



▶ 학습내용

01. 모듈과 라이브러리를 학습합니다.
02. tkinter 모듈을 사용하는 방법을 학습합니다.
03. 필로우(Pillow) 라이브러리를 사용하는 방법을 학습합니다.
04. 맷플롯립(Matplotlib) 라이브러리를 사용하는 방법을 학습합니다

01. 모듈(module)



- '어떤 것의 일부', '부품'
- 특정 기능을 하는 컴퓨터 시스템
- 기능별 단위로 분할한 것
- 데이터, 함수, 클래스 등을 모아서 파일로 저장해 놓은 것

01. 모듈(module)



import를 이용하여 모듈을 가져옵니다.

코드

```
import math  
result = math.gcd(255, 300)  
print(result)
```

실행 결과

15

01. 모듈(module)



모듈 이름 없이 함수 이름만 쓰고 싶은 경우 from 키워드를 이용합니다.

코드

```
from math import *  
result = gcd(255, 300)  
print(result)
```

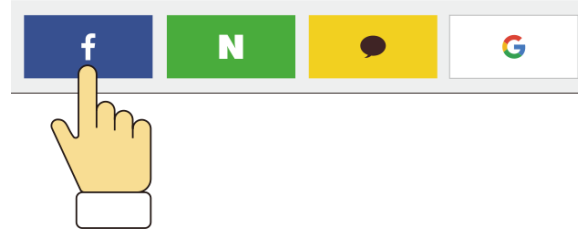
실행 결과

15

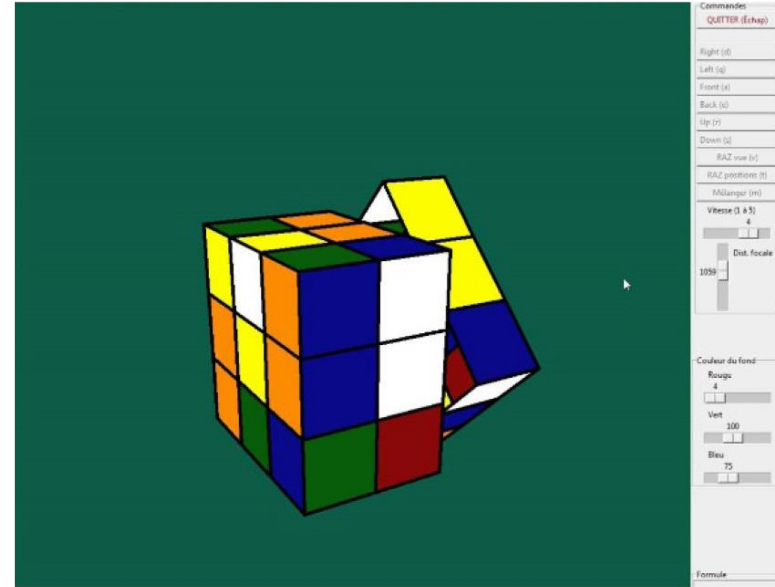
02. tkinter란?



- 그래픽 사용자 인터페이스
(GUI : Graphical User Interface)



사용자 인터페이스는
사용자가 내리는 명령을
컴퓨터가 알아듣도록
바꿔 전달하는 기능입니다.



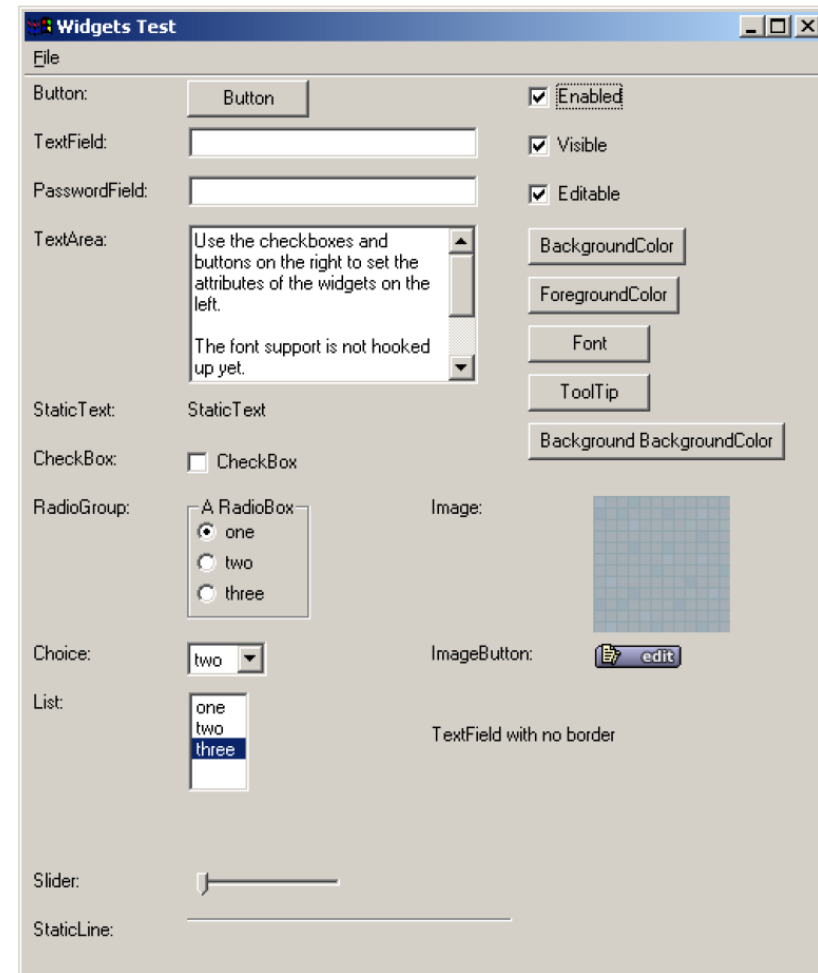
02. tkinter란?



