

문제해결을 위한 코딩 첫걸음

⋮ 3장 그래픽(Graphic)에 대해 알아보시다_터틀 그래픽

한성대학교 노은희 교수

서울세계도서관정보학회 도서관방문기관으로 선정



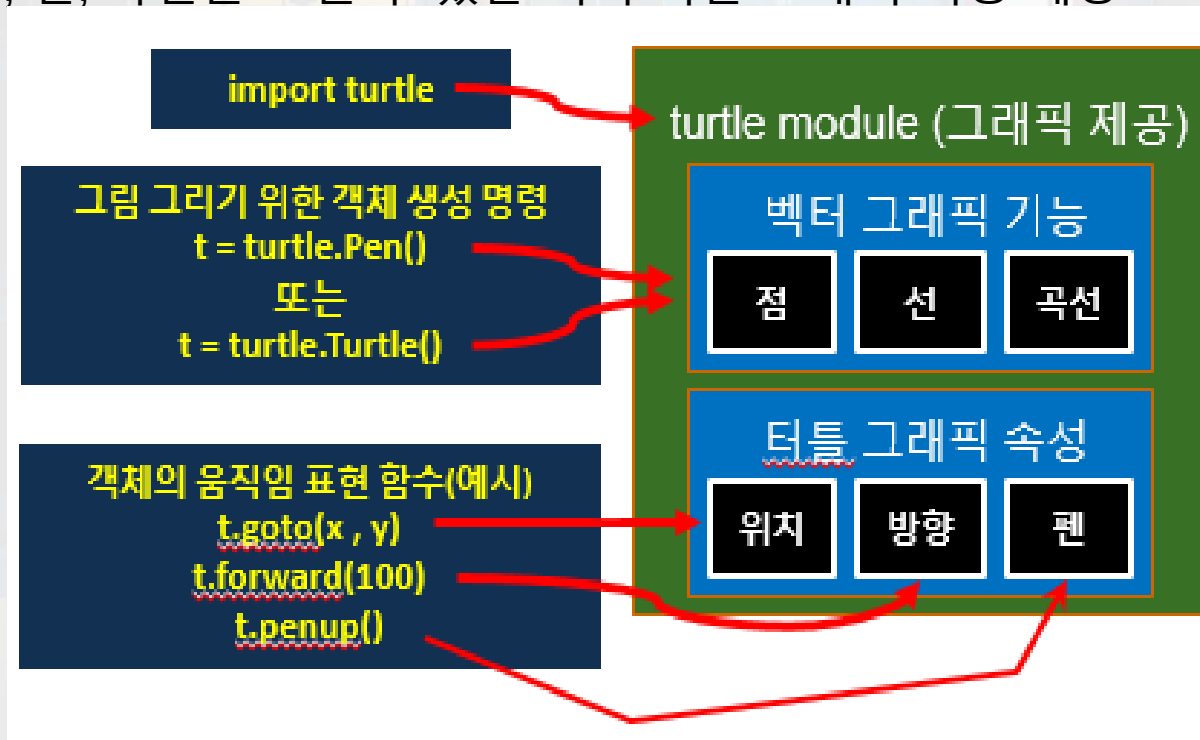
학습 목표

- 터틀 그래픽 환경과 터틀 모듈의 개념을 이해합니다.
- 터틀 모듈에서 제공하는 함수의 의미와 작성규칙을 이해합니다.



3.1 터틀 그래픽(turtle graphic)

- Turtle 모듈
 - Turtle는 그래픽 환경에서 프로그래밍을 쉽게 배울 수 있도록 파이썬에서 제공하는 모듈
 - 마우스나 키보드를 이용하여 그림을 그릴 수 있도록 지원해주는 모듈
 - 화면에서 커서가 지나간 흔적을 통해 그림을 그림
 - 점, 선, 곡선을 그릴 수 있는 벡터 기반 그래픽 기능 제공

