.#사각형 그리기
13.t.pencolor("blue")
14.for _ in range(4):
15.    t.left(90)
16.    t.forward(150)
17.
18.#삼각형 그리기
19.t.pencolor("green")
20.t.goto(100, -100)
21.t.goto(-100, -100)
22.t.home()
```

⚠ 거북이로 삼각형과 사각형 그리기

[코딩실습] 거북이로 삼각형과 사각형 그리기 난이도 기본

결과



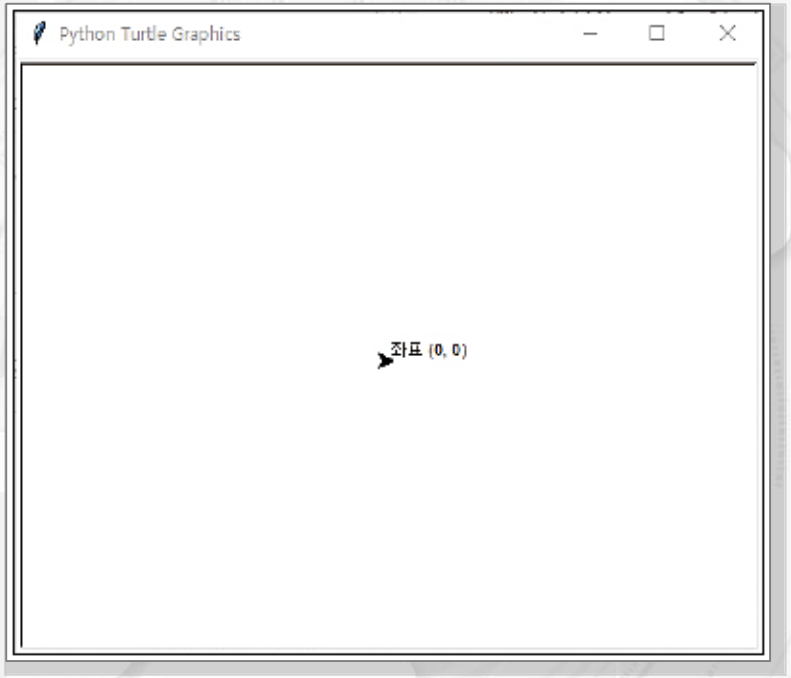
⚠ 터틀 윈도 크기와 터틀 모양 수정

+ 거북이 모양은 다음 모양이 'classic'으로 기본

- shape('turtle')로 예쁜 거북이 모양으로 수정
- 'classic', 'turtle', 'arrow', 'circle', 'square', 'triangle'

- 다음 코드로 가로 세로가 각각 500, 400인 윈도 위에, >와 비슷한 기본적인 터틀 모양으로 중앙에 위치하며, 좌표 (0,0)을 출력한다.

```
import turtle as t
t.setup(500, 400) # 초기 원도의 크기 조정
t.speed(1) # 1에서 10까지 거북이 속도 증가, 0이면 최고속
t.home() # 기본(classic 모양) 모양으로 (0,0)에 위치
t.write ('좌표 (0,0)') # 글씨 쓰기
```



⚠ 원도 중앙에 색상이 있는 원 그리기

[코딩실습] 원도 중앙에 색상이 있는 원 그리기

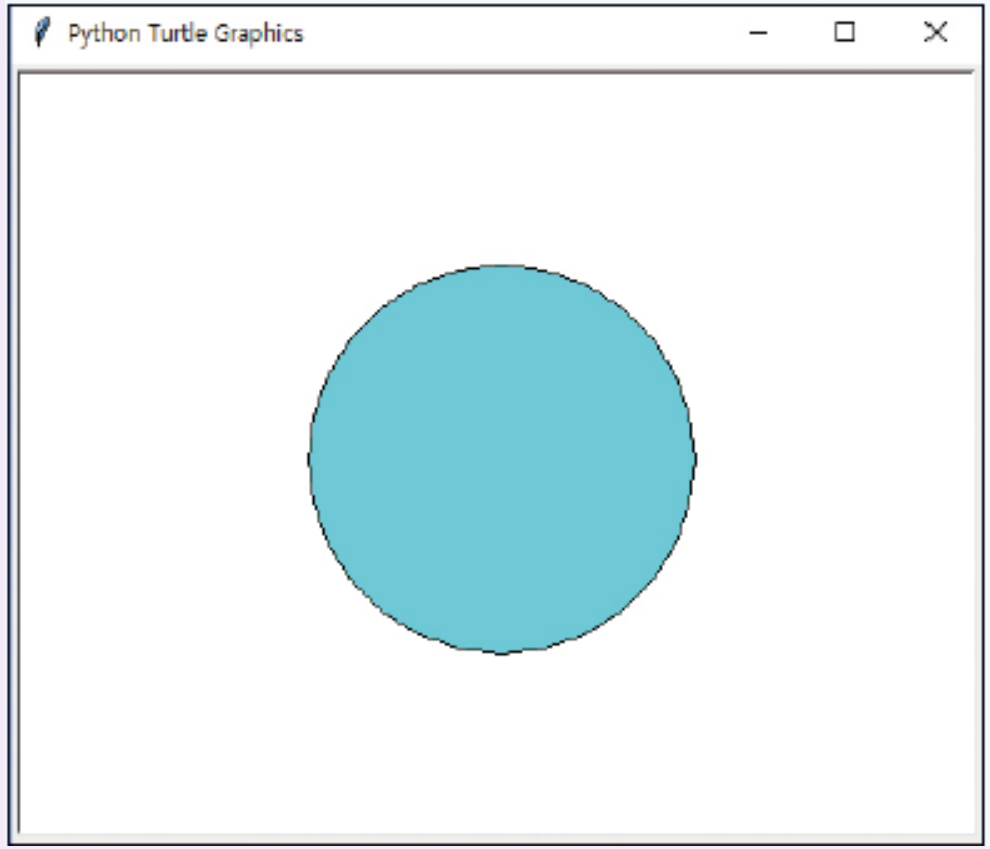
난이도 기본