- TK interface
- 파이썬을 설치할 때 기본으로 포함되는 그래픽 모듈
- 윈도우를 생성하고 위젯을 이용하여 사용자와 상호작용하는 프로그램을 작성할 수 있도록 함.
- 기본적인 도형을 빠르게 그릴 수 있음

03. tkinter의 위젯들

- 위젯(widget) : tkinter에서 제공하는 조그마한 도구들(버튼, 텍스트 상자 등)



(이미지 출처: pythoncard.sourceforge.net)

03. tkinter의 위젯들



위젯	설명
Button	간단한 버튼으로 명령을 수행할 때 사용합니다.
Canvas	화면에 무엇인가를 그릴 때 사용합니다.
Checkbutton	2가지의 구별되는 값을 가지는 변수를 표현합니다.
Entry	한 줄의 텍스트를 입력받는 필드입니다.
Frame	컨테이너 클래스입니다. 프레임은 경계선과 배경을 가지고 있습니다. 다른 위젯들을 그룹핑(묶기)하는 데 사용됩니다.
Label	텍스트나 이미지를 표시합니다.
Listbox	선택 사항을 표시합니다.
Menu	메뉴를 표시합니다. 풀다운 메뉴나 팝업 메뉴가 가능합니다.
Menubutton	메뉴 버튼입니다. 풀다운 메뉴가 가능합니다.

03. tkinter의 위젯들



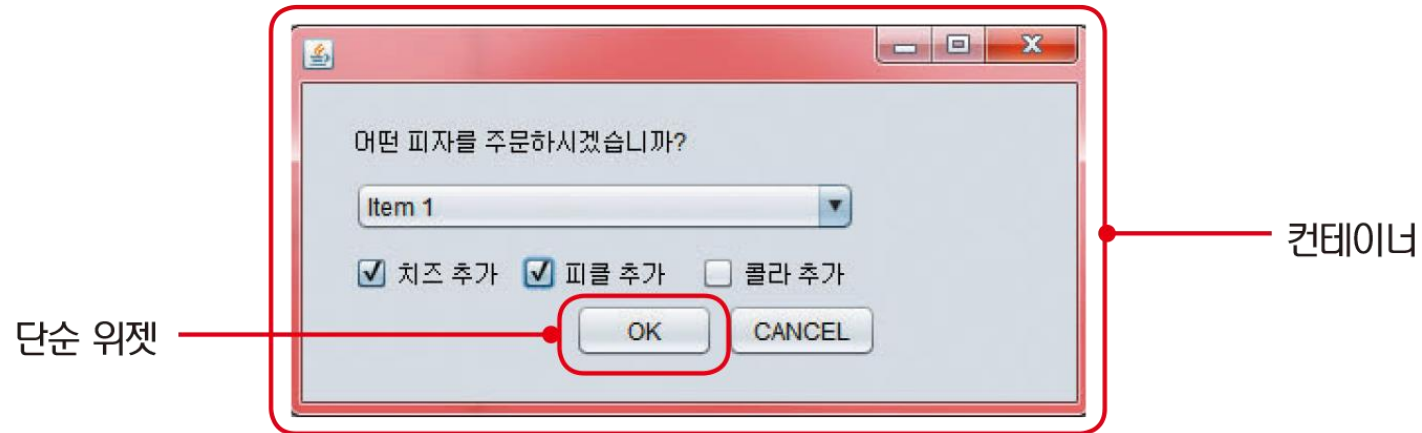
위젯	설명
Message	텍스트를 표시합니다. Label 위젯과 비슷합니다. 하지만 자동적으로 주어진 크기로 텍스트를 축소할 수 있습니다.
Radiobutton	여러 값을 가질 수 있는 변수를 표시합니다.
Scale	슬라이더를 끌어서 값을 입력하는 데 사용합니다.
Scrollbar	캔버스, 엔트리, 리스트 박스, 텍스트 위젯을 위한 스크롤바를 제공합니다.
Text	형식을 가지는 텍스트를 표시합니다. 여러 가지 스타일과 속성으로 텍스트를 표시할 수 있습니다.
Toplevel	최상위 윈도우로 표시되는 독립적인 컨테이너 위젯입니다.
LabelFrame	경계선과 제목을 가지는 프레임 위젯의 변형입니다.
PanedWindow	자식 위젯들을 크기조절이 가능한 패널로 관리하는 컨테이너 위젯입니다.
Spinbox	특정한 범위에서 값을 선택하는 엔트리 위젯의 변형입니다.

