



## 18장. 그래픽

파이썬 정보



# Contents

## ❖ 목차

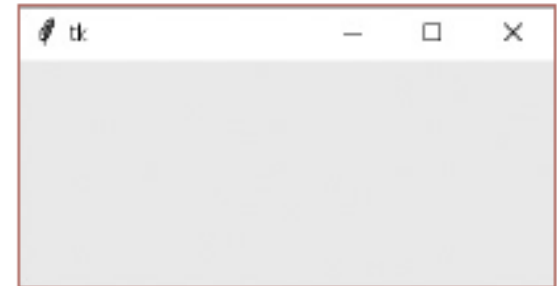
- 1. Tkinter
- 2. 터틀 그래픽



# 1. Tkinter

## ❖ 윈도우 생성

- **Tcl/Tk**, 그 하위 모듈인 **터틀로** 파이썬에서 그래픽 지원
- **Tcl** (Tool Command Language)
  - 스크립트 언어 일종
- **Tk**
  - GUI 툴킷으로 크로스 플랫폼 지원
  - 타이틀 바, 경계선, 출력 표면 가진 메인 윈도우 생성
- **Tkinter**
  - Tcl/Tk를 파이썬에서 사용할 수 있도록 인터페이스하는 모듈
  - 파이썬에 내장



# 1. Tkinter

## ❖ 위젯 배치

- 메인 윈도우 안에 위젯 배치하고 이벤트 처리하여 프로그램 형성
- Tk() 함수 호출문과 mainloop 호출문 사이 위젯 생성

widget

```
from tkinter import *
main = Tk()
main.title("Tk Test")
main.geometry("300x200")

lbl = Label(main, text="Label", font="Arial 20")
lbl.pack()
apple = Button(main, text="Apple", foreground="Red")
apple.pack()
orange = Button(main, text="Orange", foreground="Green")
orange.pack()

main.mainloop()
```



# 1. Tkinter

- 가로, 세로 변경 금지하여 윈도우 크기 고정

```
main.resizable(False, False)
```

- 메인 윈도우 내 차일드 위젯 배치하여 사용자 인터페이스 디자인
  - 레이블 생성 및 폰트 등 옵션 지정

```
lbl = Label(main)
lbl["text"]="Label"
lbl["font"]="Arial 20"
```

- 위젯 생성 후 pack 메서드 호출하여 내용에 맞게 크기 계산 및 속성 조정



# 1. Tkinter

## ❖ 대화상자

### ■ tkinter.messagebox 모듈

- 실행 중 메시지 박스 통해 정보 보이거나 질문하는 기능 제공

```
showinfo(title, message, option)
showwarning(title, message, option)
showerror(title, message, option)
askyesno(title, message, option)
askyesnocancel(title, message, option)
askokcancel(title, message, option)
askretrycancel(title, message, option)
askquestion(title, message, option)
```



