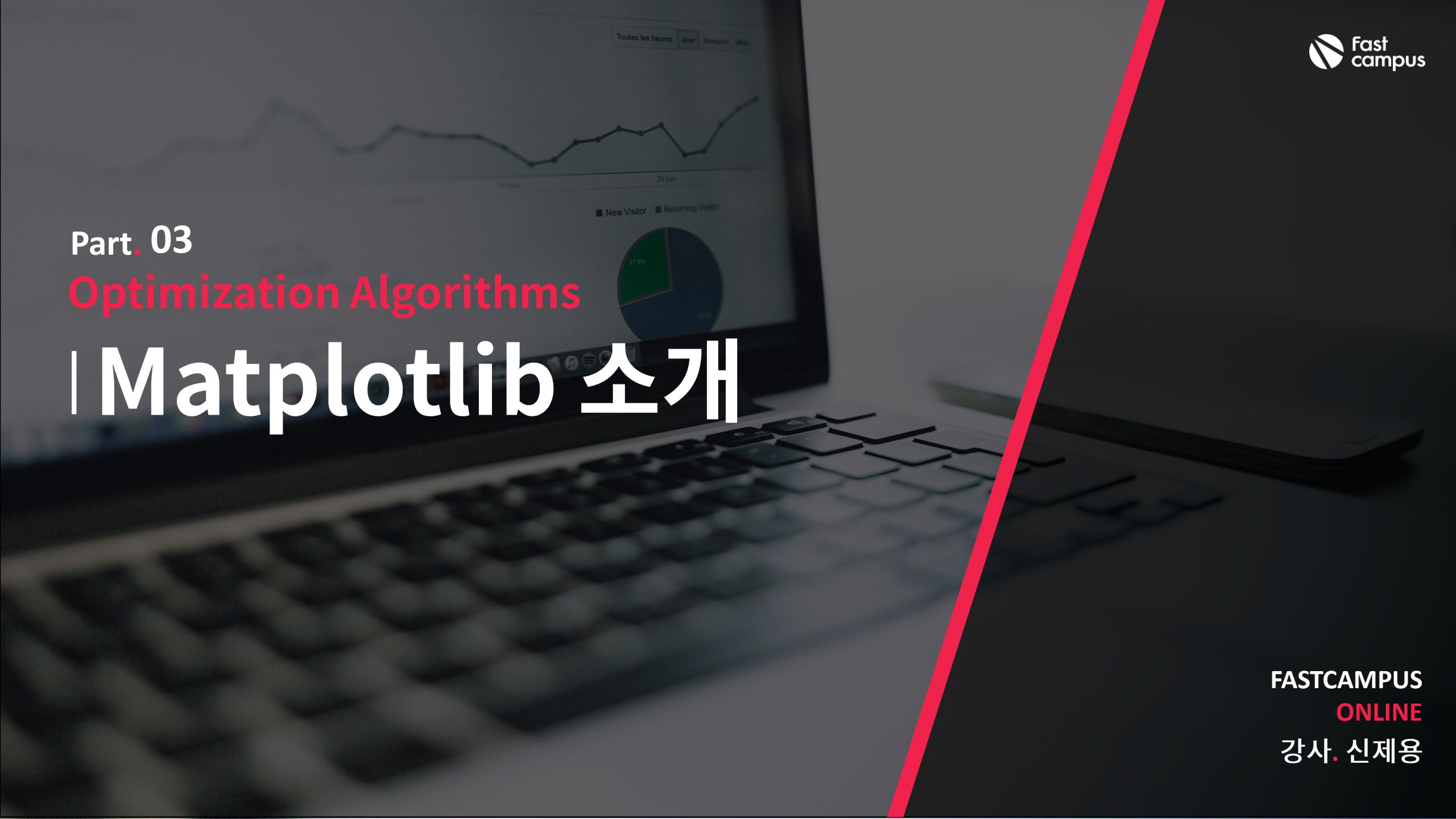


Part. 03

Optimization Algorithms

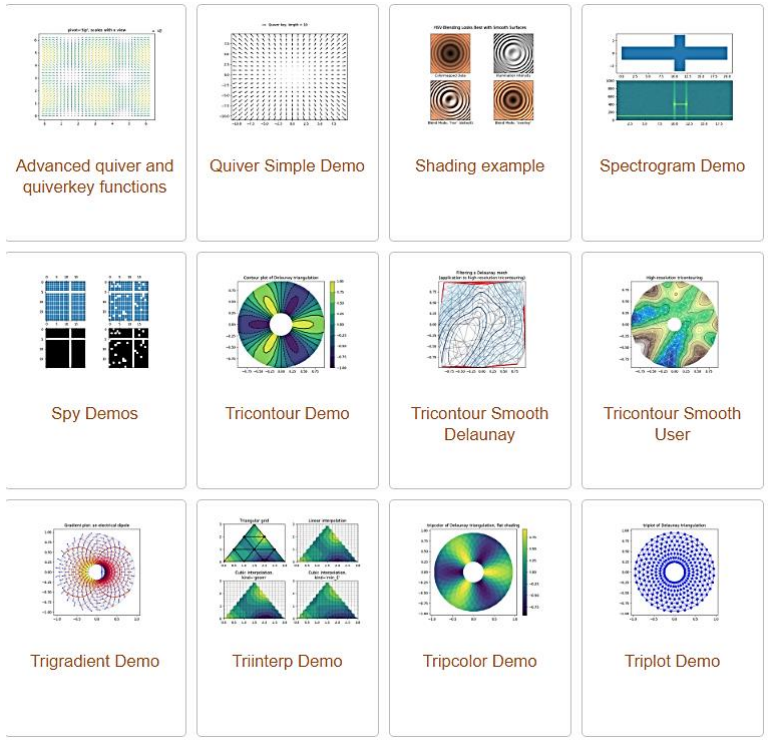
