Matplotlib

Prof.Dr. Bahadır AKTUĞ
JFM212 Python ile Mühendislik Uygulamaları

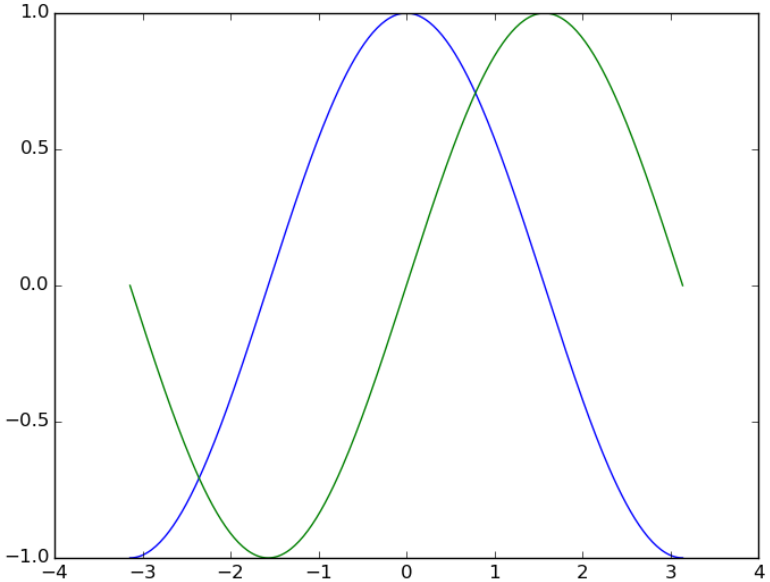
*Kaynakça bölümünde verilen kaynaklardan derlenmiştir.

Matplotlib

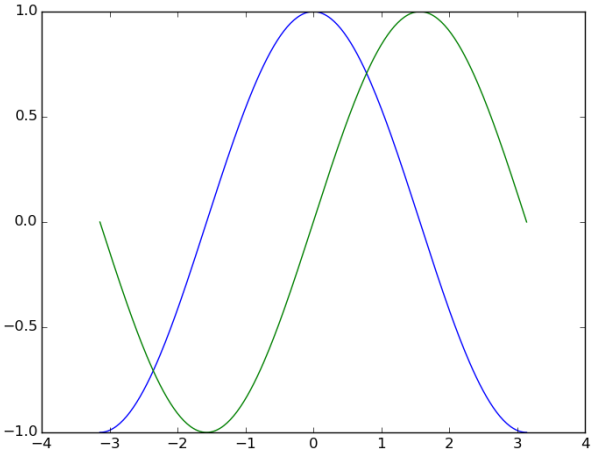
- Matplotlib, Python grafikleri için en çok kullanılan eklentidir.
- Matplotlib, Numpy dizileriyle çalışabilecek şekilde tasarlanmıştır.
- Matplotlib grafik kütüphanesi altında Pylab kütüphanesi bulunmaktadır ve büyük ölçüde Matlab grafik komutlarıyla benzerlik taşımaktadır.

Matplotlib – İlk Grafik

```
import numpy as np
import matplotlib.pyplot as plt
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
plt.plot(X, C)
plt.plot(X, S)
plt.show()
```



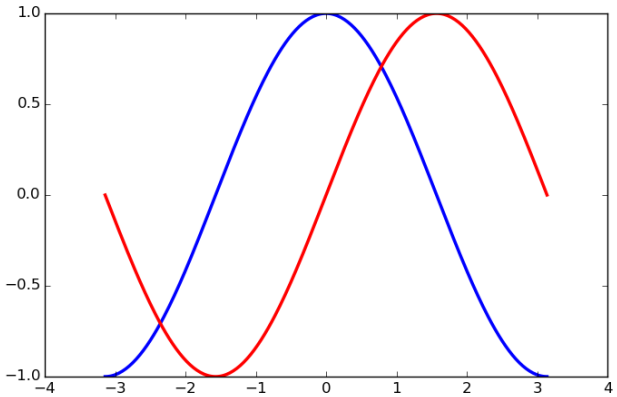
Matplotlib – Varsayılan Grafik Parametreleri




```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-")
plt.plot(X, S, color="green", linewidth=1.0, linestyle="-")
plt.xlim(-4.0, 4.0)
plt.xticks(np.linspace(-4, 4, 9, endpoint=True))
plt.ylim(-1.0, 1.0)
plt.yticks(np.linspace(-1, 1, 5, endpoint=True))
plt.show()
```

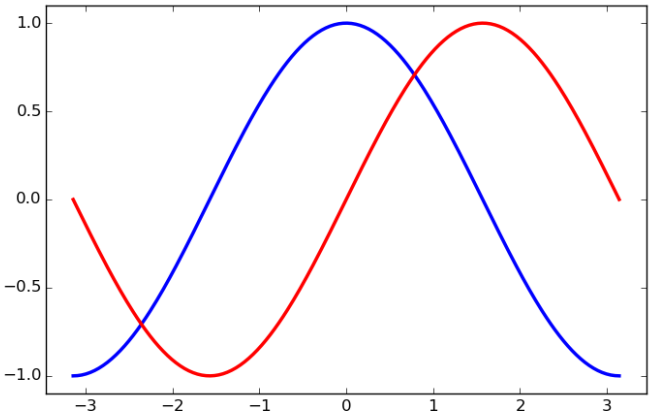
Matplotlib – Renkler ve Çizgi Kalınlıkları