# 1. Tkinter

## messagebox

```
from tkinter import *
import tkinter.messagebox

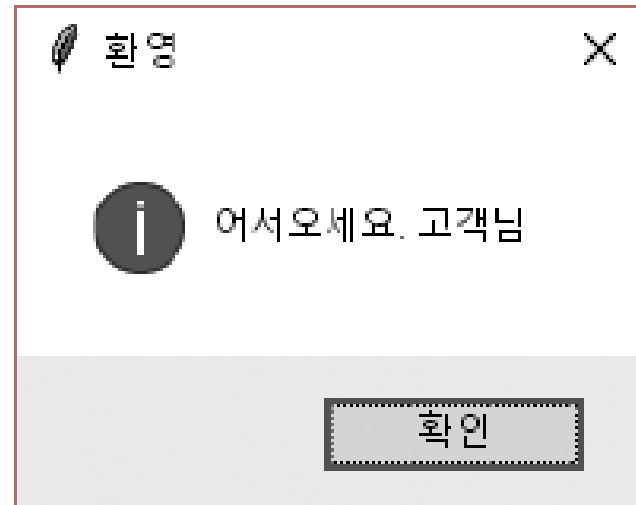
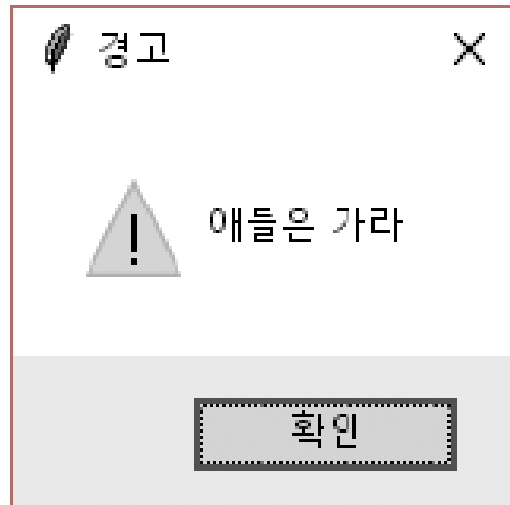
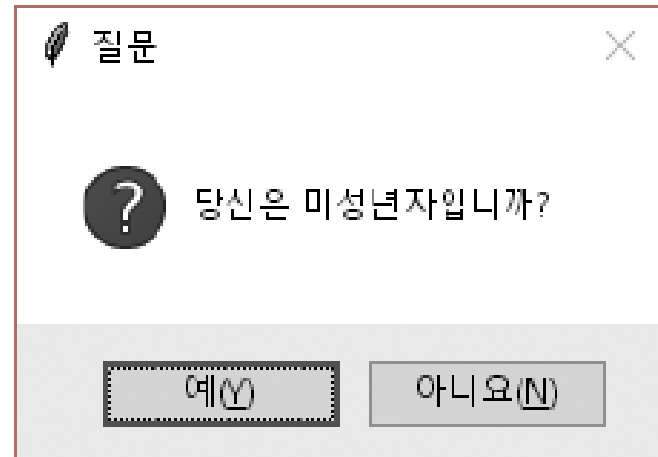
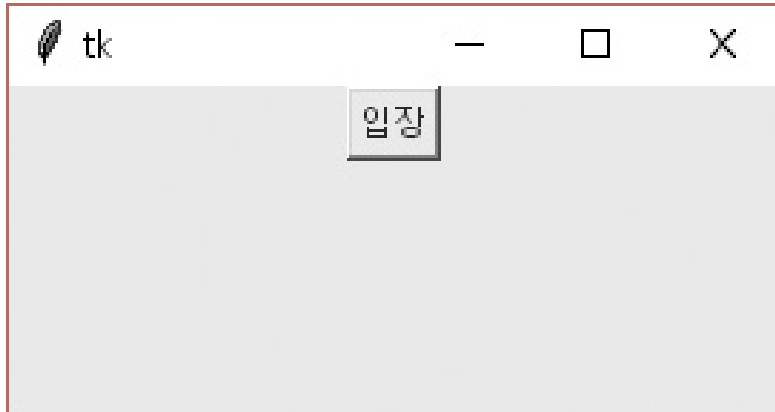
main = Tk()

def btnclick():
    if tkinter.messagebox.askyesno("질문", "당신은 미성년자입니까?"):
        tkinter.messagebox.showwarning("경고", "애들은 가라")
    else:
        tkinter.messagebox.showinfo("환영", "어서오세요. 고객님")
btn = Button(main, text="입장", foreground="Blue", command = btnclick)
btn.pack()

main.mainloop()
```