Matplotlib 소개

FASTCAMPUS
ONLINE

강사. 신제용

Matplotlib

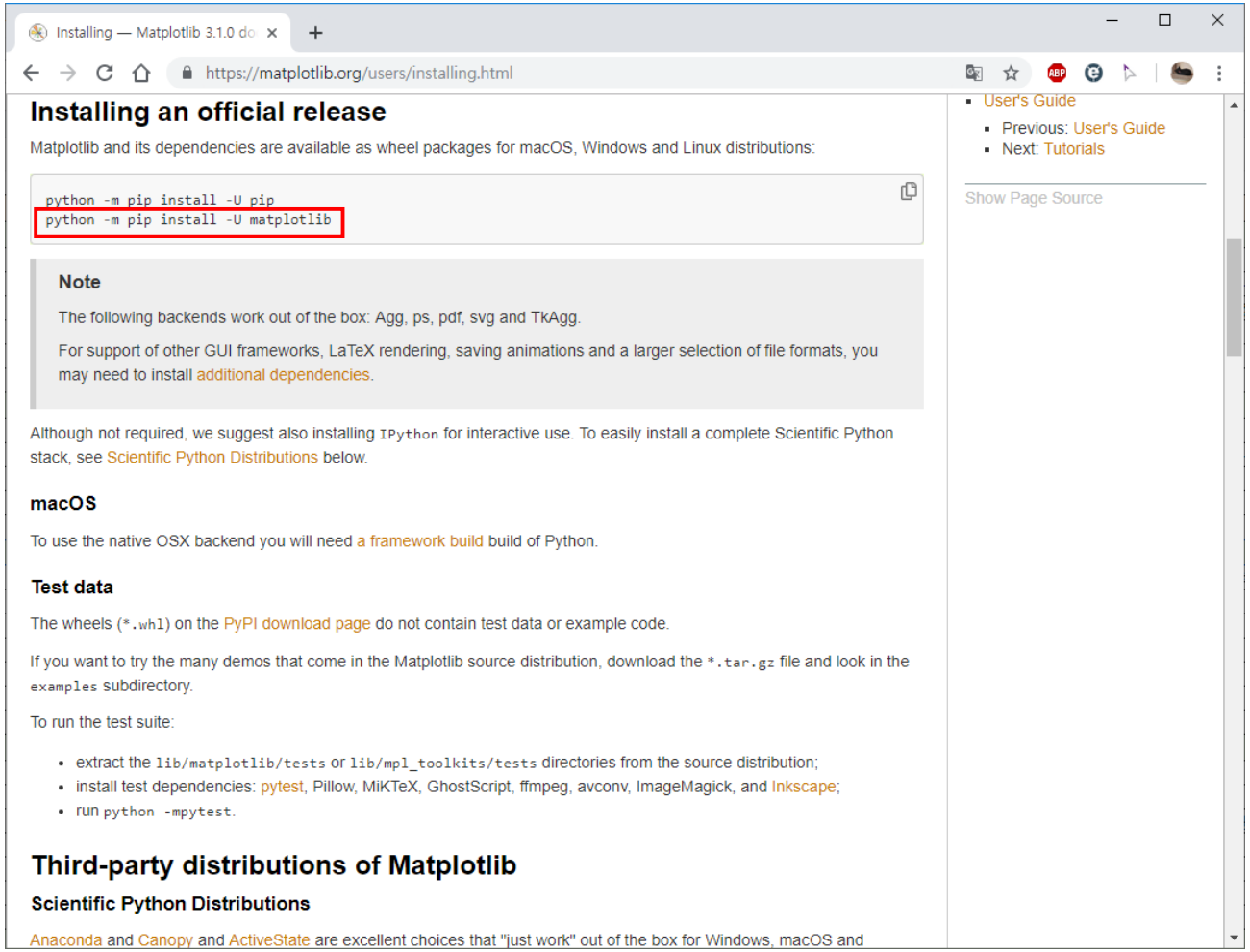
- 출판 가능한 품질로 다양한 그림(Figure)를 그려주는 오픈소스 라이브러리
- 상호작용이 가능하고 다양한 플랫폼으로 출력할 수 있음



이외에도 다양한 Figure를 표현할 수 있으며, 예제가 수록되어 있다.