03. tkinter의 위젯들



- 단순 위젯 : Button, Canvas, Checkbutton, Entry, Label, Message 등
- 컨테이너 위젯 : Frame, Toplevel, LavelFrame, PanedWindow 등

.....



04. [tkinter] 윈도우와 버튼을 만들어 봅시다

12

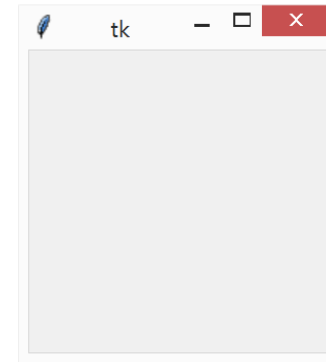


tkinter를 사용하여 간단히 빈 윈도우 창을 하나 띄어 보겠습니다.

코드

```
from tkinter import *  
  
window=Tk()  
  
# 이 부분에 화면을 구성할 요소들을 작성합니다.  
  
window.mainloop()
```

실행 결과



04. [tkinter] 윈도우와 버튼을 만들어 봅시다

13

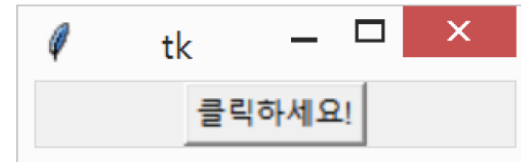


윈도우에 버튼을 추가해 보겠습니다.

코드

```
from tkinter import *  
  
window=Tk()  
  
button = Button(window, text = "클릭하세요!")  
button.pack()  
  
window.mainloop()
```

실행 결과



05. [tkinter] 윈도우 배치 관리자



- 위젯들은 컨테이너 내부에 배치됨 : 프로그래머가 하나하나 구체적으로 지정 가능
- 윈도우 크기가 바뀌거나 사용자 컴퓨터 환경이 다른 경우 위젯 배치 설정 어려움
- 위젯 배치 관리자(Layout manager) : 자동으로 컨테이너 안의 위젯의 크기와 위치 관리
- 종류
 - ✓ Pack : 압축 배치 관리자
 - ✓ Place : 절대 배치 관리자 또는 absolute
 - ✓ Grid : 격자 배치 관리자

05. [tkinter] 윈도우 배치 관리자

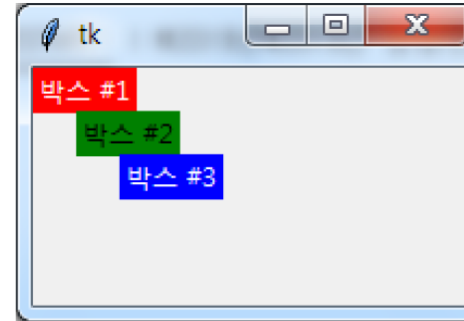


절대 위치 배치 관리자(place geometry manager)를 살펴보겠습니다.

코드

```
from tkinter import *  
window = Tk()  
  
w = Label(window, text="박스 #1", bg="red",  
fg="white")  
w.place(x=0, y=0)  
w = Label(window, text="박스 #2", bg="green",  
fg="black")  
w.place(x=20, y=20)  
w = Label(window, text="박스 #3", bg="blue",  
fg="white")  
w.place(x=40, y=40)  
  
window.mainloop()
```