# 1. Tkinter

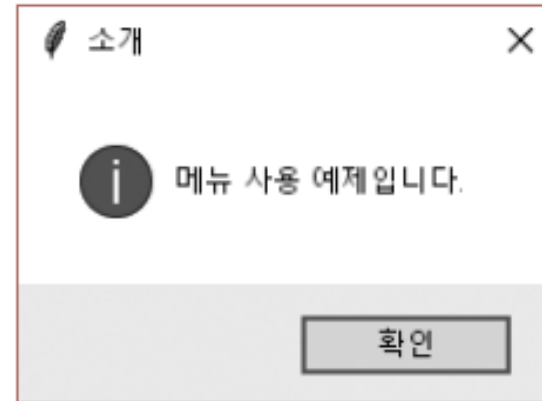
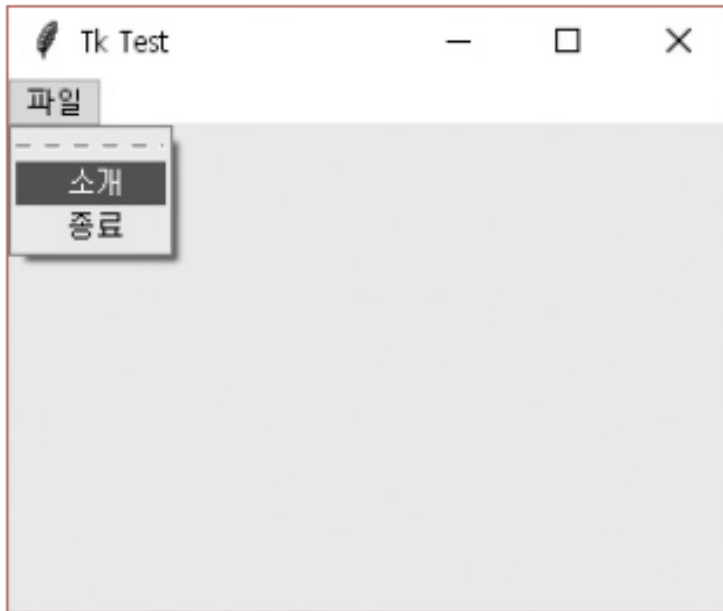




# 1. Tkinter

## ❖ 메뉴

- 프로그램의 모든 기능을 집약적으로 보여주는 인터페이스
- Menu 클래스로 생성
- 메뉴 구성하는 각 객체 생성하여 구조에 따라 연결



# 1. Tkinter



tkmenu

```
from tkinter import *
import tkinter.messagebox

main = Tk()
main.title("Tk Test")
main.geometry("300x200")

menubar = Menu(main)
main.config(menu = menubar)

popup = Menu(menubar)
menubar.add_cascade(label = "파일", menu = popup)

def about():
    tkinter.messagebox.showinfo("소개", "메뉴 사용 예제입니다.")

popup.add_command(label = "소개", command = about)
popup.add_command(label = "종료", command = quit)

main.mainloop()
```



# 1. Tkinter

## ❖ 그래픽 출력

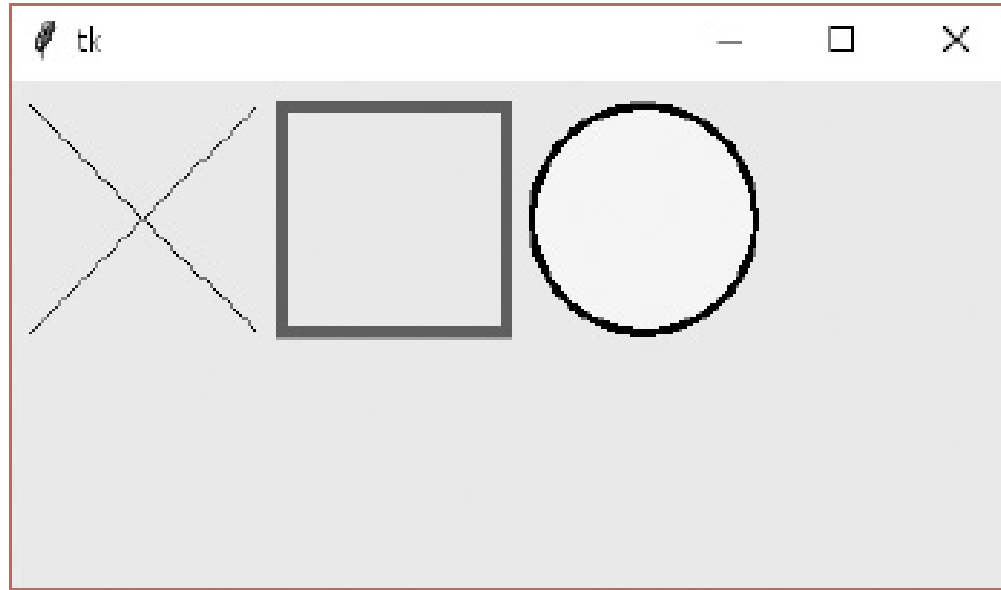
- Tk 통해 그래픽 그릴 수 있음
- 캔버스 위젯 및 그 메서드로 도형 배치

canvas

```
from tkinter import *  
main = Tk()  
c = Canvas(main, width=400, height=200)  
c.pack()  
  
c.create_line(10, 10, 100, 100)  
c.create_line(10, 100, 100, 10, fill="blue")  
c.create_rectangle(110, 10, 200, 100, outline="red", width=5)  
c.create_oval(210, 10, 300, 100, width=3, fill="yellow")  
  
main.mainloop()
```