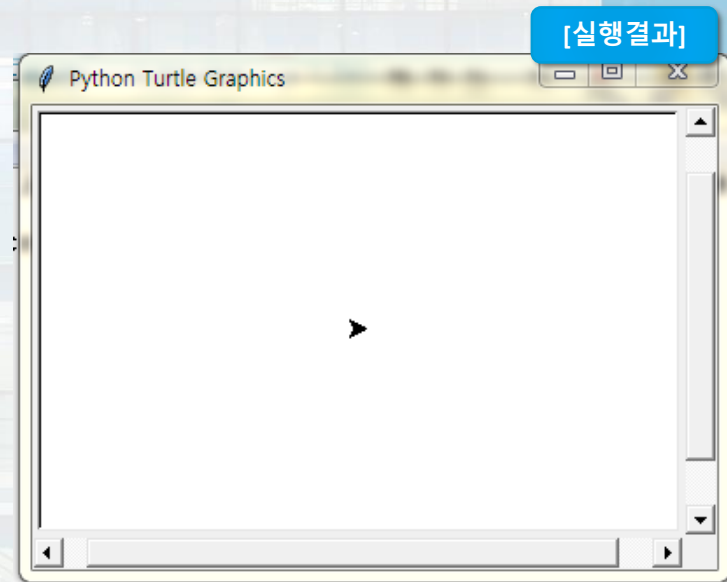
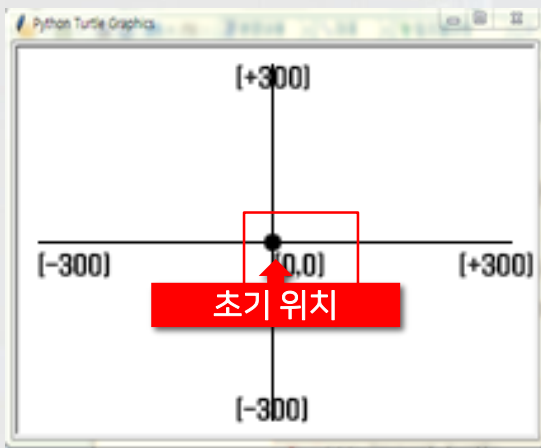


3.2 터틀 그래픽(turtle graphic) 활용

- 터틀 그래픽으로 그림 그리기
 - import 명령을 통해 turtle 모듈을 작업 환경으로 가지고 옴

```
>>> import turtle  
>>> t=turtle.Turtle()
```







- 초기 커서의 위치 (x좌표=0, y좌표=0)



[실행결과]

3.2 터틀 그래픽 활용 - 커서 모양 변경

- 커서의 모양 변경하기
 - turtle.shape()

종류	형태	예시 명령
classic(기본형태)		<code>turtle.shape("classic")</code>
arrow		<code>turtle.shape("arrow")</code>
turtle		<code>turtle.shape("turtle")</code>
circle		<code>turtle.shape("circle")</code>
square		<code>turtle.shape("square")</code>
triangle		<code>turtle.shape("triangle")</code>