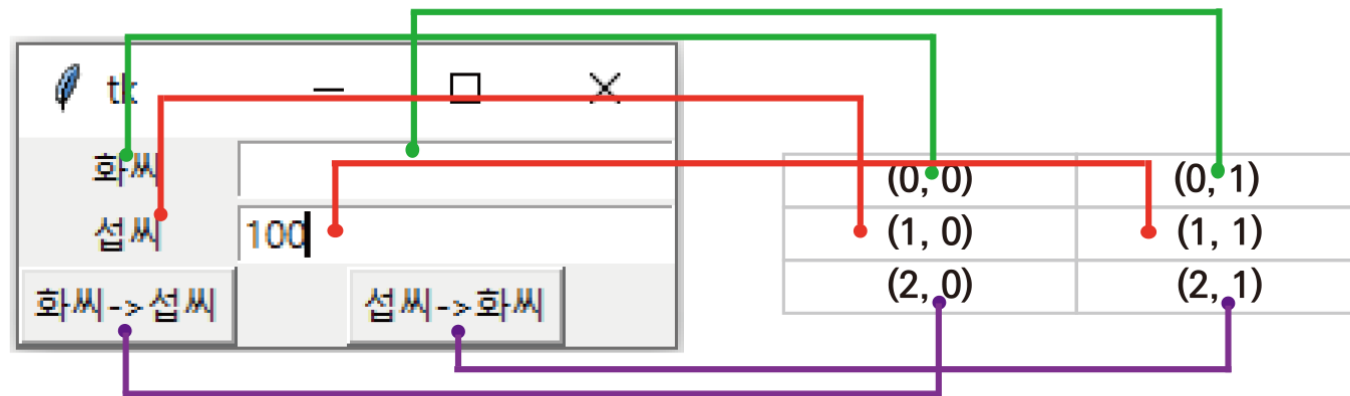
실행 결과



05. [tkinter] 윈도우 배치 관리자



격자 배치 관리자(grid geometry manager)를 살펴보겠습니다.



05. [tkinter] 윈도우 배치 관리자



코드

```
from tkinter import *

window = Tk()

l1 = Label(window, text="화씨")
l2 = Label(window, text="섭씨")
l1.grid(row=0, column=0)
l2.grid(row=1, column=0)

e1 = Entry(window)
e2 = Entry(window)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

b1 = Button(window, text="화씨->섭씨")
b2 = Button(window, text="섭씨->화씨")
b1.grid(row=2, column=0)
b2.grid(row=2, column=1)

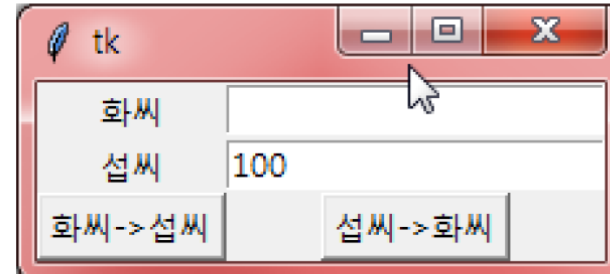
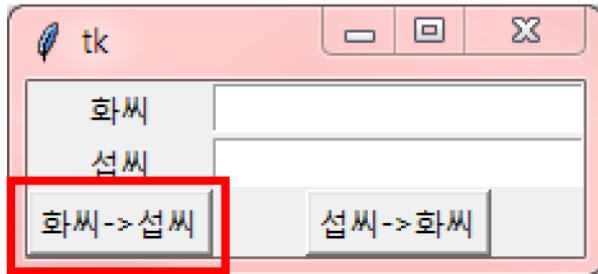
window.mainloop()
```


06. [tkinter] 버튼 이벤트 처리 #1



화씨->섭씨

버튼을 클릭하여 섭씨의 엔트리에 '100'이 보여지는 이벤트 처리를 해보겠습니다.



06. [tkinter] 버튼 이벤트 처리 #1



버튼을 클릭할 때마다 파이썬 셸에 “안녕하세요?” 텍스트가 출력됩니다.

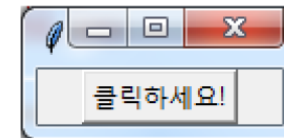
코드

```
from tkinter import *  
  
def process():  
    print("안녕하세요!")  
  
window = Tk()  
  
button = Button(window, text="클릭하세요!", command=process)  
button.pack()  
  
window.mainloop()
```

실행 결과

안녕하세요?

윈도우 창



06. [tkinter] 버튼 이벤트 처리 #1



코드

```
from tkinter import *
```

```
def process():  
    e2.insert(0, "100")
```

```
window = Tk()
```

```
l1 = Label(window, text="화씨")
```

```
l2 = Label(window, text="섭씨")
```

```
l1.grid(row=0, column=0)
```

```
l2.grid(row=1, column=0)
```