# 1. Tkinter



- 캔버스에 배치되는 도형 및 이미지는 각각 독립된 개체
- 임의의 그래픽 직접 그리려면 터틀 그래픽 사용




## 2. 터틀 그래픽

### ❖ 터틀 그래픽

- Logo 언어의 그래픽 라이브러리
- Tkinter의 서브 모듈

```
import turtle as t  
t.shape("turtle")
```

	classic	turtle	triangle	arrow	square	circle
모양						

- 거북이를 원하는 방향으로 움직여 캔버스에 그림 그릴 수 있음

메서드	단축명	설명
forward(픽셀)	fd	앞으로 전진
backward(픽셀)	back	뒤로 후진
left(각도)	lt	왼쪽으로 회전
right(각도)	rt	오른쪽으로 회전

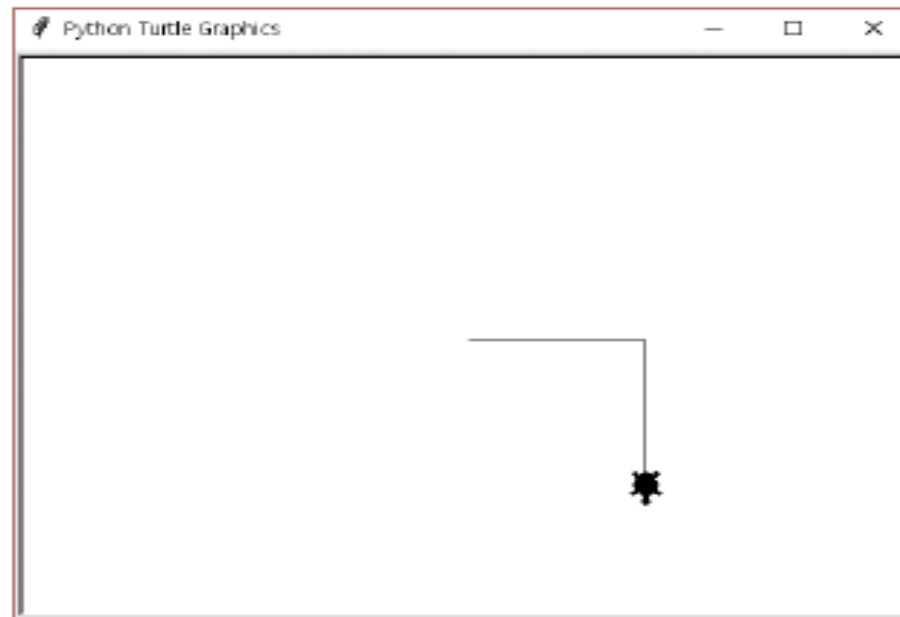


## 2. 터틀 그래픽

turtle

```
import turtle as t
t.shape("turtle")

t.forward(100)
t.right(90)
t.forward(100)
t.done()
```



## 2. 터틀 그래픽

### ❖ 거북이 조정 메서드

- 거북이의 이동 속도, 동작 등 조정

메서드	단축명	설명
pendown	down	꼬리를 내림(그리기 시작)
penup	up	꼬리를 올림(그리지 않고 이동)
speed(속도)		거북이 이동속도 변경 1이 느리고 10이 빠르며 0이 가장 빠르다. 인수가 없으면 현재 속도를 조사하여 리턴한다
showturtle	st	거북이를 표시한다
hideturtle	ht	거북이를 숨긴다.
stamp		현재 위치에 거북이 모양의 도장을 찍는다. 도장의 id를 리턴하며 이후 개별적으로 또는 전부 삭제할 수 있다
clear		화면을 지운다
reset		화면을 지우고 거북이를 중앙에 배치한다