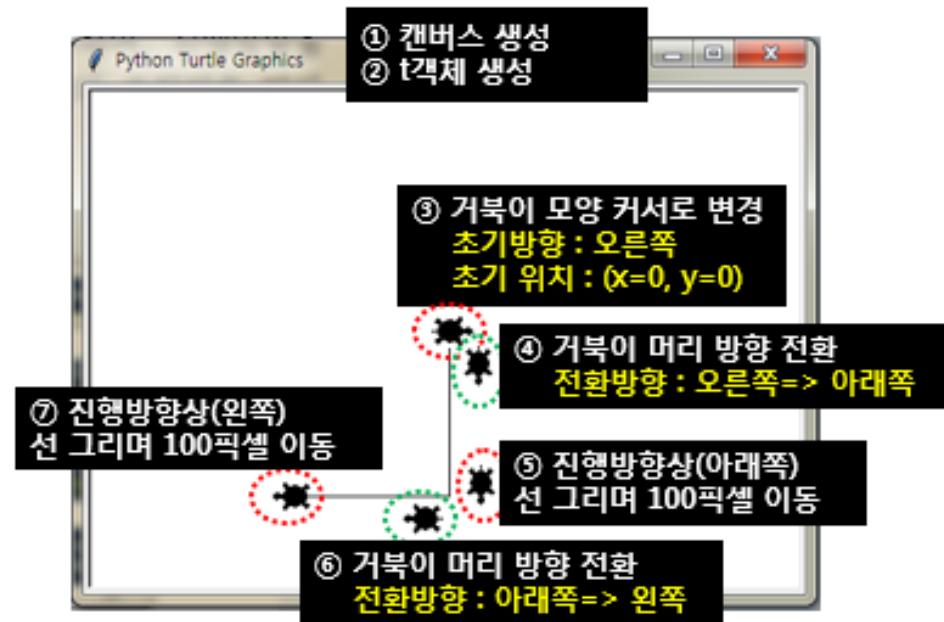


3.2 터틀 그래픽 활용 - 이동과 회전

- 이동과 회전
 - `forward(이동길이)` : 진행방향으로 이동
 - `left(회전각도)` : 왼쪽 방향으로 회전
 - `right(회전각도)` : 오른쪽 방향으로 회전

python shell

```
① >>> import turtle
② >>> t=turtle.Turtle()
③ >>> t.shape("turtle")
④ >>> t.right(90)
⑤ >>> t.forward(100)
⑥ >>> t.right(90)
⑦ >>> t.forward(100)
```





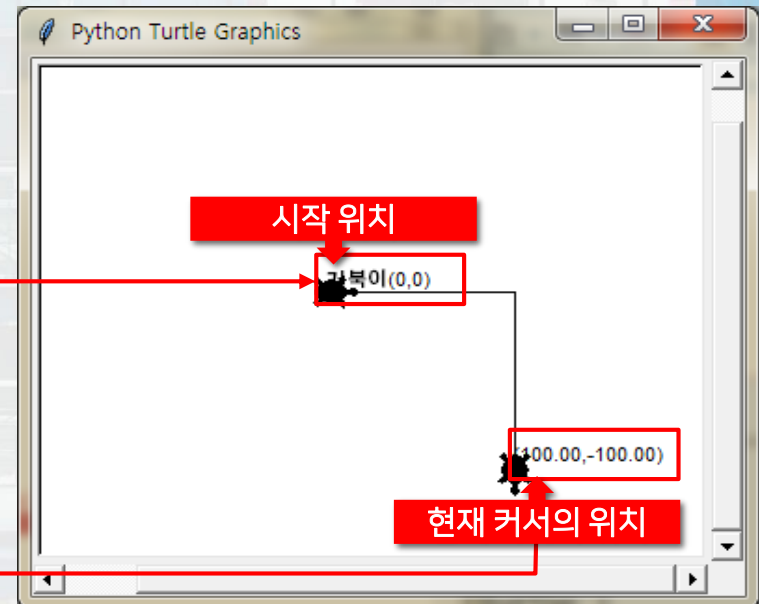
3.2 터틀 그래픽 활용 - 모양과 문자열 출력

- 커서 모양과 문자열 출력
 - stamp(): 커서 모양 출력
 - write("출력문자열"): 문자열 출력
 - position(): 현재 위치 좌표 출력

python shell

```
>>> import turtle
>>> t=turtle.Turtle()
>>> t.shape("turtle")
>>> t.stamp()
>>> t.write("거북이(0,0)")
>>> t.forward(100)
>>> t.right(90)
>>> t.forward(100)
>>> t.write(t.position())
```

[실행결과]