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-")
plt.xlim(-4.0, 4.0)
plt.xticks(np.linspace(-4, 4, 9, endpoint=True))
plt.ylim(-1.0, 1.0)
plt.yticks(np.linspace(-1, 1, 5, endpoint=True))
plt.show()
```



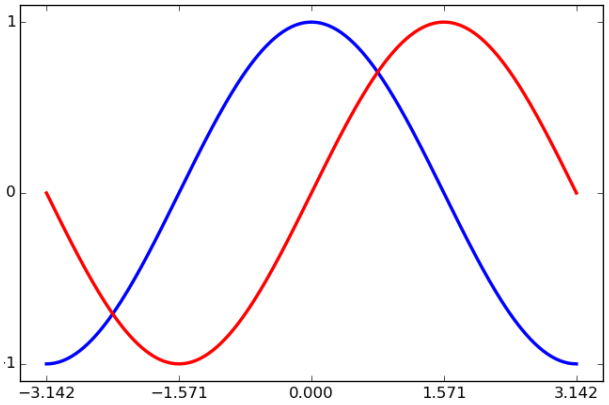
Matplotlib – Eksen Limitleri

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
S = np.sin(X)
C = np.cos(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-")
plt.xlim(X.min() * 1.1, X.max() * 1.1)
plt.ylim(C.min() * 1.1, C.max() * 1.1)
plt.show()
```



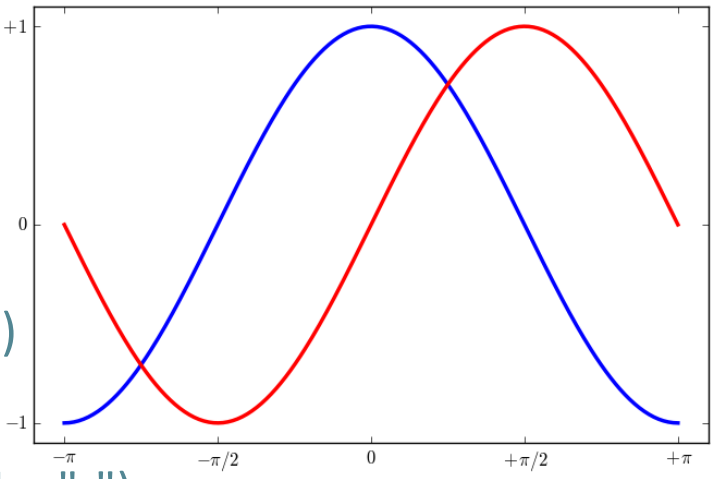
Matplotlib – Eksen Bölümlemeleri

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
S = np.sin(X)
C = np.cos(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-")
plt.xlim(X.min() * 1.1, X.max() * 1.1)
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi])
plt.ylim(C.min() * 1.1, C.max() * 1.1)
plt.yticks([-1, 0, +1])
plt.show()
```



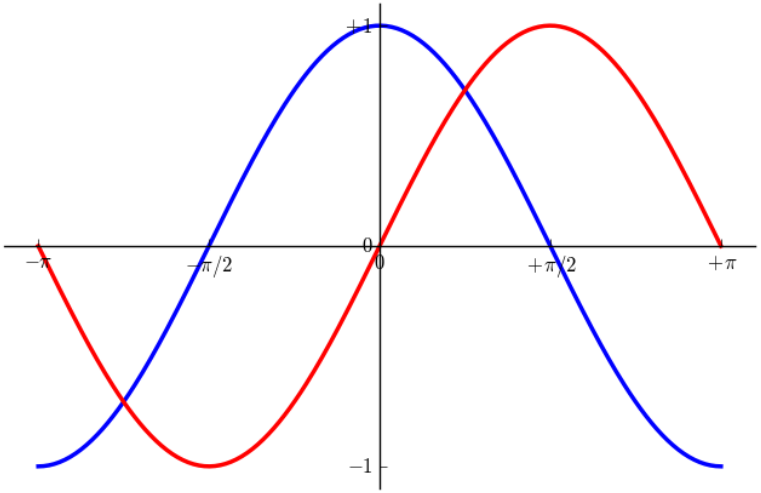
Matplotlib – Eksen Bölümleme Etiketleri

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-")
plt.xlim(X.min() * 1.1, X.max() * 1.1)
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi], [r'$-\pi$', r'$-\pi/2$',
r'$0$', r'$+\pi/2$', r'$+\pi$'])
plt.ylim(C.min() * 1.1, C.max() * 1.1)
plt.yticks([-1, 0, +1], [r'$-1$', r'$0$', r'$+1$'])
plt.show()
```



