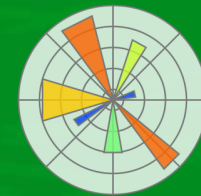


Matplotlib



Η Matplotlib είναι μια βιβλιοθήκη γραφικών της Python που μας παρέχει δυνατότητες σχεδίασης γραφικών παραστάσεων υψηλής ποιότητας. Η δημιουργία γραφημάτων μπορεί να γίνει είτε μέσα από προγράμματα ή μέσω διαδραστικού περιβάλλοντος όπως είναι το IPython ή με ενσωμάτωση σε ιστοσελίδες και σε εφαρμογές με γραφικό περιβάλλον.

Θα επικεντρωθούμε στη βιβλιοθήκη **Pyplot**, ένα τμήμα της Matplotlib που αποτελεί μια συλλογή εντολών με λειτουργίες παρόμοιες μ' αυτές του MATLAB.

Περισσότερες πληροφορίες μπορούμε να βρούμε στον ιστότοπο <http://matplotlib.org>.

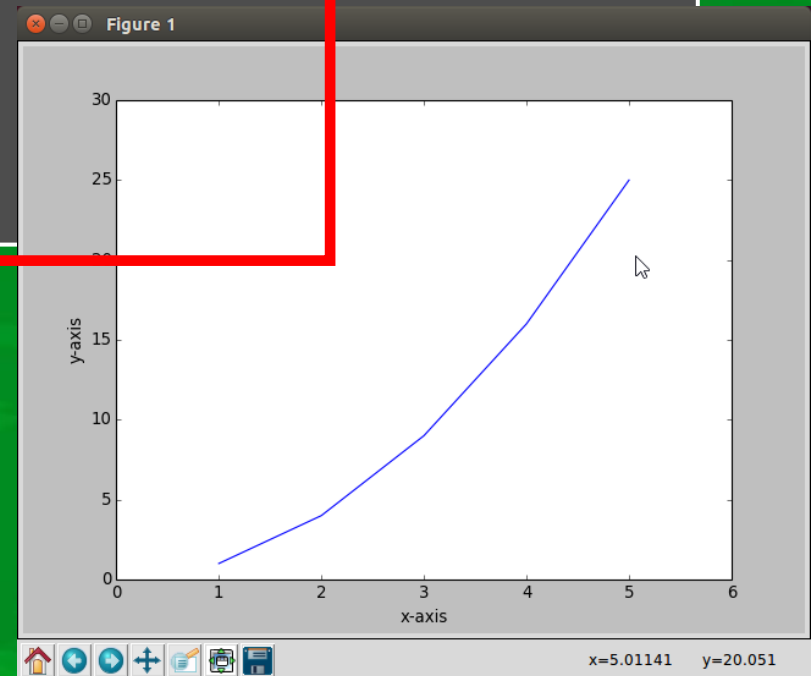
Matplotlib

Στο πρώτο μας παράδειγμα θα ενώσουμε με μια μπλε γραμμή τις συντεταγμένες των σημείων (1, 1), (2, 4), (3, 9), (4, 16) και (5, 25):

```
# simple plot example
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4, 5], [1, 4, 9, 16, 25])
plt.axis([0, 6, 0, 30])
plt.ylabel('y-axis')
plt.xlabel('x-axis')
plt.show()
```

Η plot σχεδιάζει τη γραμμή που καθορίζουν τα σημεία με συντεταγμένες που δίνονται ως δύο ξεχωριστές λίστες. Αν δώσουμε ως όρισμα μία μόνο λίστα, θα χρησιμοποιηθεί αυτόματα για την άλλη διάσταση η ακολουθία 0, 1, ... N-1.