3.2 터틀 그래픽 활용 - 이동과 circle() 활용

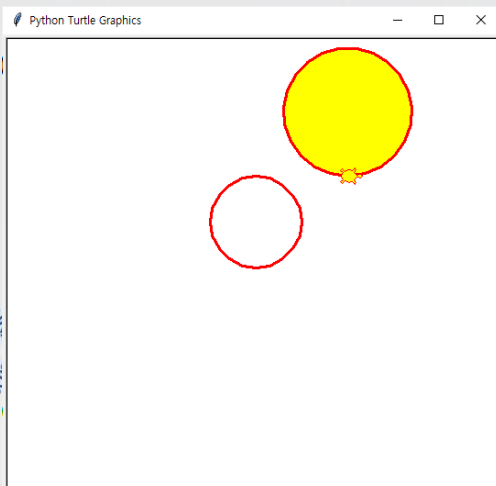
- 이동과 circle() 함수의 활용
 - goto(x,y): 지정된 위치로 이동
 - circle(반지름) : 원 도형 그리기
 - circle(반지름, 회전각도, 꼭지점갯수)
 - 다각형 도형 그리기 가능
 - hideturtle() : 커서 모양 숨기기



3.2 터틀 그래픽 활용 - 선 굵기와 색상, 도형

[소스코드] lile_color.py

- 선의 굵기와 색상, 도형 지정
 - width(선굵기): 선의 굵기
 - color("색상명"): 선의 색상 지정
 - fillcolor("색상명"): 채우기 색상 지정
 - begin_fill(): 색상 채우기 준비
 - end_fill(): 색상 채우기 완료
 - up(): 펜 들기 (=penup())
 - down(): 펜 내리기 (=pendown())
 - circle(반지름): 원 도형 그리기



[실행결과]

선의 굵기와 색상 원그리기

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
```

```
t.width(3)          # 선 굵기 지정
t.color("red")       # 선 색 지정하기
t.circle(50)         # 원 그리기
```