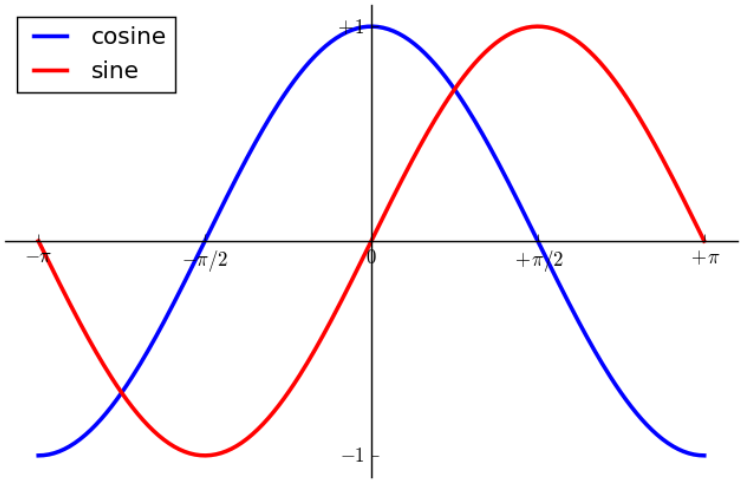
Matplotlib – Koordinat Eksenlerinin Konumu

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C, S = np.cos(X) , np.sin(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-")
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
plt.xlim(X.min() * 1.1, X.max() * 1.1)
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$'])
plt.ylim(C.min() * 1.1, C.max() * 1.1)
plt.yticks([-1, 0, +1], [r'$-1$', r'$0$', r'$+1$'])
plt.show()
```



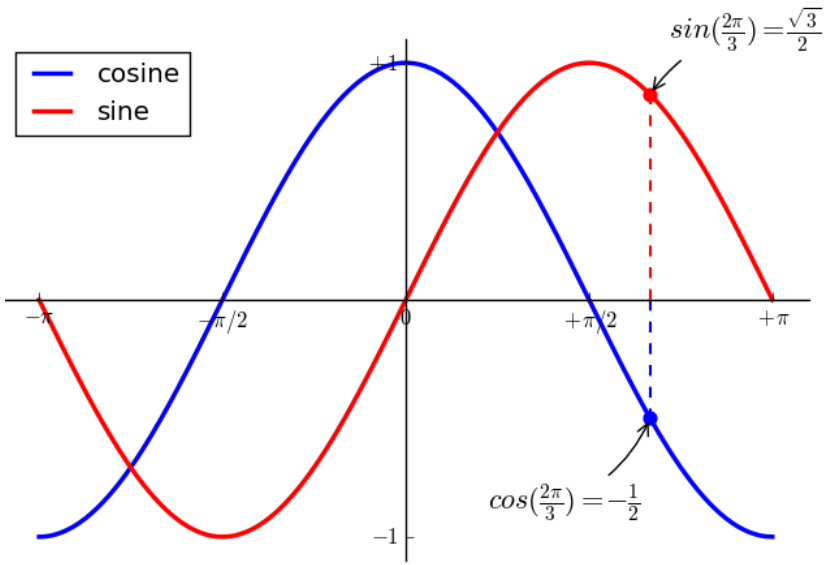
Matplotlib – Legend

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5), dpi=80)
plt.subplot(111)
X = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(X), np.sin(X)
plt.plot(X, C, color="blue", linewidth=2.5, linestyle="-", label="cosine")
plt.plot(X, S, color="red", linewidth=2.5, linestyle="-", label="sine")
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
plt.xlim(X.min() * 1.1, X.max() * 1.1)
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$'])
plt.ylim(C.min() * 1.1, C.max() * 1.1)
plt.yticks([-1, +1], [r'$-1$', r'$+1$'])
plt.legend(loc='upper left')
plt.show()
```