https://matplotlib.org/gallery/index.html

Matplotlib 설치



pip install matplotlib

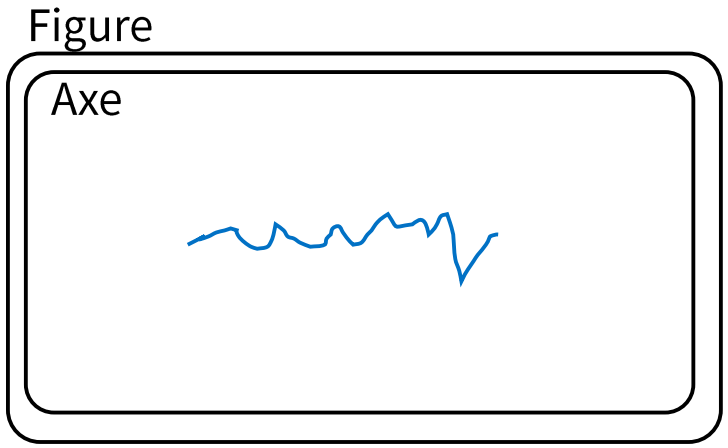
→ 최신 버전 설치

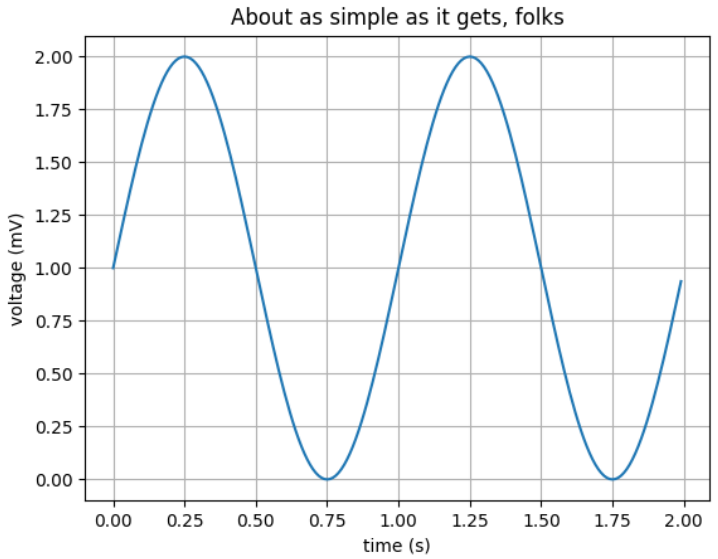
pip install matplotlib==2.1.0

→ 특정 버전 설치 (정상 동작 확인)

기본 구성 요소

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, s)

ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='About as simple as it gets, folks')
ax.grid()

fig.savefig("test.png")
plt.show()
```

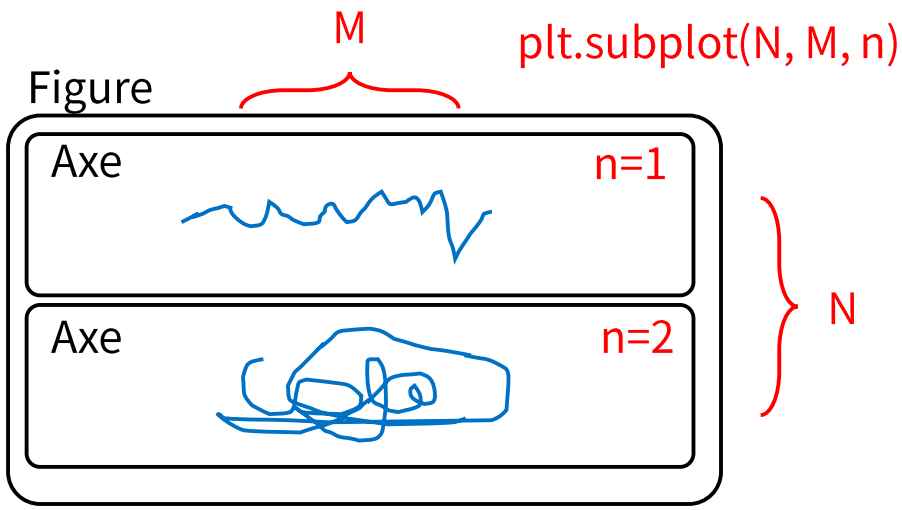


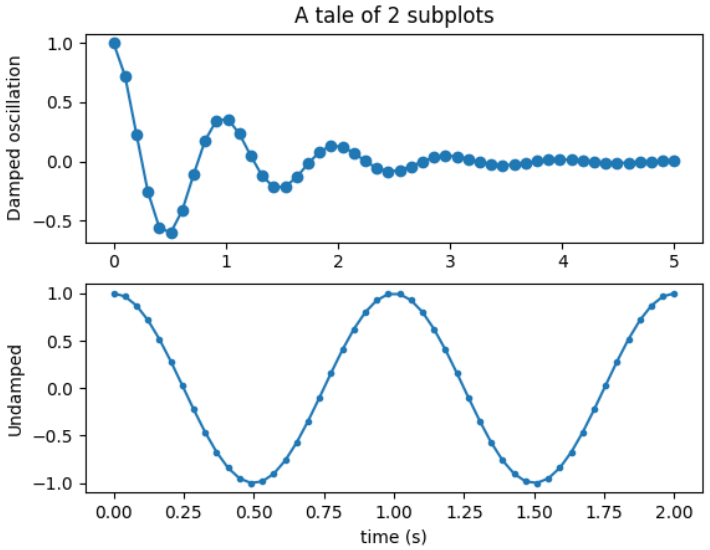


https://matplotlib.org/gallery/lines_bars_and_markers/simple_plot.html

여러 개의 그림 동시에 그리기