06. [tkinter] 버튼 이벤트 처리 #1



```
e1 = Entry(window)
e2 = Entry(window)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

b1 = Button(window, text="화씨->섭씨", command=process)
b2 = Button(window, text="섭씨->화씨")
b1.grid(row=2, column=0)
b2.grid(row=2, column=1)

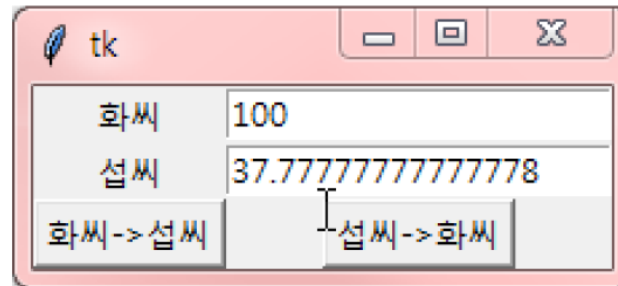
window.mainloop()
```

07. [tkinter] 버튼 이벤트 처리 #2



화씨->섭씨

버튼을 클릭하면 화씨 엔트리에 입력된 화씨온도가 섭씨온도로 계산되어 섭씨 엔트리에 출력되도록 하는 이벤트 처리를 해 보겠습니다.



07. [tkinter] 버튼 이벤트 처리 #2



코드

```
from tkinter import *

def process():
    temperature = float(e1.get())
    mytemp = (temperature-32)*5/9
    e2.insert(0, str(mytemp))

window = Tk()

l1 = Label(window, text="화씨")
l2 = Label(window, text="섭씨")
l1.grid(row=0, column=0)
l2.grid(row=1, column=0)
```

07. [tkinter] 버튼 이벤트 처리 #2



```
e1 = Entry(window)
e2 = Entry(window)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

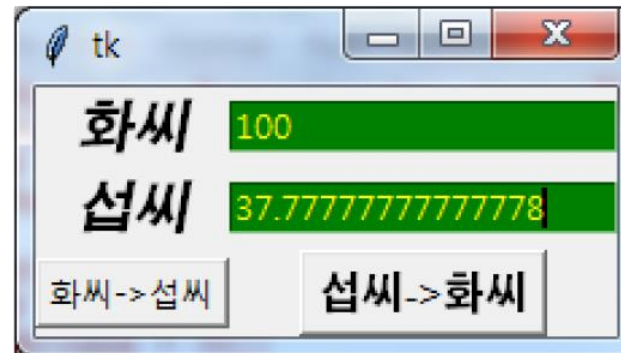
b1 = Button(window, text="화씨->섭씨", command=process)
b2 = Button(window, text="섭씨->화씨")
b1.grid(row=2, column=0)
b2.grid(row=2, column=1)

window.mainloop()
```

08. [tkinter] 위젯의 색상과 폰트 변경하기



우리가 작성한 온도 변환기의 색상과 폰트를 약간 변경해서 좀 더 알록달록 꾸며보겠습니다.



08. [tkinter] 위젯의 색상과 폰트 변경하기



코드

```
from tkinter import *

def process():
    temperature = float(e1.get())
    mytemp = (temperature-32)*5/9
    e2.insert(0, str(mytemp))

window = Tk()

l1 = Label(window, text="화씨", font='helvetica 16 italic')
l2 = Label(window, text="섭씨", font='helvetica 16 italic')
l1.grid(row=0, column=0)
l2.grid(row=1, column=0)
```

08. [tkinter] 위젯의 색상과 폰트 변경하기



```
e1 = Entry(window, bg="green", fg="yellow")
e2 = Entry(window, bg="green", fg="yellow")
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

b1 = Button(window, text="화씨->섭씨", command=process)
b2 = Button(window, text="섭씨->화씨", font='helvetica 12')
b1.grid(row=2, column=0)
b2.grid(row=2, column=1)

window.mainloop()
```

09. [tkinter] 윈도우 창 메뉴 만들기



윈도우 창의 메뉴를 생성하는 방법을 살펴보겠습니다.

코드

```
import tkinter as tk

def open():
    pass

def quit():
    window.quit()

window = tk.Tk()      # 윈도우를 생성합니다.

menubar = tk.Menu(window)

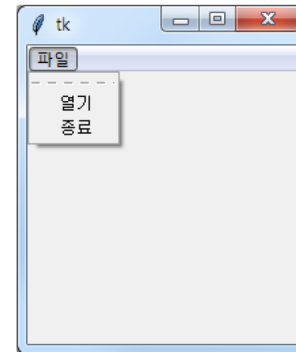
filemenu = tk.Menu(menubar)

filemenu.add_command(label="열기",
                      command=open)
filemenu.add_command(label="종료",
                      command=quit)

menubar.add_cascade(label="파일",
                    menu=filemenu)

window.config(menu=menubar)
window.mainloop()
```

실행 결과



10. [tkinter] MyPaint 프로그램



마우스만으로 엄청난 걸작을 그리는 마우스 그림의 천재 화가 두근이는 윈도우의 그림판과 같은 MyPaint 프로그램을 작성하려고 합니다.