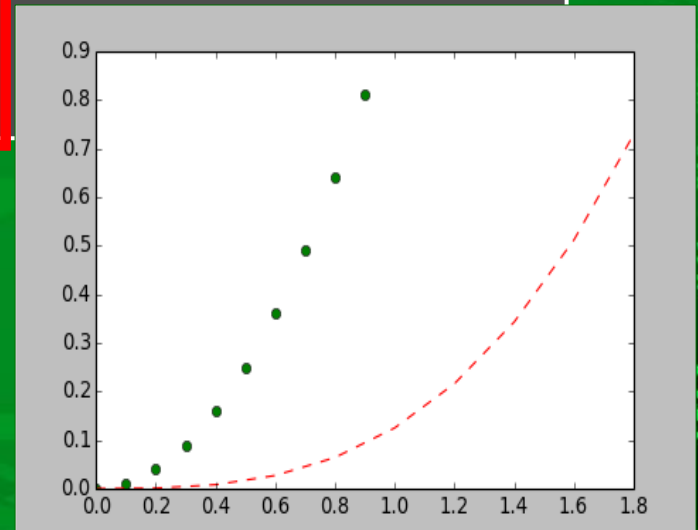
Matplotlib

Η plot έχει τη δυνατότητα να σχεδιάσει περισσότερες από μία γραφικές παραστάσεις στο ίδιο σύστημα συντεταγμένων.

```
# plot 2 lines
import numpy as np
import matplotlib.pyplot as plt

x1 = np.arange(0, 1, .1); y1 = x1 * x1
x2 = 2 * x1; y2 = y1 ** 1.5

plt.figure(figsize=(6, 4)) # set size to 6x4 inches
plt.plot(x1, y1, 'go', x2, y2, 'r--')
plt.show()
```



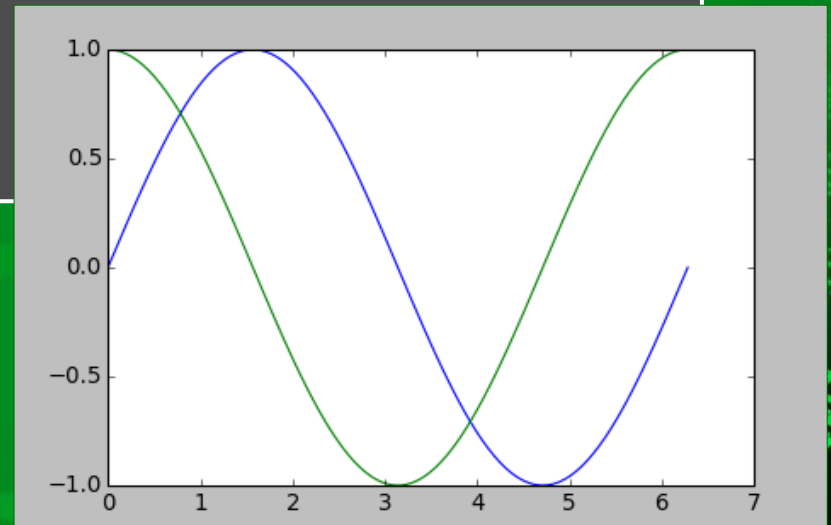
Matplotlib

Γραφικές παραστάσεις του ημιτόνου και του συνημιτόνου στο διάστημα $[0, 2\pi]$.

```
# plot sin, cos
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(0, 2*np.pi, 50, endpoint=True)
S = np.sin(X)
C = np.cos(X)

plt.figure(figsize=(6, 4))
plt.plot(X, C, X, S)
plt.show()
```



Matplotlib

Υπάρχουν διάφοροι τρόποι να αλλάξουμε τη μορφοποίηση μιας γραμμής. Ο συνηθέστερος είναι μέσω ονομαστικών ορισμάτων στην εντολή plot:

```
plt.plot(x, y, color='red', linewidth=2.0, linestyle='-.')
```

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε την αντίστοιχη μέθοδο setter του αντικειμένου. Η plot επιστρέφει μια λίστα με τις γραμμές που σχεδίασε, οπότε μπορούμε να εφαρμόσουμε αλλαγές στο αντικείμενο που επιθυμούμε:

```
line1, line2 = plt.plot(x1, y1, x2, y2)
line1.set_visible(False)      # turn off visibility of 1st line
line2.set_marker('*')         # set * as point marker for 2nd
```