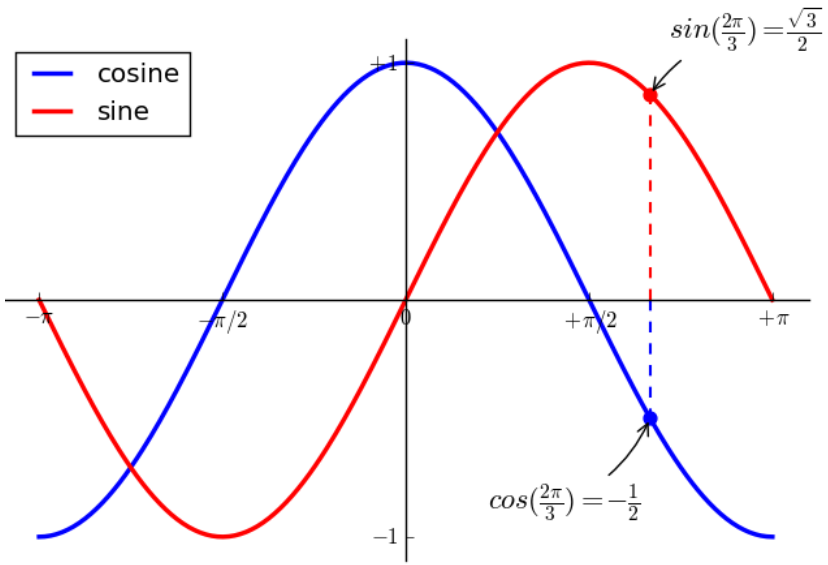
Matplotlib – Annotation

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5), dpi=80)
plt.subplot(111)
.
.
plt.scatter([t, ], [np.cos(t), ], 50, color='blue')
plt.annotate(r'$sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
xy=(t, np.sin(t)), xycoords='data',
xytext=(+10, +30), textcoords='offset points', fontsize=16,
arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
.
.
plt.scatter([t, ], [np.sin(t), ], 50, color='red')
plt.annotate(r'$cos(\frac{2\pi}{3})=-\frac{1}{2}$', xy=(t, np.cos(t)),
xycoords='data', xytext=(-90, -50), textcoords='offset points',
fontsize=16,  arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
plt.show()
```



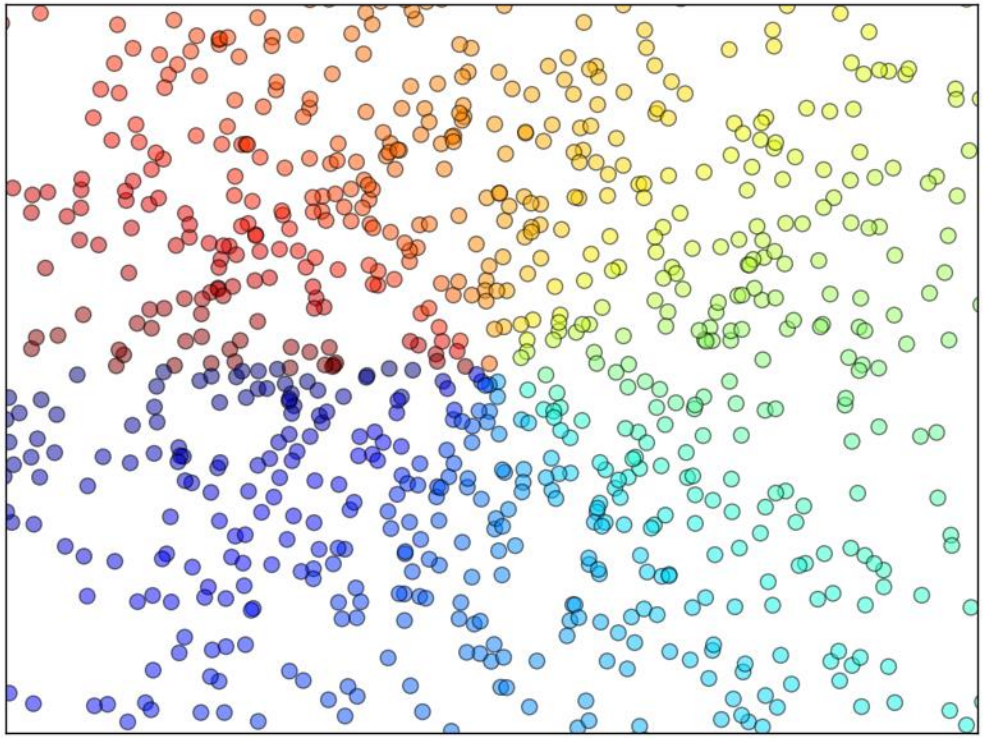
Matplotlib – Annotation

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5), dpi=80)
plt.subplot(111)
.
.
plt.scatter([t, ], [np.cos(t), ], 50, color='blue')
plt.annotate(r'$sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
xy=(t, np.sin(t)), xycoords='data',
xytext=(+10, +30), textcoords='offset points', fontsize=16,
arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
.
.
plt.scatter([t, ], [np.sin(t), ], 50, color='red')
plt.annotate(r'$cos(\frac{2\pi}{3})=-\frac{1}{2}$', xy=(t, np.cos(t)),
xycoords='data', xytext=(-90, -50), textcoords='offset points',
fontsize=16,  arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2"))
plt.show()
```



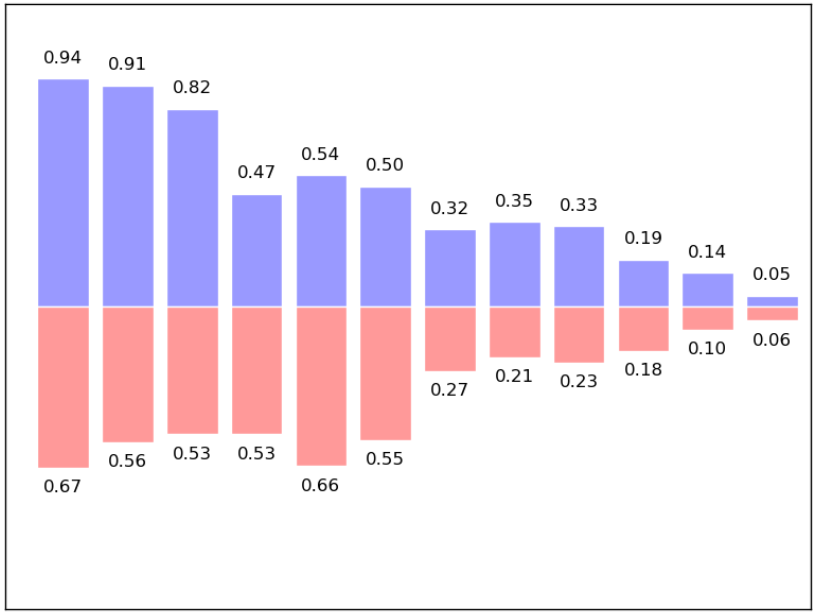
Matplotlib-Nokta Dağılım (Scatter)

```
import numpy as np
import matplotlib.pyplot as plt
n = 1024
X = np.random.normal(0, 1, n)
Y = np.random.normal(0, 1, n)
T = np.arctan2(Y, X)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.scatter(X, Y, s=75, c=T, alpha=.5)
plt.xlim(-1.5, 1.5)
plt.xticks(())
plt.ylim(-1.5, 1.5)
plt.yticks(())
plt.show()
```