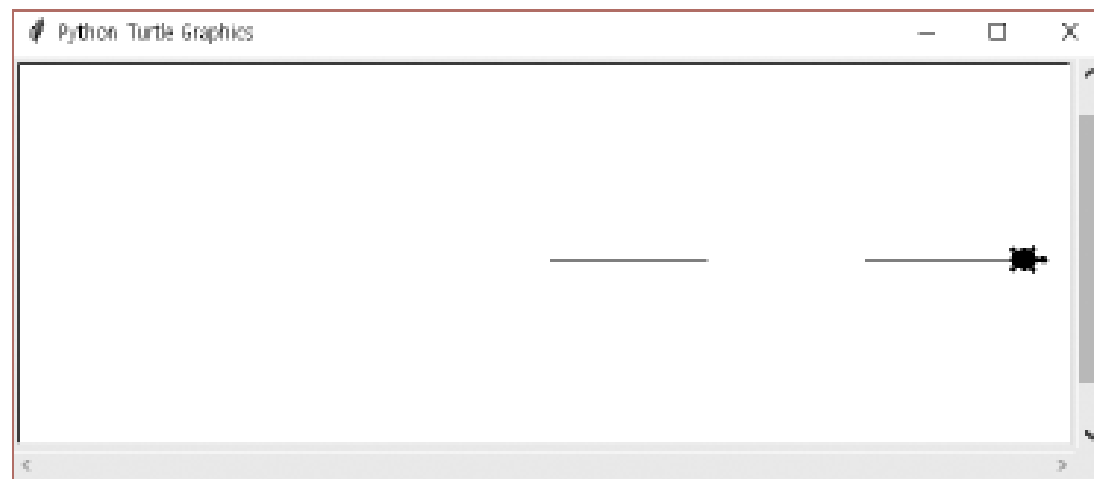


## 2. 터틀 그래픽

updown

```
import turtle as t
t.shape("turtle")

t.speed(1)
t.forward(100)
t.up()
t.forward(100)
t.down()
t.forward(100)
t.done()
```





## 2. 터틀 그래픽

메서드	단축명	설명
circle(반지름, 각도, 조각)		원을 그린다. 각도를 지정하면 원호를 그리며 조각을 지정하면 다각형을 그린다.
pensize(굵기)	width	펜의 굵기를 바꾼다.
pencolor(색)		펜의 색상을 바꾼다. 색상 이름 또는 r, g, b 인수로 지정한다.
bgcolor(색)		배경색을 바꾼다.
fillcolor(색)		채우기 색을 변경한다.
color(펜, 채우기)		펜의 색과 채우기 색을 한꺼번에 지정한다.
begin_fill()		칠할 준비를 한다.
end_fill()		도형 내부를 칠한다.



## 2. 터틀 그래픽

메서드	단축명	설명
<code>xcor(), ycor()</code>		거북이의 현재 좌표를 구한다. $x$ , $y$ 각각을 구할 수도 있고 두 값을 한꺼번에 구할 때는 <code>pos</code> 함수를 사용한다.
<code>setpos(x, y)</code>	<code>goto</code>	거북이의 좌표를 옮긴다.
<code>distance(x, y)</code>		현재 좌표에서 $(x, y)$ 까지의 거리를 구한다.
<code>heading</code>		거북이의 각도를 구한다.
<code>towards(x, y)</code>		$(x, y)$ 좌표 쪽으로 각도를 변경한다.
<code>setheading(각도)</code>	<code>seth</code>	거북이의 각도를 변경한다. 오른쪽이 0도이며 위쪽이 90도이다.
<code>home</code>		거북이를 화면 중앙, 오른쪽 방향으로 옮긴다.
<code>write(문자열)</code>		문자열을 출력한다. 문자열, 이동 여부, 정렬 폰트, 크기 등의 인수를 전달한다.