```
import numpy as np
import matplotlib.pyplot as plt


x1 = np.linspace(0.0, 5.0)
x2 = np.linspace(0.0, 2.0)

y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)

plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'o-')
plt.title('A tale of 2 subplots')
plt.ylabel('Damped oscillation')

plt.subplot(2, 1, 2)
plt.plot(x2, y2, '.-')
plt.xlabel('time (s)')
plt.ylabel('Undamped')

plt.show()
```

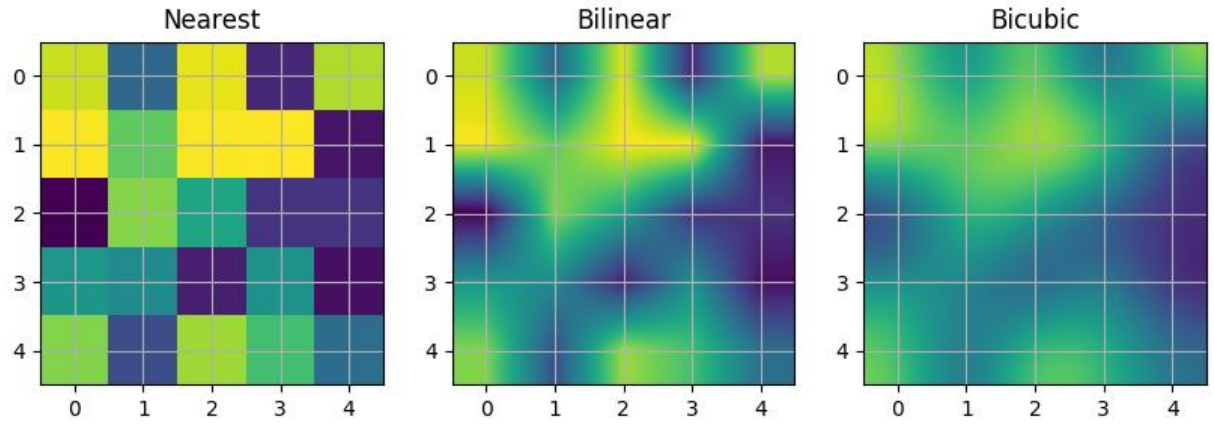




https://matplotlib.org/gallery/subplots_axes_and_figures/subplot.html

행렬 그리기


```
import numpy as np
import matplotlib.pyplot as plt


A = np.random.rand(5, 5)

fig, axs = plt.subplots(1, 3, figsize=(10, 3))
for ax, interp in zip(axs, ['nearest', 'bilinear',
'bicubic']):
    ax.imshow(A, interpolation=interp)
    ax.set_title(interp.capitalize())
    ax.grid(True)