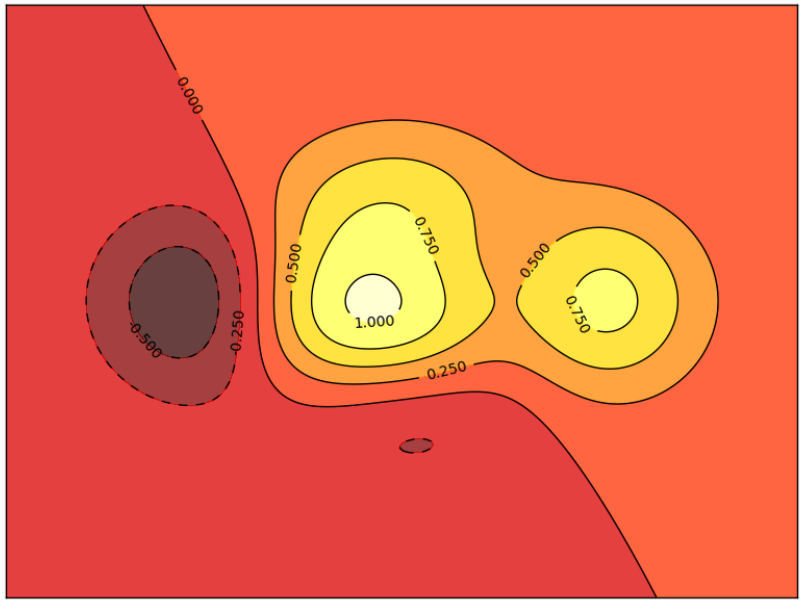
Matplotlib-Çubuk Grafikleri

```
import numpy as np
import matplotlib.pyplot as plt
n = 12
X = np.arange(n)
Y1 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
Y2 = (1 - X / float(n)) * np.random.uniform(0.5, 1.0, n)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')
plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')
for x, y in zip(X, Y1):
plt.text(x + 0.4, y + 0.05, '%.2f' % y, ha='center', va= 'bottom')
for x, y in zip(X, Y2):
plt.text(x + 0.4, -y - 0.05, '%.2f' % y, ha='center', va= 'top')
plt.xlim(-.5, n)
plt.xticks(())
plt.ylim(-1.25, 1.25)
plt.yticks(())
plt.show()
```



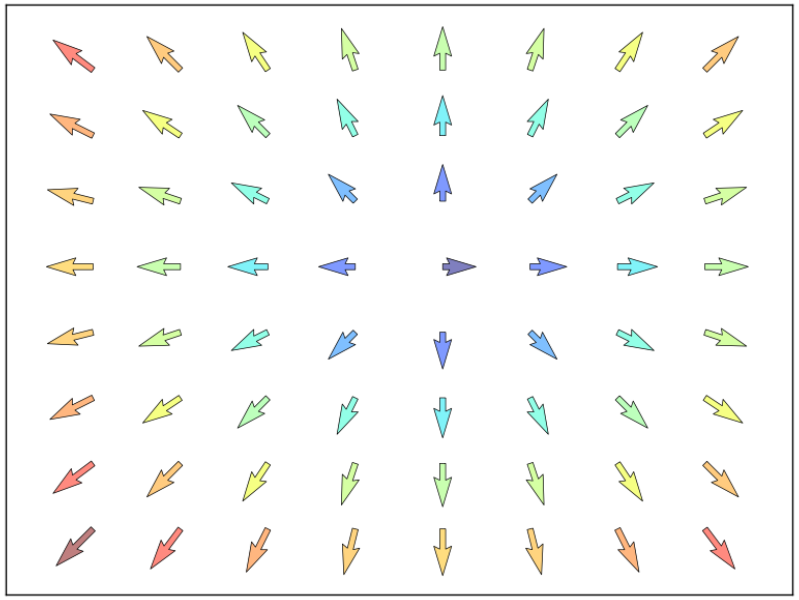
Matplotlib-Münhani Grafikleri

```
import numpy as np
import matplotlib.pyplot as plt
def f(x,y):
return (1 - x / 2 + x**5 + y**3) * np.exp(-x**2 -y**2)
n = 256
x = np.linspace(-3, 3, n)
y = np.linspace(-3, 3, n)
X,Y = np.meshgrid(x, y)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.contourf(X, Y, f(X, Y), 8, alpha=.75, cmap=plt.cm.hot)
C = plt.contour(X, Y, f(X, Y), 8, colors='black', linewidth=.5)
plt.clabel(C, inline=1, fontsize=10)
plt.xticks(())
plt.yticks(())
plt.show()
```