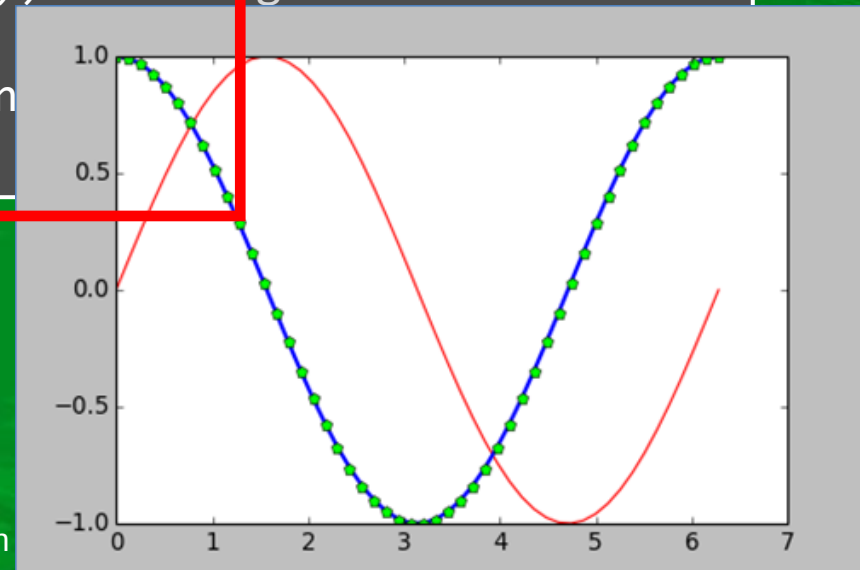
Τέλος, η εντολή **setp** μας επιτρέπει να ορίσουμε τις τιμές των ιδιοτήτων μιας ή περισσότερων γραμμών ταυτόχρονα:

```
line, = plt.plot(x, y)        # 1st list element
plt.setp(line, linestyle='--') # set dashed linestyle
```

Matplotlib

```
# line attributes
import numpy as np
import matplotlib.pyplot as plt

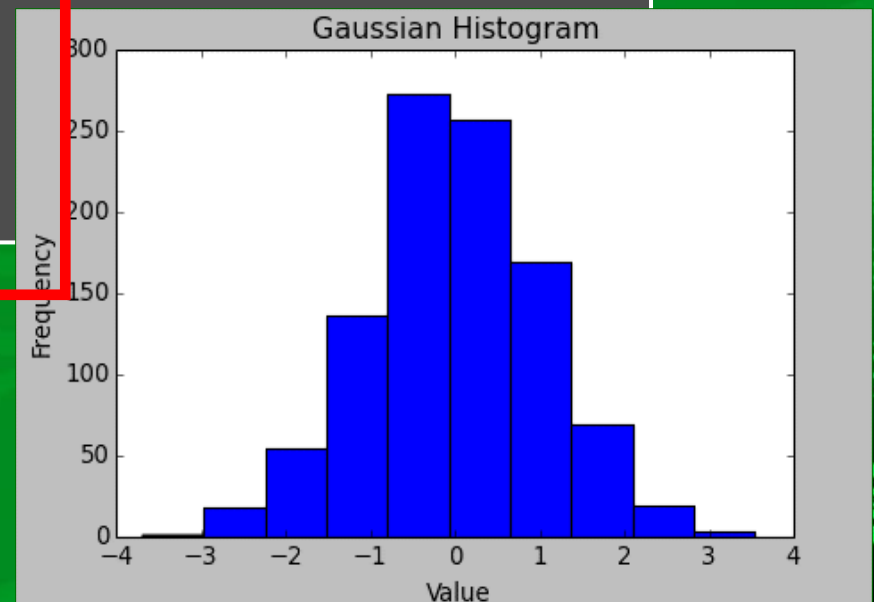
X = np.linspace(0, 2*np.pi, 50, endpoint=True)
S = np.sin(X); C = np.cos(X); plt.figure(figsize=(6, 4))
sinLine, cosLine = plt.plot(X, S, X, C)
plt.setp(sinLine, color='r', linewidth=1) # keyword args
cosLine.set_marker('p') # pentagon marker
cosLine.set_markerfacecolor((0,1,0)) # green in RGB model
cosLine.set_markersize(6)
plt.setp(cosLine, 'color', 'b', 'linestyle', 'solid')
plt.show()
```



