



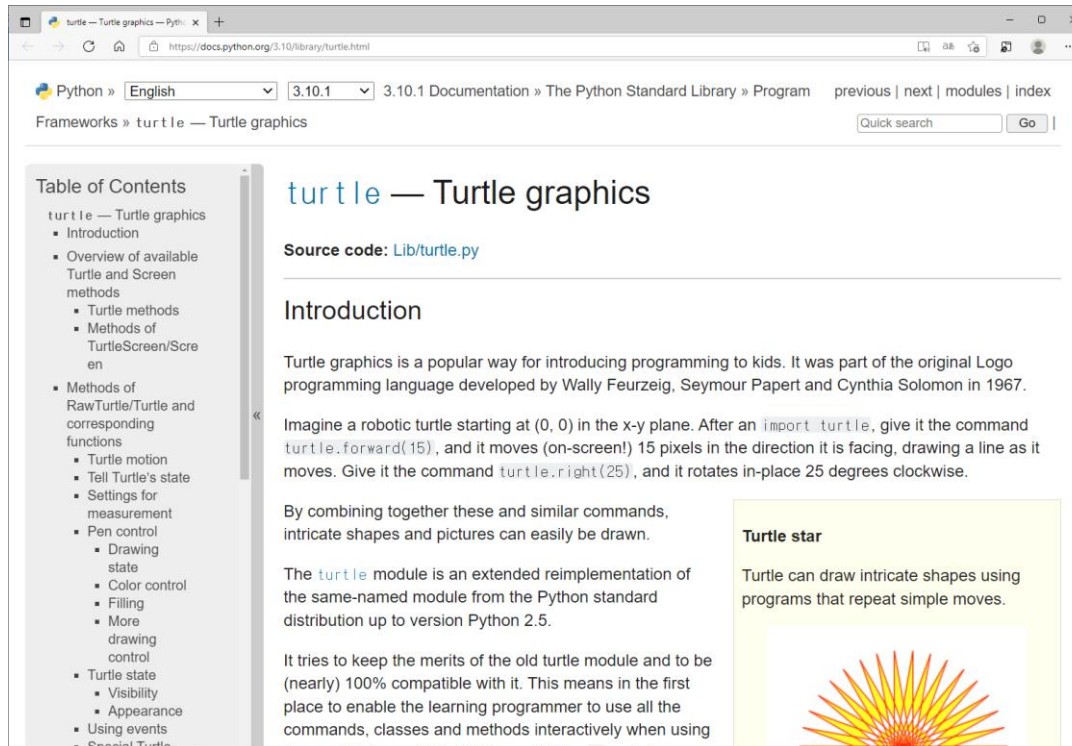
Python으로 배우는 소프트웨어 원리

Appendix 04-2. Turtle 그래픽스 모듈 활용
- 마우스 클릭 함수 호출 -

01. Turtle Graphics Module

[Turtle Graphics Module]

- 파이썬에서는 기본적인 그래픽 함수들을 사용할 수 있는 모듈을 제공 한다.
- [turtle — Turtle graphics — Python 3.10.1 documentation](https://docs.python.org/3.10/library/turtle.html) 사이트
 - <https://docs.python.org/3.10/library/turtle.html>
- 거북이의 이동을 중심으로 다양한 그래픽 구현이 가능하도록 각종 메소드(method)들을 제공한다.



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]