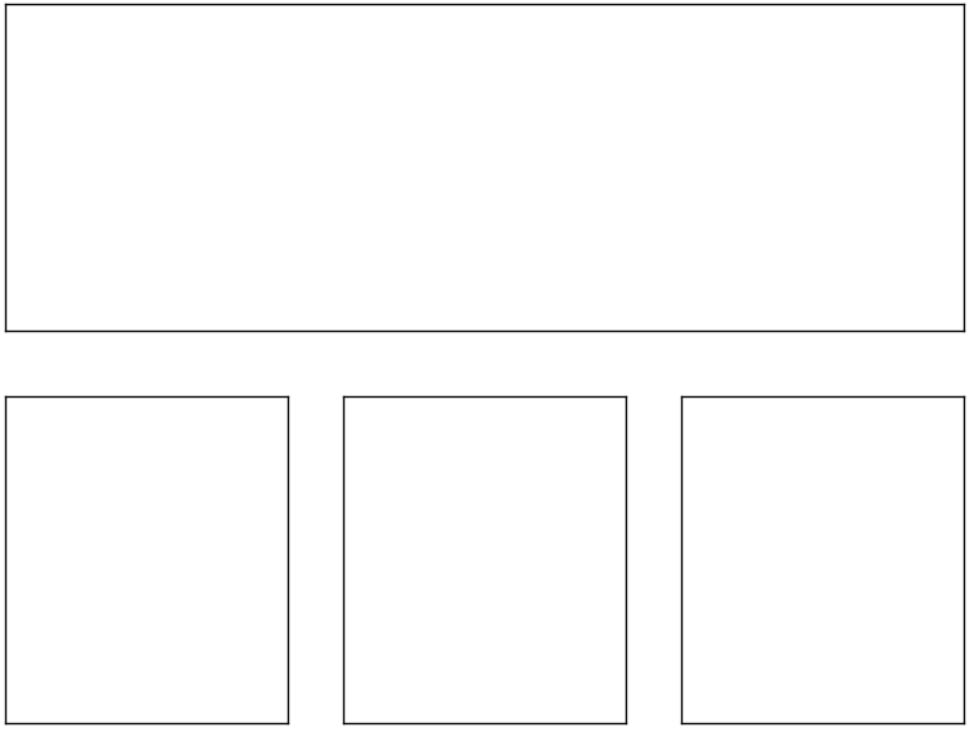
Matplotlib-Vektör Alan Grafikleri

```
import numpy as np
import matplotlib.pyplot as plt
n = 8
X, Y = np.mgrid[0:n, 0:n]
T = np.arctan2(Y - n / 2., X - n/2.)
R = 10 + np.sqrt((Y - n / 2.0) ** 2 + (X - n / 2.0) ** 2)
U, V = R * np.cos(T), R * np.sin(T)
plt.axes([0.025, 0.025, 0.95, 0.95])
plt.quiver(X, Y, U, V, R, alpha=.5)
plt.quiver(X, Y, U, V, edgecolor='k', facecolor='None', linewidth=.5)
plt.xlim(-1, n)
plt.xticks(())
plt.ylim(-1, n)
plt.yticks(())
plt.show()
```



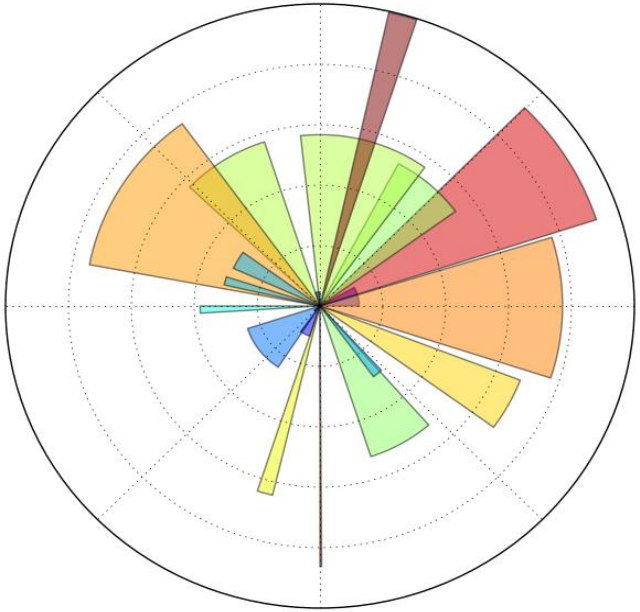
Matplotlib-Çoklu Grafikler

```
import matplotlib.pyplot as plt
fig = plt.figure()
fig.subplots_adjust(bottom=0.025, left=0.025, top = 0.975, right=0.975)
plt.subplot(2, 1, 1)
plt.xticks(()), plt.yticks(())
plt.subplot(2, 3, 4)
plt.xticks(())
plt.yticks(())
plt.subplot(2, 3, 5)
plt.xticks(())
plt.yticks(())
plt.subplot(2, 3, 6)
plt.xticks(())
plt.yticks(())
plt.show()
```