코드

```
from tkinter import *  
  
def paint(event):  
    x1, y1 = ( event.x-1 ), ( event.y+1 )  
    x2, y2 = ( event.x-1 ), ( event.y+1 )  
    canvas.create_oval( x1, y1, x2, y2)  
  
window = Tk()  
canvas = Canvas(window)  
canvas.pack()  
canvas.bind("<B1-Motion>", paint)  
window.mainloop()
```

실행 결과



11. 라이브러리



- PIL(Python Imaging Library) 또는 필로우(Pillow) : 영상처리 라이브러리
- 맷플롯립(Matplotlib)
: 자료를 차트(chart)나 플롯(plot)으로 시각화(visualization)하는 라이브러리
- 넘파이(NumPy)
: 통계, 선형 대수, 행렬 계산, 금융 운용 등을 포함한 과학 계산과 수학 작업에 많이 사용되는 라이브러리
- Scrapy : 웹에서 자료를 모을 때 사용하는 라이브러리
- 파이게임(Pygame) : 게임을 제작하기 위한 프레임워크

12. 필로우(Pillow) 설치



pip 명령

```
d:\W>pip
```

Usage:

```
pip <command> [options]
```

Commands:

install	Install packages.
download	Download packages.
uninstall	Uninstall packages.
freeze	Output installed packages in requirements format.

...

12. 필로우(Pillow) 설치



pip를 이용하여 필로우 라이브러리를 설치해 보겠습니다.

```
d:\W>pip install Pillow
Collecting Pillow
  Downloading Pillow-3.3.0-cp35-cp35m-win32.whl (1.3MB)
    100% |#####| 1.3MB 867kB/s
Installing collected packages: Pillow
Successfully installed Pillow-3.3.0
You are using pip version 8.1.1, however version 8.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip'
command.
```

12. 필로우(Pillow) 설치



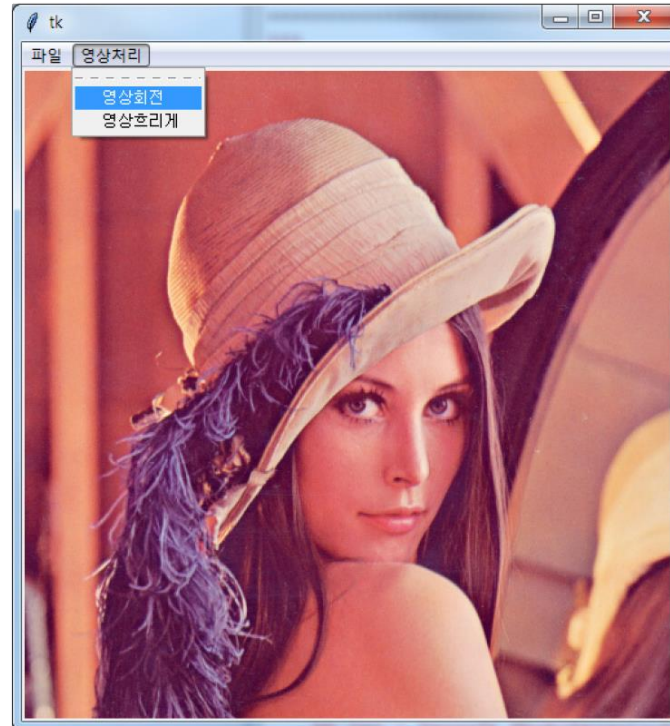
만약 다음과 같은 메시지가 나오고 설치가 실패하면 pip 버전을 업그레이드해야 합니다.

You are using pip version 7.1.2, however version 8.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

15. (Pillow) 이미지 처리 기능을 윈도우의 메뉴와 연결



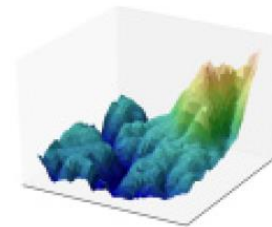
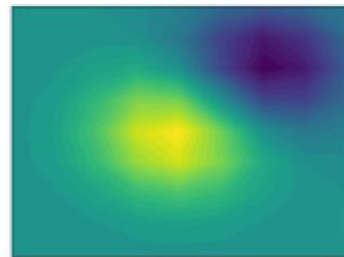
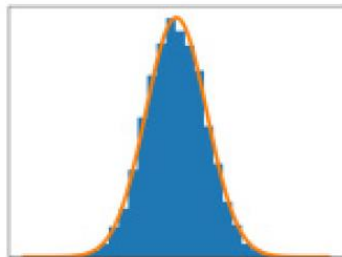
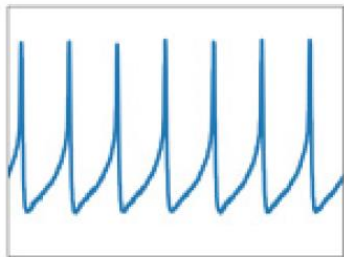
회전하기와 흐리게 하는 이미지 처리 기능을 윈도우 창의 메뉴와 연결하여 메뉴의 선택만으로 이미지 처리가 이루어지도록 해 보겠습니다.