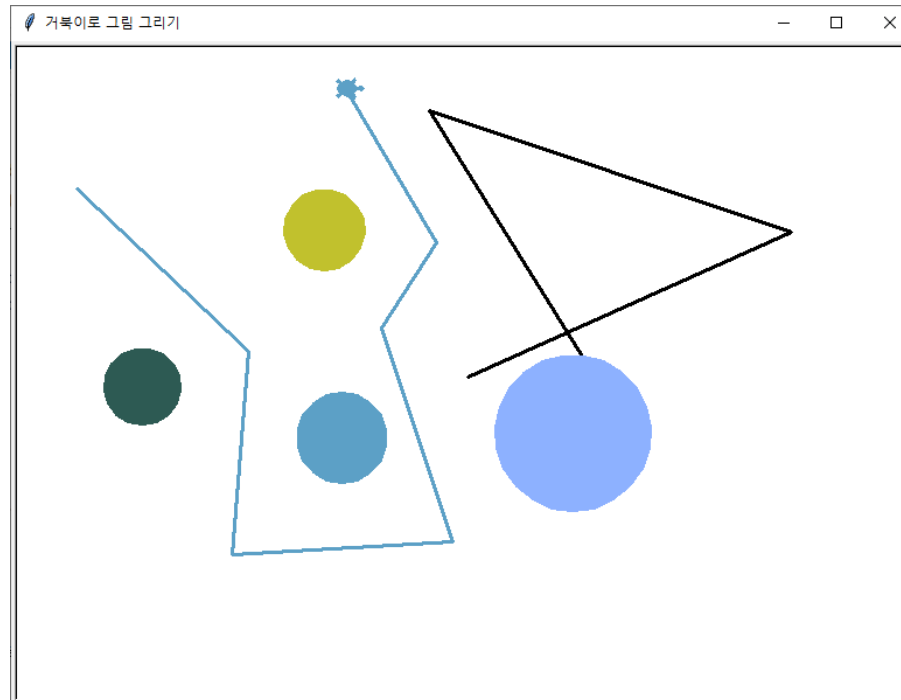
표 6-5 캔버스와 윈도우 설정 관련 메소드

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. Turtle Graphics Module 활용 예제

[활용 예제 요구사항] 마우스 클릭으로 이동

- [기능] 마우스 선택 키에 따른 거북이 이동 및 도형 그리기
 - [왼쪽 버튼 선택] pendown() 후, 커서 위치로 거북이를 이동
 - [오른쪽 버튼 선택] penup() 후, 커서 위치로 거북이를 이동
 - [가운데 버튼 선택] penup() 후, 선택, 면색 바꾸기
 - ✓ 선과 면의 색상은 랜덤한 값을 사용



03. Turtle Graphics Module 활용 예제

Ch04-TurtleMouse.py

[Python] 실습

- import turtle
- turtle.title('title')
- turtle.pensize(sz)
- turtle.onscreenclick(fn, 1)
- turtle.done()

```
import turtle
import random

## 함수 선언 부분 ##
def screenLeftClick(x, y):
def screenMidClick(x, y):
def screenRightClick(x, y):
```

- ## Turtle 모듈 임포트
- ## 윈도우 타이틀 명시
- ## 펜 굵기 지정(정수, 클수록 굵음, default=1)
- ## 마우스 왼쪽 버튼(1) 클릭 시 fn() 함수 호출
- ## Event Loop 이벤트를 계속 받아 처리

```
## 변수 선언 부분 ##
pSize = 3
r, g, b = 0.0, 0.0, 0.0

## 메인 코드 부분 ##
turtle.title('거북이로 그림 그리기')
turtle.shape('turtle')
turtle.pensize(pSize)

turtle.onscreenclick(screenLeftClick, 1)
turtle.onscreenclick(screenMidClick, 2)
turtle.onscreenclick(screenRightClick, 3)

turtle.done()
```

03. Turtle Graphics Module 활용 예제

[Python] 실습

Ch04-TurtleMouse.py

- import turtle
- turtle.pendown()
- turtle.up()
- turtle.goto(x, y)
- turtle.onscreenclick(fn, 2)

- ## 난수 발생 모듈 임포트
- ## 펜을 내려서 그리기 시작
- ## 펜을 올려서 그리기 종료
- ## 화면 (x, y) 좌표 점으로 이동
- ## 마우스 가운데 버튼(2) 클릭 시 fn() 함수 호출

```
import turtle
import random

## 함수 선언 부분 ##
def screenLeftClick(x, y):
    turtle.pendown()
    turtle.goto(x, y)

def screenMidClick(x, y):

def screenRightClick(x, y):
    turtle.pendown()
    turtle.goto(x, y)
```

```
## 변수 선언 부분 ##
pSize = 3
r, g, b = 0.0, 0.0, 0.0

## 메인 코드 부분 ##
turtle.title('거북이로 그림 그리기')
turtle.shape('turtle')
turtle.pensize(pSize)

turtle.onscreenclick(screenLeftClick, 1)
turtle.onscreenclick(screenMidClick, 2)
turtle.onscreenclick(screenRightClick, 3)

turtle.done()
```