```
t.up()              # 펜 들기
t.goto(100,100)     # (100,100)이동
```

```
t.down()            # 펜 내리기
```

도형에 색 채우기

```
t.fillcolor("yellow") # 채우기 색상준비
t.begin_fill()        # 색 채우기 준비
t.circle(70)          # 원 그리기
t.end_fill()          # 색 채우기 완료
```



3.2 터틀 그래픽 활용 - 참고색상

color example code: `named_colors.py`

(Source code, png, pdf)

색상에 오류가 있으면 다른 색상을 사용하세요

주의사항 : 한글자 색상은
안됩니다

	black		k		dimgray		dimgrey
	gray		grey		darkgray		darkgrey
	silver		lightgray		lightgrey		gainsboro
	whitesmoke		w		white		snow
	rosybrown		lightcoral		indianred		brown
	firebrick		maroon		darkred		r
	red		mistyrose		salmon		tomato
	darksalmon		coral		orangered		lightsalmon
	sienna		seashell		chocolate		saddlebrown
	sandybrown		peachpuff		peru		linen
	bisque		darkorange		burlywood		antiquewhite
	tan		navajowhite		blanchedalmond		papayawhip
	moccasin		orange		wheat		oldlace
	floralwhite		darkgoldenrod		goldenrod		cornsilk
	gold		lemonchiffon		khaki		palegoldenrod
	darkkhaki		ivory		beige		lightyellow
	lightgoldenrodyellow		olive		y		yellow
	olivedrab		yellowgreen		darkolivegreen		greenyellow
	chartreuse		lawngreen		honeydew		darkseagreen
	palegreen		lightgreen		forestgreen		limegreen
	darkgreen		g		green		lime
	seagreen		mediumseagreen		springgreen		mintcream
	mediumspringgreen		mediumaquamarine		aquamarine		turquoise
	lightseagreen		mediumturquoise		azure		lightcyan
	paleturquoise		darkslategray		darkslategray		teal
	darkcyan		c		aqua		cyan
	darkturquoise		cadetblue		powderblue		lightblue
	deepskyblue		skyblue		lightskyblue		steelblue
	aliceblue		dodgerblue		lightslategray		lightslategray
	slategray		slategrey		lightsteelblue		cornflowerblue
	royalblue		ghostwhite		lavender		midnightblue
	navy		darkblue		mediumblue		b
	blue		slateblue		darkslateblue		mediumslateblue
	mediumpurple		rebeccapurple		blueviolet		indigo
	darkorchid		darkviolet		mediumorchid		thistle
	plum		violet		purple		darkmagenta
	m		fuchsia		magenta		orchid
	mediumvioletred		deeppink		hotpink		lavenderblush
	palevioletred		crimson		pink		lightpink



3.2 터틀 그래픽 활용 - 참고색상

https://www.w3schools.com/colors/colors_picker.asp

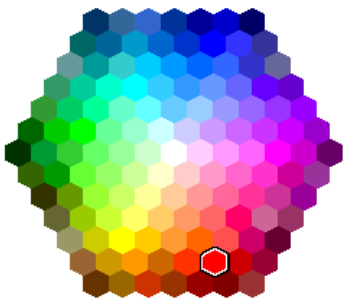
← → ↻ 🔒 w3schools.com/colors/colors_picker.asp

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP MORE ▾ REFERENCES ▾ EXERCISES ▾

Colors Tutorial

- Colors HOME
- Color Names
- Color Values
- Color Groups
- Color Shades
- Color Picker**
- Color Mixer
- Color Converter
- Color RGB
- Color HEX
- Color HSL
- Color HWB
- Color CMYK
- Color NCol
- Color Gradient
- Color Theory
- Color Wheels
- Color currentcolor
- Color Hues
- Color Schemes
- Color Palettes

Pick a Color:



Selected Color:

Black Text

Shadow

White Text

Shadow

Lighter / Darker:

100%	#ffffff
95%	#ffe6e6
90%	#ffcccc
85%	#ffb3b3
80%	#ff9999
75%	#ff8080
70%	#ff6666
65%	#ff4d4d
60%	#ff3333
55%	#ff1a1a
50%	#ff0000
45%	#e60000
30%	#990000
25%	#800000
20%	#660000
15%	#4d0000

Or Enter a Color:


Color value OK

Red

hs1(0, 100%, 50%)

색상값 : #ffcccc 형식을 넣어도 됩니다

COLOR PICKER



HOW TO

<출처> https://www.w3schools.com/colors/colors_picker.asp



3.2 터틀 그래픽 활용 - 다각형 그리기

[3장 추가소스코드] shape.py

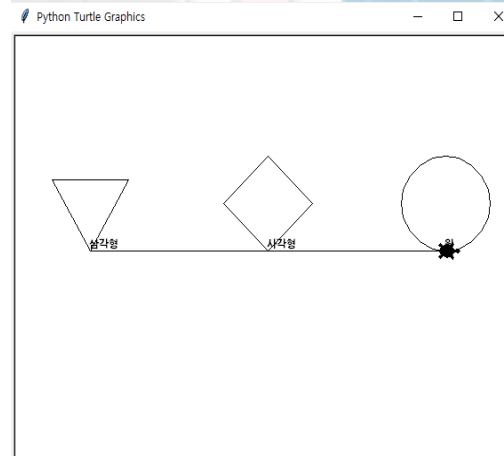
```
import turtle
t = turtle.Turtle()
t.shape("turtle")
t.up()                # 선 그리지 않고 이동하기 위해 펜 들기
t.goto(-200,0)        # x 축 -200, y축 0으로 이동

# 삼각형 그리기
t.down()              # 그림을 그리기 위해 펜 내리기
t.write("삼각형")     # 글자 출력하기
t.circle(50,360,3)    #반지름 50, 회전각도 360, 꼭지점 3개

t.goto(0,0)
# 사각형 그리기
t.write("사각형")
t.circle(50,360,4)    #반지름 50, 회전각도 360, 꼭지점 4개
t.goto(200,0)