16. 맷플롯립(Matplotlib) 설치



- 복잡한 자료를 숫자나 텍스트보다는 그래프나 차트로 데이터를 시각화하여 이해하고 분석하는 것이 효과적
- 데이터를 시각화하는데 편리한 라이브러리



16. 맷플롯립(Matplotlib) 설치



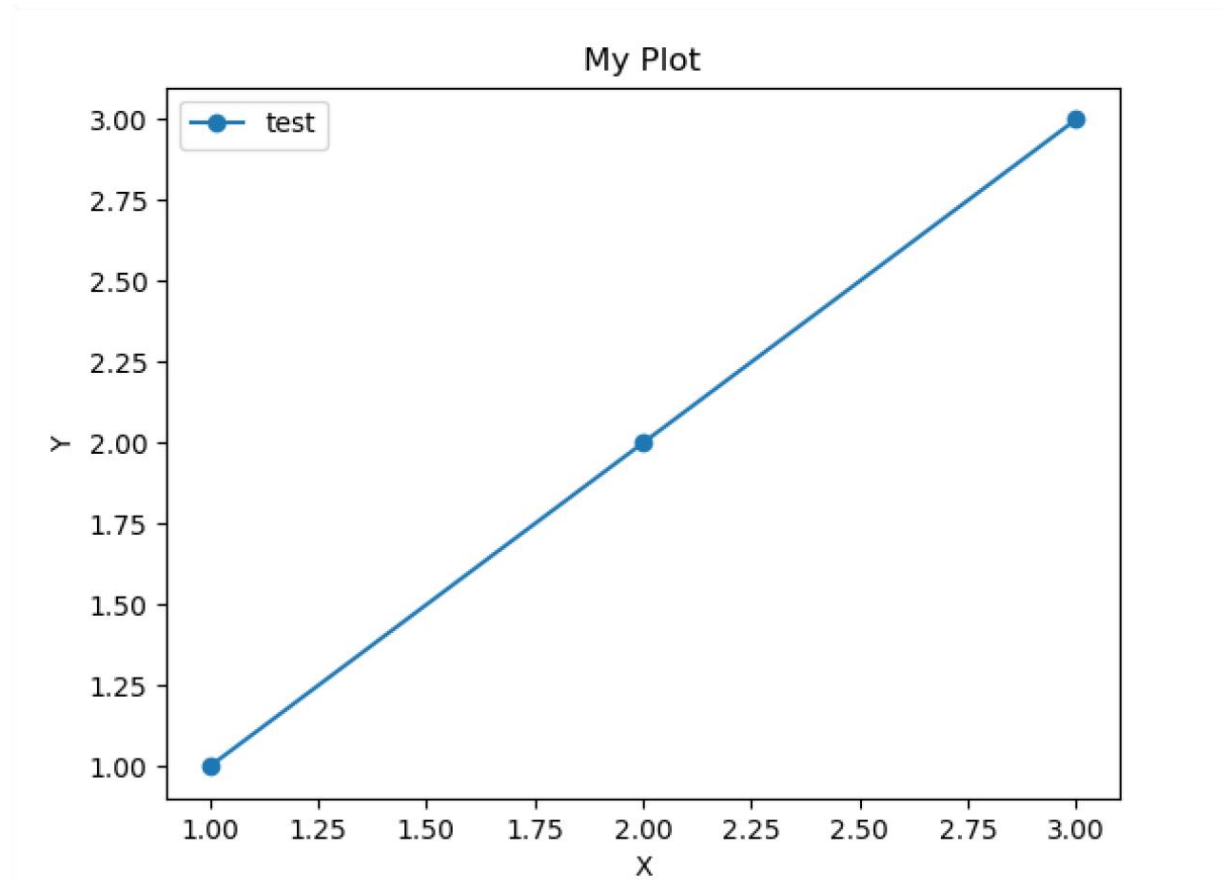
먼저, pip를 이용하여 맷플롯립 라이브러리를 설치해 보겠습니다.

```
c:\W>python -m pip install matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/39/ae/60ec3ec8f8a18c5eef71c6dff7ed0dfa0cd2f2d57b6691e595e2bc43325f/matplotlib-3.1.1-cp36-cp36m-win32.whl (8.9MB)
    #####| 8.9MB 6.8MB/s
...(생략)...
0f9d07b4fe144b3/numpy-1.17.2-cp36-cp36m-win32.whl (10.8MB)
    #####| 10.8MB 1.6MB/s
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/15/14/a7ae5a7d5fae78b40dd0d388d83010330cba2df1721b1bd91a8b99044141/kiwisolver-1.1.0-cp36-none-win32.whl (44kB)
    #####| 51kB 1.7MB/s
...(생략)...
Installing collected packages: cyclor, numpy, kiwisolver, matplotlib
Successfully installed cyclor-0.10.0 kiwisolver-1.1.0 matplotlib-3.1.1 numpy-1.17.2
```