Matplotlib

```
# simple Gaussian histogram
import numpy as np
import matplotlib.pyplot as plt

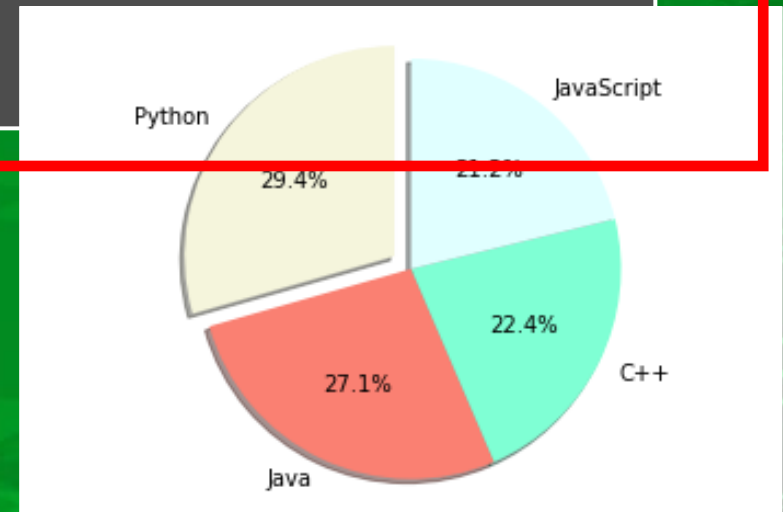
data = np.random.normal(size=1000)
plt.figure(figsize=(6, 4))
plt.hist(data)
plt.title('Gaussian Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



Matplotlib

```
# simple pie chart
import matplotlib.pyplot as plt

labels = 'Python', 'Java', 'C++', 'JavaScript'
data = [250, 230, 190, 180]
colors = ['beige', 'salmon', 'aquamarine', 'lightcyan']
explode = (0.1, 0, 0, 0)      # explode 1st slice (Python)
plt.pie(data, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=90)
plt.axis('equal')
plt.show()
```



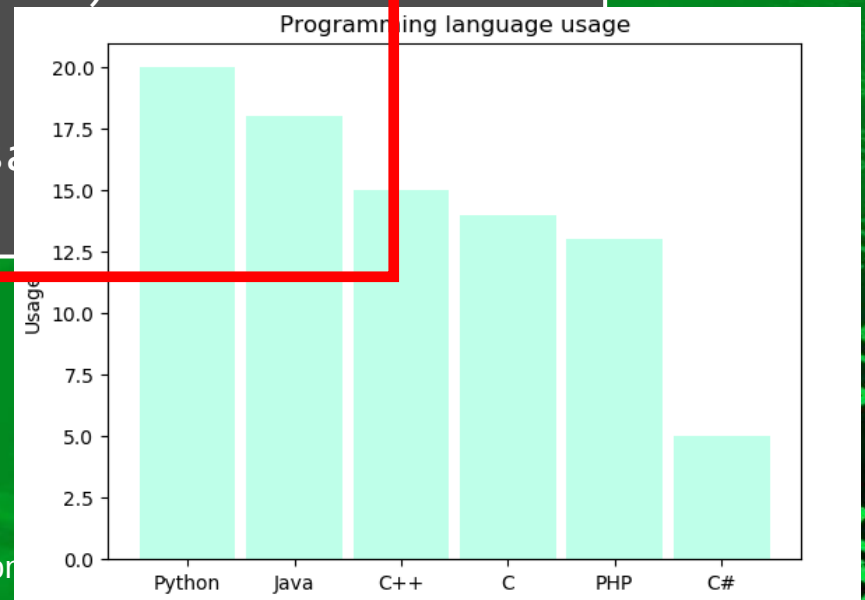
Matplotlib