## 2. 터틀 그래픽

### ❖ 이벤트 처리

- 터틀 그래픽은 Tkinter의 하위 모듈
- GUI 환경에서 일반적인 대부분 이벤트 처리 가능

메서드	설명
onkeypress(함수, "키")	키를 누르면 지정한 함수를 호출한다.
onkeyrelease(함수, "키")	키를 놓으면 지정한 함수를 호출한다.
onscreenclick(함수, 버튼)	마우스 버튼을 누르면 지정한 함수를 호출한다. 버튼 생략 시 1번인 왼쪽 버튼이 적용된다.
ontimer(함수, 시간)	일정 시간 후에 함수를 호출한다. 시간은 1/1000초 단위이다. 반복적으로 호출하려면 타이머를 계속 설정한다.
listen()	키 입력 포커스를 준다.



## 2. 터틀 그래픽

freeline

```
import turtle as t

def draw(head,dist):
    t.setheading(head)
    t.forward(dist)

def toleft():
    draw(180, 15)

def toright():
    draw(0, 15)

def toup():
    draw(90, 15)

def todown():
    draw(270, 15)

def move(x,y):
```



## 2. 터틀 그래픽

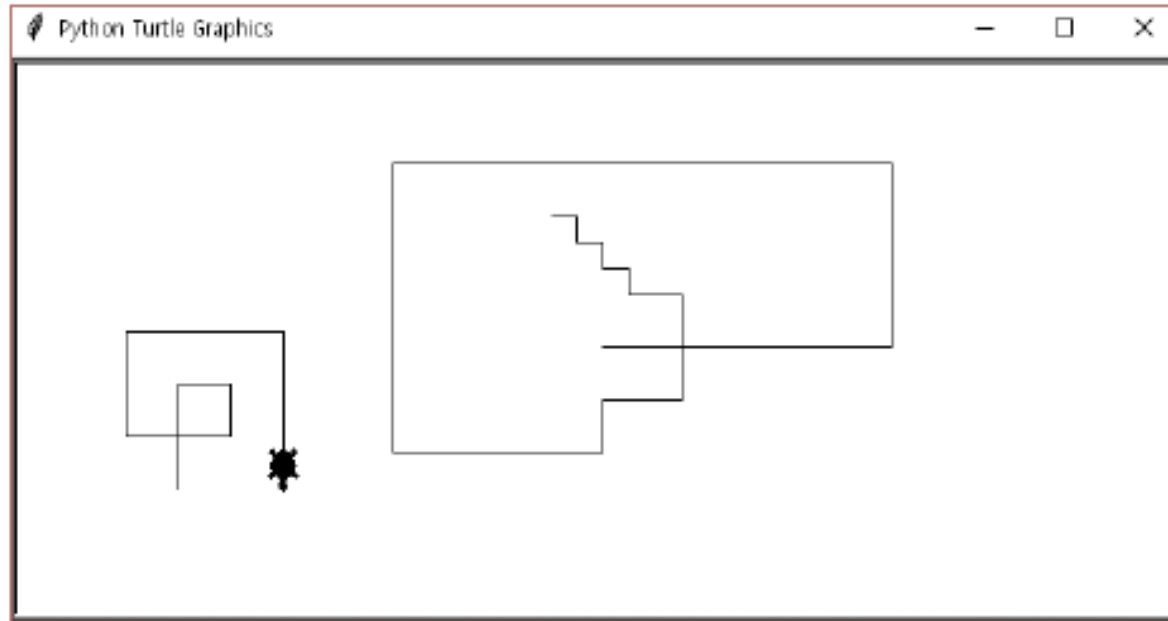


```
t.up()  
t.setpos(x,y)  
t.down()
```

```
t.shape("turtle")  
t.speed(0)  
t.onkeypress(toleft, "Left")  
t.onkeypress(toright, "Right")  
t.onkeypress(toup, "Up")  
t.onkeypress(todown, "Down")  
t.onscreenclick(move)  
t.listen()  
t.done()
```