# 원그리기
t.write("원")
t.circle(50)
t.goto(200,0)
```

[실행결과]



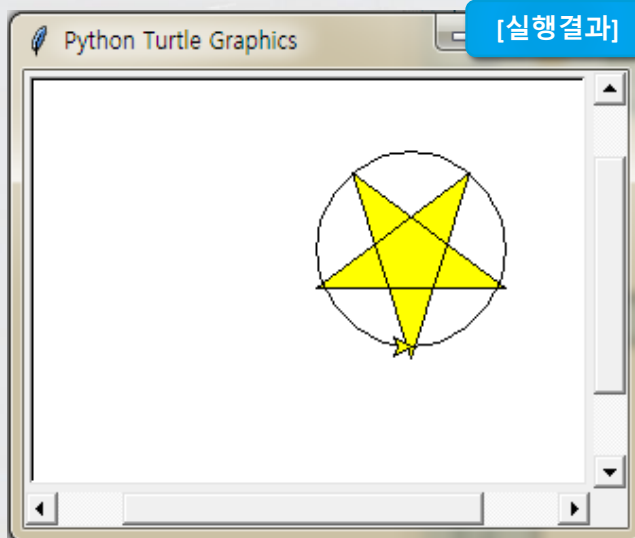


3.2 터틀 그래픽 활용 - 다양하게 표현하기

- 터틀 그래픽으로 다양하게 표현하기
 - 노란색 별을 둘러싸는 원 그리기
 - 별을 그리는 한 선의 길이 100,
 - 별을 그리는 각도 144도
 - 원의 반지름은 50
 - 원이 별을 감싸도록 적당한 위치로 이동하여 원 그리기

[소스코드] 3-1.py

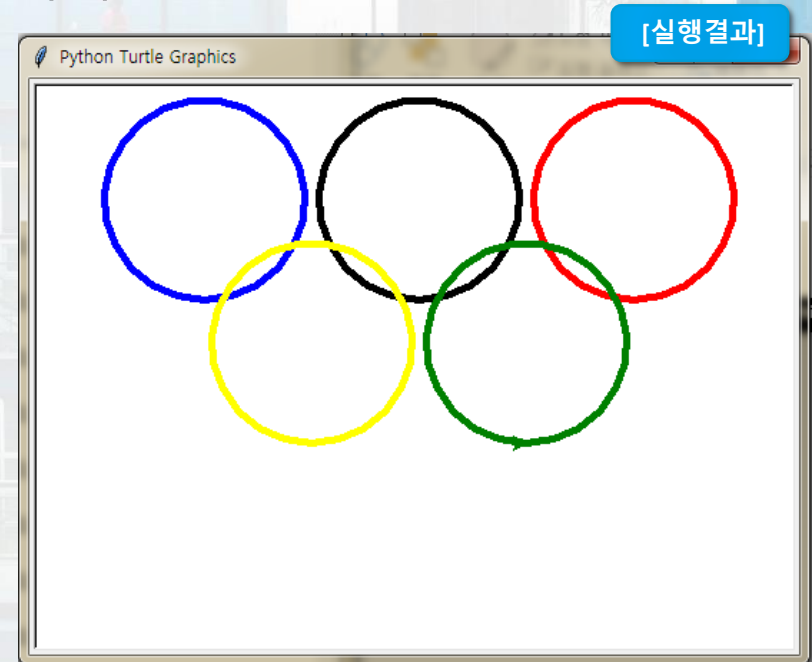
```
import turtle
t=turtle.Turtle()
t.fillcolor("yellow")
t.begin_fill()
t.forward(100)
t.left(144)
t.forward(100)
t.left(144)
t.forward(100)
t.left(144)
t.forward(100)
t.left(144)
t.forward(100)
t.left(144)
t.end_fill()
t.up()
t.goto(50, -30)
t.down()
t.circle(50)
```





3.2 터틀 그래픽 활용 - 다양하게 표현하기

- 터틀 그래픽으로 다양하게 표현하기
 - 오륜기 그리기
 - 원이 그려지는 시작 위치
 - x축 -150, y축 50
 - 이후 적당히 위치 변경하며 그리기
 - 원의 반지름은 70
 - 선의 굵기 5
 - 도형이 그려지는 속도 5
 - speed(속도) : 실행 속도





3.2 터틀 그래픽 활용 - 다양하게 표현하기

- 오른기 그리기

시작