17. [Matplotlib] 그래프 그리기 기초



(1, 1), (2, 2), (3, 3)을 선으로 잇는 간단한 그래프를 만들어 보도록 하겠습니다.



17. [Matplotlib] 그래프 그리기 기초



코드

```
from matplotlib import pyplot as plt

x = [1, 2, 3]
y = [1, 2, 3]

plt.plot(x, y, marker='o')
plt.title("My Plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend(['test'])
plt.show()
```

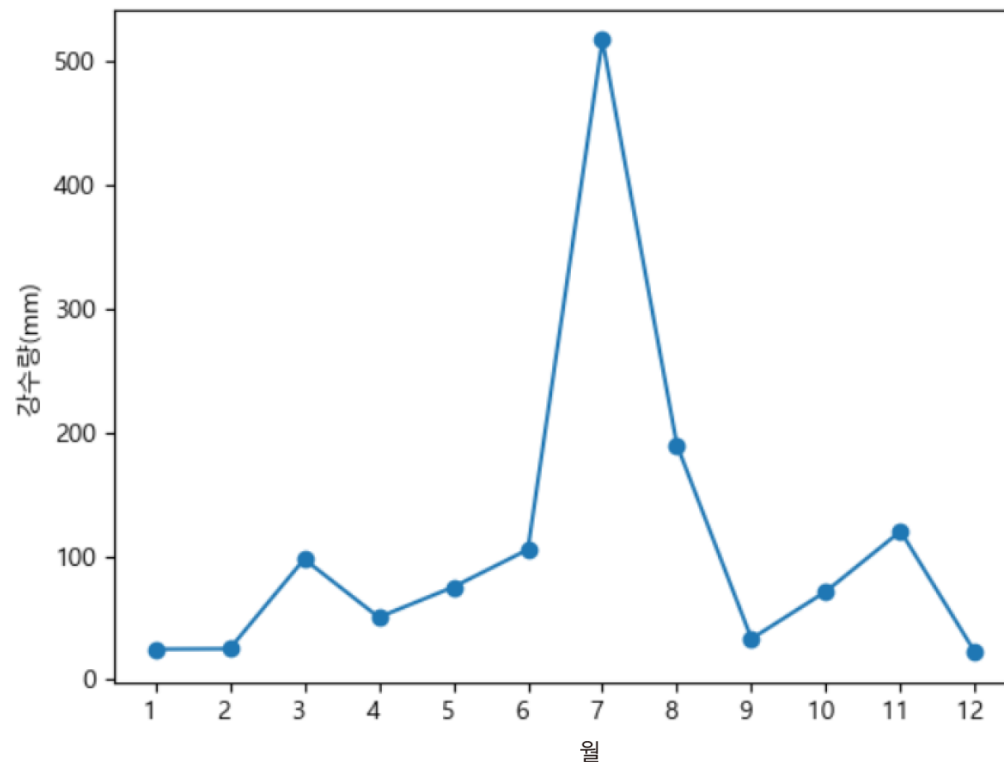
18. [Matplotlib] 강수량 그래프 만들기



우리나라 2009년의 대관령 월평균 강수량을 그래프로 그려보겠습니다.

	A	B	C
1	1	100	24.6
2	2	100	24.9
3	3	100	97.5
4	4	100	50.4
5	5	100	74.9
6	6	100	105.2

2009년 대관령 월평균 강수량



18. [Matplotlib] 강수량 그래프 만들기



코드

```
from matplotlib import pyplot as plt
import csv
from matplotlib import font_manager, rc

font_name = font_manager.FontProperties(fname="c:\Windows\Fonts\malgun.ttf").
get_name()
rc('font', family=font_name)

infile = open("d:\weather_input.csv", "r")
data = csv.reader(infile)

x = []
y = []

for line in data:
    x.append(line[0])
    y.append(float(line[2]))

plt.plot(x, y, marker='o')
plt.title("2009년 대관령 월평균 강수량")
plt.xlabel("월")
plt.ylabel("강수량(mm)")
plt.show()
infile.close()
```