```
# bars chart
import numpy as np
import matplotlib.pyplot as plt

languages = ('Python', 'Java', 'C++', 'C', 'PHP', 'C#')
x = np.arange(len(languages))
height = [20, 18, 15, 14, 13, 5]

plt.bar(x, height, width=0.9, align='center',
        color='aquamarine', alpha=0.5)

plt.xticks(x, languages)
plt.ylabel('Usage')
plt.title('Programming language usage')
plt.show()
```



Python

Matplotlib

Η βιβλιοθήκη Matplotlib σχεδιάσθηκε για δισδιάστατα (2-Δ) διαγράμματα. Πολύ σύντομα, όμως, άρχισε η προσθήκη κάποιων βασικών λειτουργιών σχεδίασης τρισδιάστατων (3-Δ) γραφημάτων. Όλες οι 3-Δ λειτουργίες βρίσκονται στο πακέτο **mplot3d** της βιβλιοθήκης **mpl_toolkits**, οπότε στα προγράμματα απαιτείται και η εισαγωγή του με μια εντολή όπως την επόμενη:

```
from mpl_toolkits import mplot3d
```

Αρχικά δημιουργούμε ένα νέο γράφημα με την εντολή **pyplot.axes**:

```
axes(projection='3d')
```

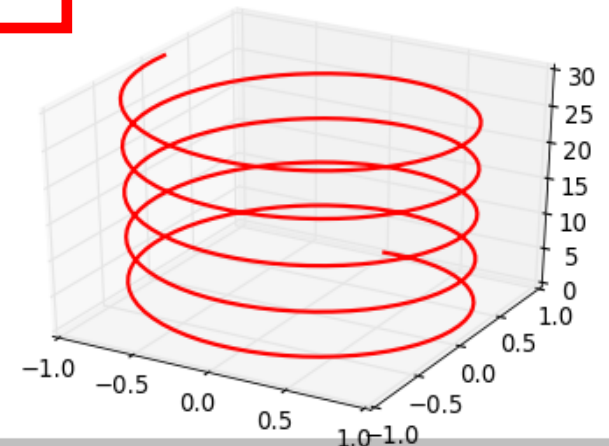
Η επιλογή **projection='3d'** δημιουργεί τρισδιάστατο γράφημα. Στη συνέχεια μπορούμε να σχεδιάσουμε τον επιθυμητό τύπο διαγράμματος.

Το πρόγραμμα που ακολουθεί σχεδιάζει μία καμπύλη στον χώρο με χρήση της εντολής **plot3D**:

Matplotlib

```
# 3D line plot
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits import mplot3d

plt.figure(figsize=(6, 4))
ax = plt.axes(projection='3d')
Z = np.linspace(0, 30, 300)
X = np.sin(Z); Y = np.cos(Z)
ax.plot3D(X, Y, Z, 'red', linewidth=2)
plt.show()
```



Matplotlib

Στο επόμενο παράδειγμα θα σχεδιάσουμε μια επιφάνεια και θα τροποποιήσουμε κάποιες από τις βασικές της ιδιότητες:

```
# 3D surface plot
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits import mplot3d
from matplotlib import cm

plt.figure(figsize=(7, 4))
ax = plt.axes(projection='3d')
x = np.arange(-2, 2, 0.1)
y = np.arange(-2, 2, 0.1)
X, Y = np.meshgrid(x, y)
Z = X * np.exp(-X**2 - Y**2);

surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                      cmap=cm.coolwarm, linewidth=0)
ax.set_xlabel('X'); ax.set_ylabel('Y')
```

Matplotlib

```
ax.tick_params(axis='both', colors='b', labels=7)  
cb = plt.colorbar(surf, shrink=0.7, aspect=20)  
cb.ax.tick_params(labels=7)  
plt.show()
```

Η εντολή `plot_surface` δημιουργεί το γράφημα επιφάνειας για τη συνάρτηση $z = x \cdot \exp(-x^2 - y^2)$.

Οι επιλογές **rstride** και **cstride** καθορίζουν το βήμα της γραμμής και της στήλης αντίστοιχα, που χρησιμοποιείται στη δειγματοληψία των δεδομένων. Η προκαθορισμένη τους τιμή είναι 10.

Η εντολή `colorbar` εμφανίζει στα δεξιά τη στήλη με την αντιστοιχία χρωμάτων και τιμών.