```
import turtle
t=turtle.Turtle()
t.speed(5)
t.width(5)

t.up()
t.goto(-150,50)
t.down()
t.color("blue")
t.circle(70)

t.up()
t.goto(0,50)
t.down()
t.color("black")
t.circle(70)
```

이어서

```
t.up()
t.goto(150,50)
t.down()
t.color("red")
t.circle(70)

t.up()
t.goto(-75,-50)
t.down()
t.color("yellow")
t.circle(70)

t.up()
t.goto(75,-50)
t.down()
t.color("green")
t.circle(70)
```

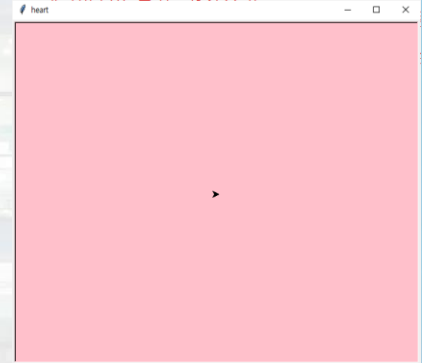
[소스코드] 3-2.py 책65페이지



3.2 터틀 그래픽 활용 - 배경색넣기

[실행결과]

- Screen() 객체
screen = turtle.Screen()
turtle 모듈의 screen() 객체를 통해
전체 화면을 제어하는 객체를 생성



Screen() 객체는 화면의 크기, 배경색, 키보드 입출력 이벤트등을 관리

[3장 추가소스파일] bg.py

```
import turtle          # turtle 모듈 가져오기
t = turtle.Turtle()     # turtle 모듈이 제공하는 Turtle()를 이용하여 t 객체생성

s = turtle.Screen()     # Screen()을 이용하여 s 스크린 객체 생성
s.title("heart")        # 스크린 제목 이름 지정
s.bgcolor("pink")       # 스크린 배경색 지정
```



3.2 터틀 그래픽 활용 - 다양하게 표현하기

<https://docs.python.org/3/library/turtle.html>

turtle — Turtle graphics — Pyth x +

← → ↺ docs.python.org/3/library/turtle.html

Python » English 3.8.2 Documentation » The Python Standard Library » Program Frameworks » Quick search Go | previous | next | modules | index

Table of Contents

- turtle — Turtle graphics
 - Introduction
 - Overview of available Turtle and Screen methods
 - Turtle methods
 - Methods of TurtleScreen/Screen
 - Methods of RawTurtle/Turtle and corresponding functions
 - Turtle motion
 - Tell Turtle's state
 - Settings for measurement
 - Pen control
 - Drawing state
 - Color control
 - Filling
 - More drawing control
 - Turtle state
 - Visibility
 - Appearance
 - Using events
 - Special Turtle methods
 - Compound shapes
 - Methods of TurtleScreen/Screen and corresponding functions

turtle — Turtle graphics

Source code: [Lib/turtle.py](#)

터틀 그래픽 관련 함수 제공 사이트

Introduction

Turtle graphics is a popular way for introducing programming to kids. It was part of the original Logo programming language developed by Wally Feurzeig, Seymour Papert and Cynthia Solomon in 1967.

Imagine a robotic turtle starting at (0, 0) in the x-y plane. After an `import turtle`, give it the command `turtle.forward(15)`, and it moves (on-screen!) 15 pixels in the direction it is facing, drawing a line as it moves. Give it the command `turtle.right(25)`, and it rotates in-place 25 degrees clockwise.


By combining together these and similar commands, intricate shapes and pictures can easily be drawn.

The `turtle` module is an extended reimplementaion of the same-named module from the Python standard distribution up to version Python 2.5.

It tries to keep the merits of the old turtle module and to be (nearly) 100% compatible with it. This means in the first place to enable the learning programmer to use all the commands, classes and methods interactively when using the module from within IDLE run with the `-n` switch.

Turtle star

Turtle can draw intricate shapes using programs that repeat simple moves.





마무리

- 이번 시간에는 터틀 그래픽 환경과 터틀 모듈의 개념을 이해하고
- 터틀 모듈에서 제공하는 함수의 의미와 작성규칙에 대해 학습하였습니다.