## 2. 터틀 그래픽

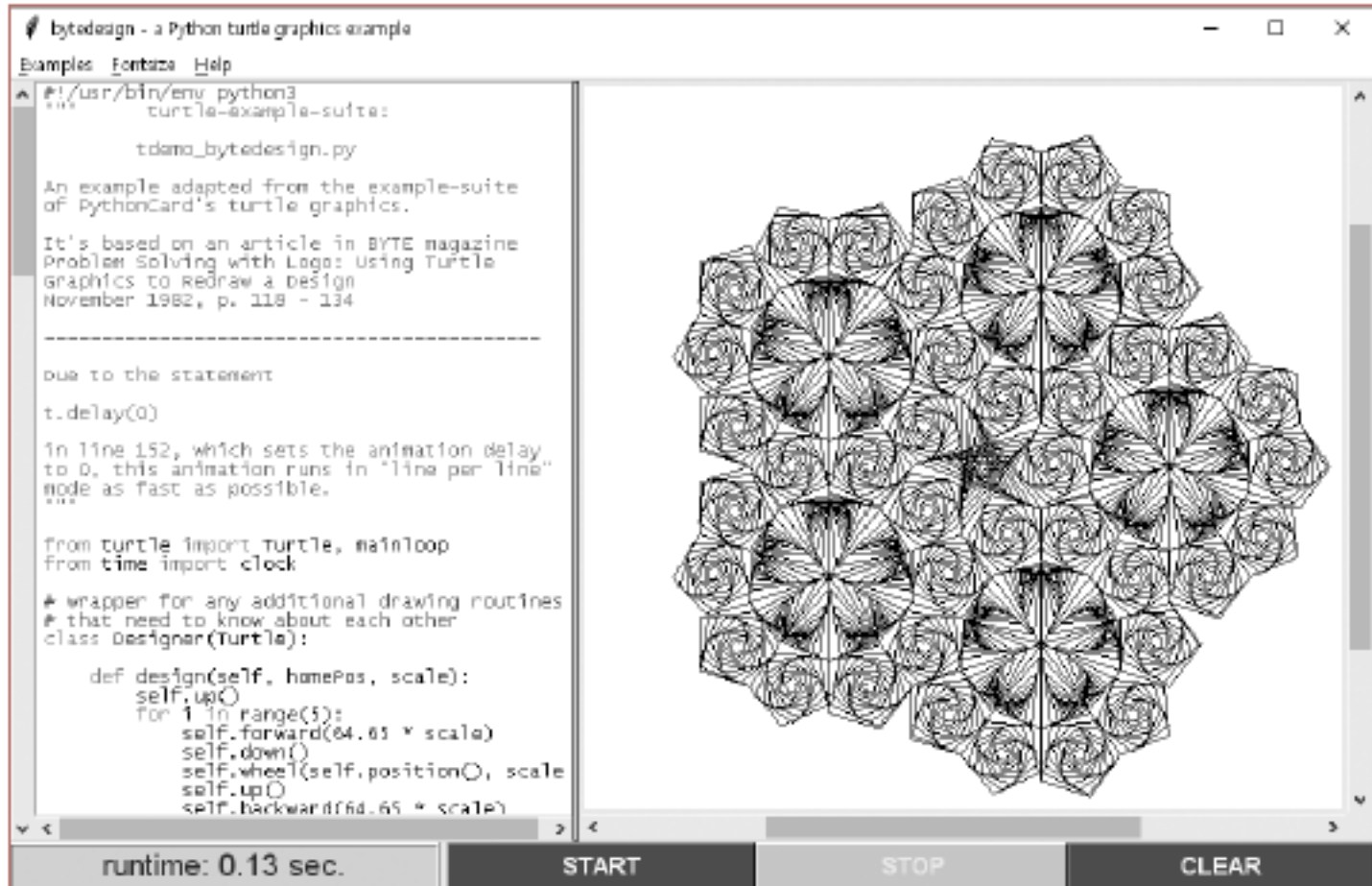


- 이동 스피드 0 설정하여 지연 없이 이동
- 상하좌우 커서 이동에 대해 각기 15픽셀만큼 이동
- 키보드 ← 키 눌러 toleft 함수 호출
  - draw 함수 호출하여 거북이를 왼쪽으로 15픽셀 이동
- 마우스 왼쪽 버튼으로 move 함수 호출
- listen 함수 호출하여 캔버스에 포커스 줌



## 2. 터틀 그래픽

- 보다 복잡한 그림까지 가능





# Thank You !

파이썬 정복