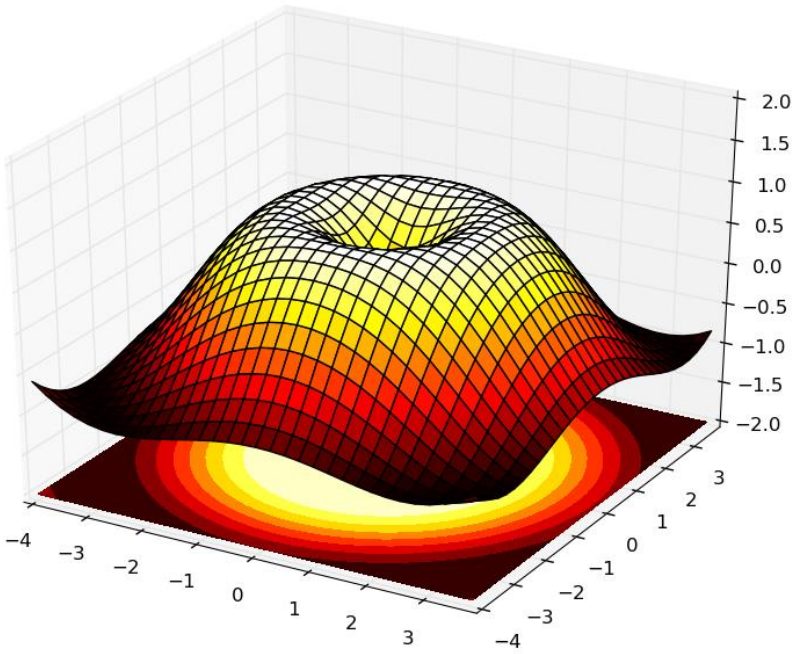
Matplotlib-Kutupsal Grafikler

```
import numpy as np
import matplotlib.pyplot as plt
ax = plt.axes([0.025, 0.025, 0.95, 0.95], polar=True)
N = 20
theta = np.arange(0.0, 2 * np.pi, 2 * np.pi / N)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
bars = plt.bar(theta, radii, width=width, bottom=0.0)
for r,bar in zip(radii, bars):
bar.set_facecolor(plt.cm.jet(r/10.))
bar.set_alpha(0.5)
ax.set_xticklabels([])
ax.set_yticklabels([])
plt.show()
```



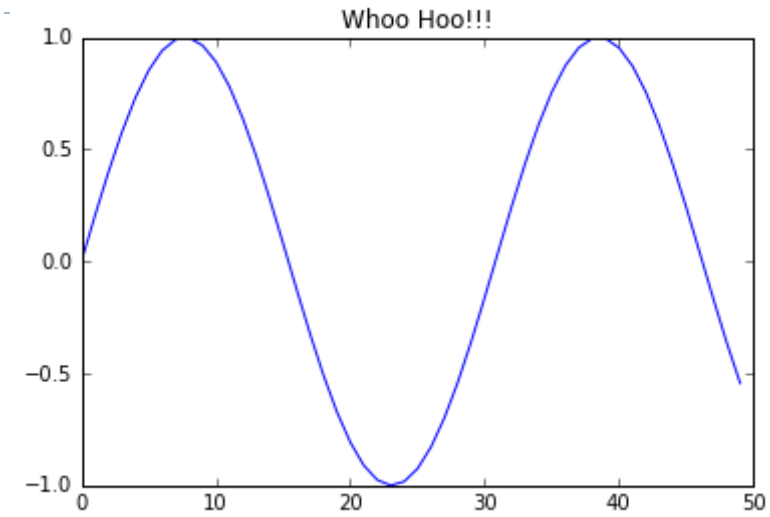
Matplotlib-3B Yüzey Grafikleri

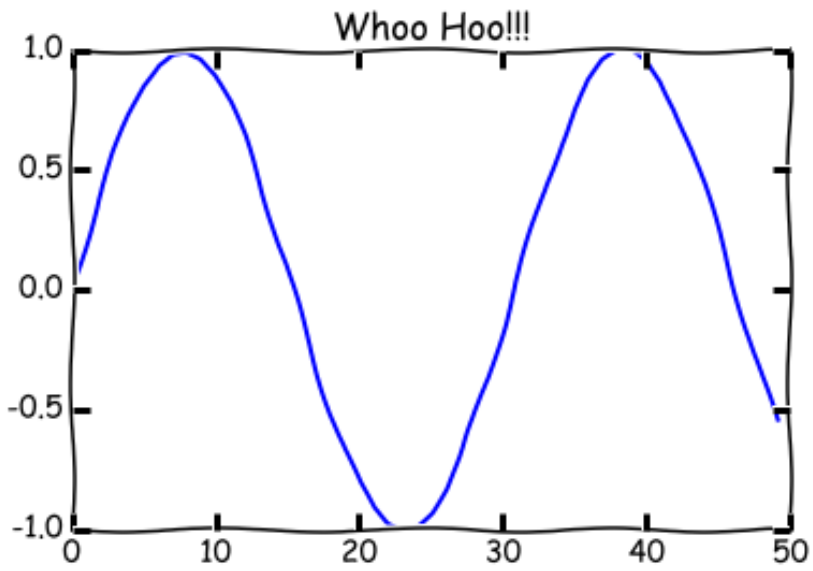
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X ** 2 + Y ** 2)
Z = np.sin(R)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.cm.hot)
ax.contourf(X, Y, Z, zdir='z', offset=-2, cmap=plt.cm.hot)
ax.set_zlim(-2, 2)
plt.show()
```



XKCD Stili Matplotlib Grafikleri