plt.show()
```

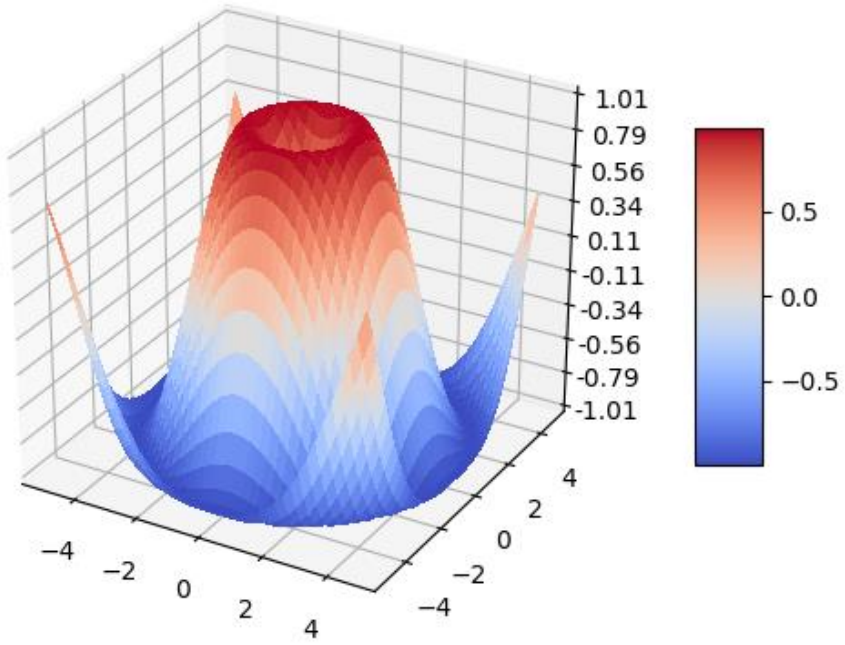


보통 Nearest를 이용하며, 학습된 필터를 확인하는 데에 유용하다.

https://matplotlib.org/gallery/images_contours_and_fields/image_demo.html

3차원 상의 표면 그리기

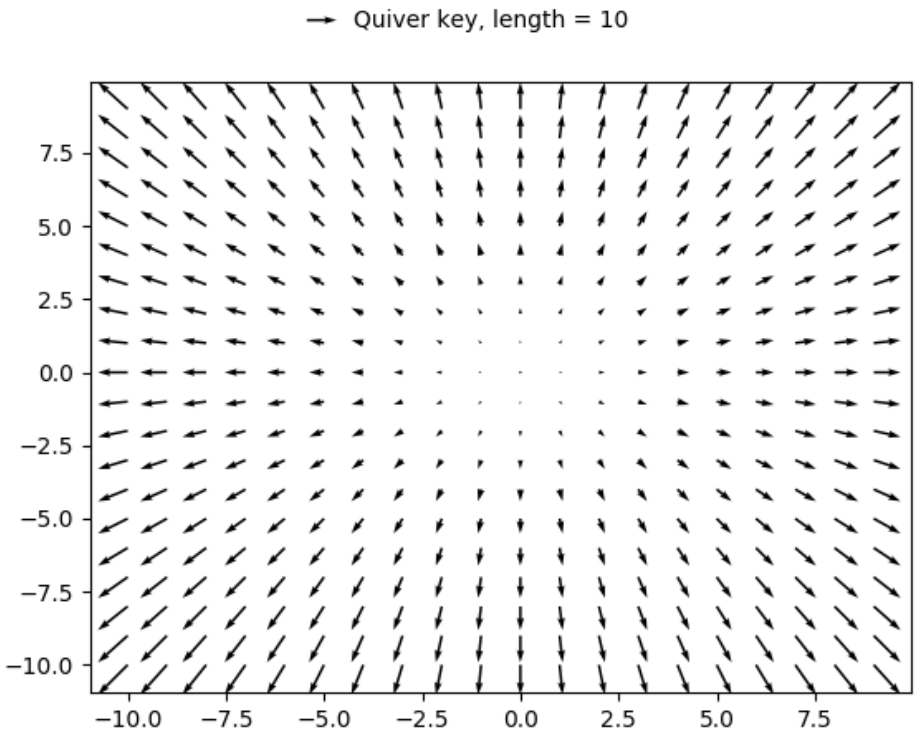
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import numpy as np


fig = plt.figure()
ax = fig.gca(projection='3d')

# Make data.
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

# Plot the surface.
surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)

# Customize the z axis.
ax.set_zlim(-1.01, 1.01)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()
```



https://matplotlib.org/gallery/mplot3d/surface3d.html

I 화살표 (Quiver) 표현하기


```
import matplotlib.pyplot as plt
import numpy as np

X = np.arange(-10, 10, 1)
Y = np.arange(-10, 10, 1)
U, V = np.meshgrid(X, Y)

fig, ax = plt.subplots()
q = ax.quiver(X, Y, U, V)
ax.quiverkey(q, X=0.3, Y=1.1, U=10,
             label='Quiver key, length = 10',
labelpos='E')

plt.show()
```



Gradient를 표현하거나, 이동 벡터를 표현하려 할 때 유용하다.

https://matplotlib.org/gallery/images_contours_and_fields/quiver_simple_demo.html#sphx-glr-gallery-images-contours-and-fields-quiver-simple-demo-py