```
1. import turtle as t
2. t.setup(500, 400) # 초기 원도의 크기 조정
3. t.speed(1) # 1에서 10까지 거북이 속도 증가, 0이면 최고속
4.
5. t.pu() # 이동에 선이 그려지지 않도록
6. t.goto(0, -100) # 이동
7.
8. t.pd() # 이동에 선이 그려지도록
9. t.hideturtle() # 거북이는 보이지 않도록
10.t.fillcolor('aqua') # 내부 칠할 색 지정
11.t.begin_fill() # 칠하기 시작
12.t.circle(100) # 원 그리기
13.t.end_fill() # 칠하기 종료
```

⚠ 원도 중앙에 색상이 있는 원 그리기

[코딩실습] 원도 중앙에 색상이 있는 원 그리기 난이도 기본

결과



⚠ 함수 정의로 다각형 그리기

[코딩실습] 함수 정의로 다각형 그리기

난이도 기본

```
1. import turtle as t
2. # 색상 리스트
3. cols = ['red', 'blue', 'green', 'purple', 'magenta', 'black',
4.         'gray', 'yellow', 'cyan', 'orange', 'aqua']
5.
6. def drawpolygon(n, size):
7.     ''' 한 변의 길이가 size인 n각형 그리기 '''
8.     for i in range(n):
9.         t.pencolor(cols[i % len(cols)]) # 펜 색상 지정
10.        t.forward(size) # 선 그리기
11.        t.left(360/n) # 각도 수정
12.
13.t.setup(500, 400) # 초기 원도의 크기 조정
14.t.speed(3) # 1에서 10까지 거북이 속도 증가, 0이면 최고속
15.
```

⚠ 함수 정의로 다각형 그리기

[코딩실습] 함수 정의로 다각형 그리기

난이도 기본

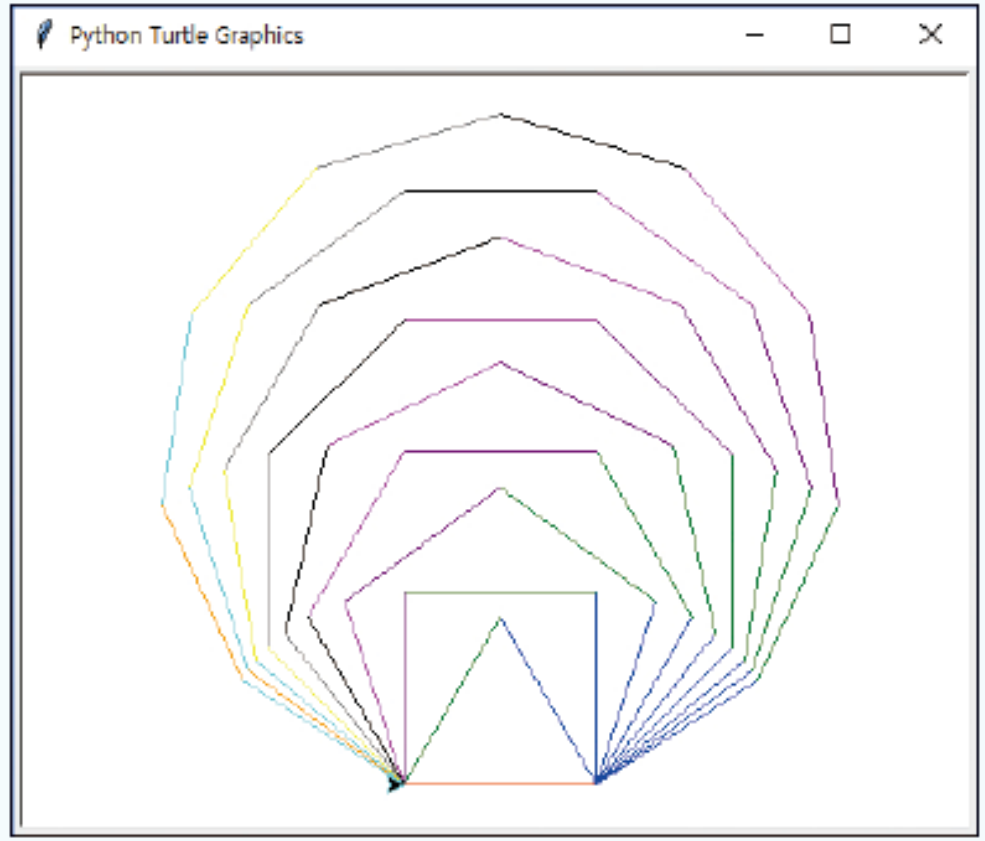
```
16.t.pu() # 이동에 선이 그려지지 않도록
17.t.goto(-50, -170) # 처음 위치로 이동
18.
19.t.pd() # 다각형을 그리기 위해 꼬리를 내리고
20.# 한 변의 길이가 100인 다각형 그리기
21.for i in range(3, 12):
22.    drawpolygon(i, 100) # 함수 호출
```

⚠ 함수 정의로 다각형 그리기

[코딩실습] 함수 정의로 다각형 그리기

난이도 기본

결과



⚠ 모듈에서 내장 변수 __name__ 사용

… 실행되는 모듈의 이름이 저장되는 변수

… 모듈 자체에서 시작되어 실행되면 __main__이 저장

… 다른 곳에서 호출이 된다면 자신의 모듈 이름이 저장

⚠ 모듈을 계층 폴더인 패키지로 구성

⚠ 표준 모듈 turtle 활용