```
import numpy as np
import matplotlib.pyplot as plt
plt.xkcd()
plt.plot(np.sin(np.linspace(0, 10)))
plt.title('Whoo Hoo!!!')
```

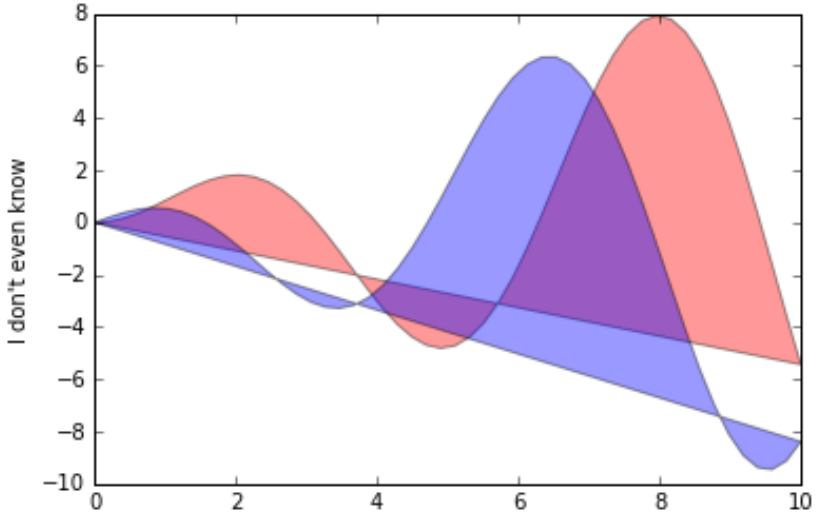


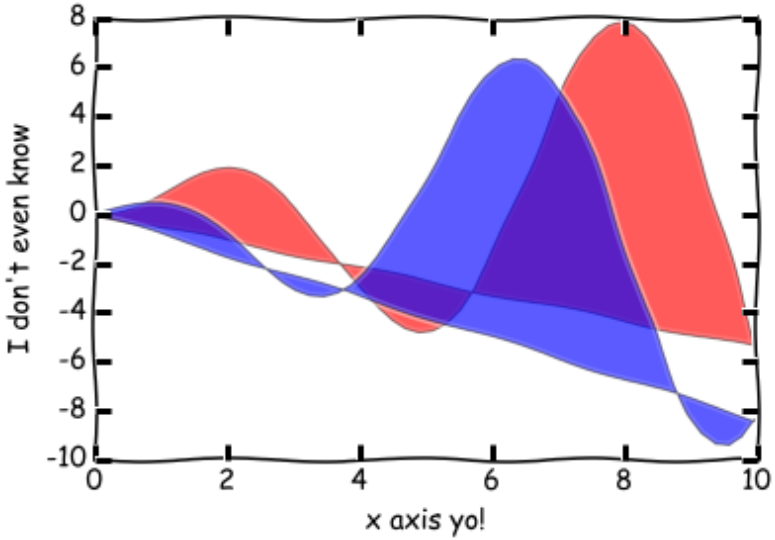


XKCD Stili Matplotlib Grafikleri

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 10)
y1 = x * np.sin(x)
y2 = x * np.cos(x)

plt.xkcd()
plt.fill(x, y1, 'red', alpha=0.4)
plt.fill(x, y2, 'blue', alpha=0.4)
plt.xlabel('x axis yo!')
plt.ylabel("I don't even know")
```





Kaynakça

1 Wentworth, P., Elkner, J., Downey, A.B., Meyers, C. (2014). How to Think Like a Computer Scientist: Learning with Python (3nd edition).
2 Pilgrim, M. (2014). Dive into Python 3 by. Free online version: DiveIntoPython3.org ISBN: 978-1430224150.
3 Summerfield, M. (2014) Programming in Python 3 2nd ed (PIP3) : - Addison Wesley ISBN: 0-321-68056-1.
4 Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/.
5 Millman, K.J., Aivazis, M. (2011). Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12.
6 John D. Hunter (2007). Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95.
7 Travis E. Oliphant (2007). Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20.
8 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). Data Structures and Algorithms in Python, Wiley.
9 http://www.diveintopython.net/
10 https://docs.python.org/3/tutorial/
11 http://www.python-course.eu
12 https://developers.google.com/edu/python/
13 http://learnpythonthehardway.org/book/