03. Turtle Graphics Module 활용 예제

Ch04-TurtleMouse.py

[Python] 실습

- global r, g, b
- random.random()
- turtle.pencolor((r, g, b))
- turtle.fillcolor((r, g, b))
- turtle.onscreenclick(fn, **3**)

```
import turtle
import random

## 함수 선언 부분 ##
def screenMidClick(x, y):
    global r, g, b
    r = random.random()
    g = random.random()
    b = random.random()
    turtle.pencolor((r, g, b))
    turtle.fillcolor((r, g, b))
```

- ## 외부 변수를 함수 내부에서 사용
- ## 난수 발생(0에서부터 1보다 작은 실수)
- ## 선 색을 (r, g, b) 색상으로 변경
- ## 면 색을 (r, g, b) 색상으로 변경
- ## 마우스 오른쪽 버튼(**3**) 클릭 시 fn() 함수 호출

```
## 변수 선언 부분 ##
pSize = 3
r, g, b = 0.0, 0.0, 0.0

## 메인 코드 부분 ##
turtle.title('거북이로 그림 그리기')
turtle.shape('turtle')
turtle.pensize(pSize)

turtle.onscreenclick(screenLeftClick, 1)
turtle.onscreenclick(screenMidClick, 2)
turtle.onscreenclick(screenRightClick, 3)

turtle.done()
```

03. Turtle Graphics Module 활용 예제

[Python] 실습

Ch04-TurtleMouse.py

```
import turtle
import random

## 함수 선언 부분 ##
def screenLeftClick(x, y):
    turtle.pendown()
    turtle.goto(x, y)

def screenRightClick(x, y):
    turtle.penup()
    turtle.goto(x, y)

def screenMidClick(x, y):
    global r, g, b
    r = random.random()
    g = random.random()
    b = random.random()
    turtle.pencolor((r, g, b))
    turtle.fillcolor((r, g, b))
```

```
## 변수 선언 부분 ##
pSize = 3
r, g, b = 0.0, 0.0, 0.0

## 메인 코드 부분 ##
turtle.title('거북이로 그림 그리기')
turtle.shape('turtle')
turtle.pensize(pSize)

turtle.onscreenclick(screenLeftClick, 1)
turtle.onscreenclick(screenMidClick, 2)
turtle.onscreenclick(screenRightClick, 3)

turtle.done()
```

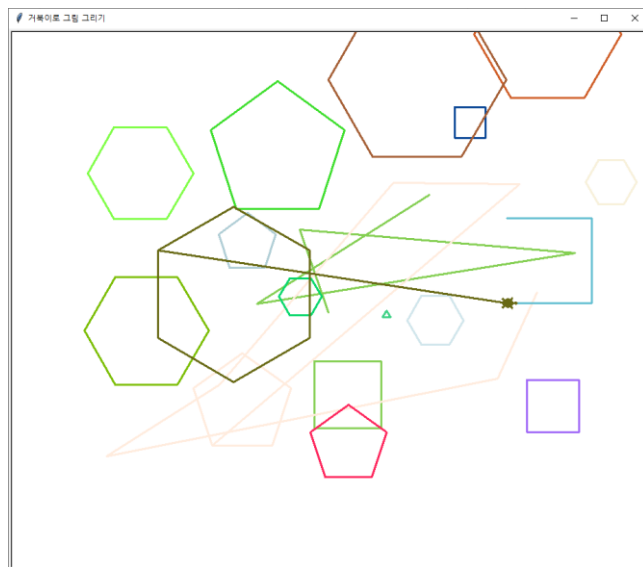
03. Turtle Graphics Module 활용 예제

Ch04-TurtleMouse다각형.py

[실습] 가운데 버튼으로 다각형 그리기

❖ 마우스 가운데 버튼 클릭으로 이동 후 정다각형 그리기

- [기능] 마우스 선택 키에 따른 거북이 이동 및 도형 그리기
 - [왼쪽 버튼 선택] pendown() 후, 커서 위치로 거북이를 이동
 - [오른쪽 버튼 선택] penup() 후, 커서 위치로 거북이를 이동
 - [가운데 버튼 선택] penup() 후, 커서 위치로 거북이를 이동 후, 선택, 면색 바꾸기
pendown() 후, 정 다각형 그리기
- ✓ 선과 면의 색상, n-각형, 면 길이는 랜덤한 값을 사용



Thank You !

[Python]