

**ΗΜΥ01Κ06**  
Επιστημονικός Προγραμματισμός με Python



Διάλεξη Δέκατη Τρίτη  
Βασικές Βιβλιοθήκες της Python

*Φθινόπωρο 2025*

# Βασικές Βιβλιοθήκες της Python

**Python:** Προϊόν ελεύθερου και ανεξάρτητου λογισμικού (*open source*)

- Δωρεάν διάθεση, συχνές *αναβαθμίσεις*
- Τεράστια (*και πολύ ενεργή*) κοινότητα υποστήριξης
- Εκτεταμένη χρήση σε όλο το φάσμα των *υπολογιστικών εφαρμογών*
- Τεράστιος αριθμός από *βιβλιοθήκες συναρτήσεων* για κάθε χρήση
  - τουλάχιστον *200* που θεωρούνται standard (οικουμενικής αποδοχής)
  - πάνω από *70.000* συνολικά σύμφωνα με πρόσφατη καταμέτρηση
  - οι αριθμοί αυτοί αυξάνουν συνεχώς

# Μερικές από τις περιοχές εφαρμογών της Python

- Επιστημονικός προγραμματισμός / Αριθμητική ανάλυση  
*(Scientific /Numeric Computing)*
- Ανάπτυξη διεπαφών χρήστη-υπολογιστή (GUI)  
*(GUI Programming)*
- Ανάπτυξη εφαρμογών Web  
*(Web Application Development)*
- Επιστήμη Δεδομένων και Μηχανική Μάθηση  
*(Data Science & Machine Learning)*
- Επεξεργασία και ανάλυση δεδομένων  
*Data manipulation and analysis*
- Ανάπτυξη παιχνιδιών  
*(Game programming)*

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

|   |   |
|---|---|
| Επιστημονικός προγραμματισμός<br>/ Αριθμητική ανάλυση<br><i>Scientific /Numeric Computing</i> | <i>NumPy, SciPy, Matplotlib</i>         |
| Ανάπτυξη διεπαφών χρήστη-υπολογιστή<br><i>GUI Programming</i>                                 | <i>Tkinter, wxPython</i>                |
| Συγκομιδή περιεχομένου ιστοσελίδων<br><i>Web Scraping</i>                                     | <i>BeautifulSoup, Requests, Urllib3</i> |
| Ανάπτυξη εφαρμογών Web<br><i>Web Application Development</i>                                  | <i>Flask, Django</i>                    |
| Επιστήμη Δεδομένων & Μηχανική Μάθηση<br><i>Data Science &amp; Machine Learning</i>            | <i>scikit-learn, Tensorflow , Keras</i> |
| Επεξεργασία και ανάλυση δεδομένων<br><i>Data manipulation and analysis</i>                    | <i>pandas</i>                           |
| Ανάπτυξη παιχνιδιών<br><i>Game Programming</i>  | <i>PyGame, Pyglet , Panda3D</i>         |



# NumPy

- Αποτελεί την *βασικότερη βιβλιοθήκη της Python* για αριθμητική ανάλυση
- Εξέλιξη των βιβλιοθηκών *Numeric* και *Numarray*, τις οποίες αντικατέστησε το *2006*
- Προσθέτει υποστήριξη για *πολυδιάστατα arrays* / πίνακες παρέχοντας τις κατάλληλες δομές δεδομένων και τις συναρτήσεις/πράξεις πάνω σε αυτές
- Σε ένα βαθμό προσομοιάζει με το *MATLAB* (που είναι και αυτό *interpreted*)
- Βασική δομή της NumPy είναι το *n-διάστατο array (ndarray)*, το οποίο είναι βασικά ένας τρόπος θεώρησης και χειρισμού της μνήμης σε ισομεγέθεις "*δρασκελιές*" (*strided memory view*)
- Σε αντίθεση με τις λίστες της Python (*που εν γένει είναι ανομοιογενείς*) τα *arrays είναι ομοιογενή* (*όλα τα στοιχεία τους είναι του ίδιου τύπου και έχουν ίδιο μέγεθος*). Αυτός είναι και ο λόγος που μπορεί να εφαρμοστεί ο παραπάνω τρόπος θεώρησης μνήμης (*strided*).
- Η *διαλειτουργικότητα* της NumPy και του *MATLAB* επιτυγχάνεται προσθέτοντας δυο ακόμα επιστημονικές βιβλιοθήκες (*SciPy* και *Matplotlib*) οι οποίες εγκαθίστανται "πάνω" στην NumPy και την *επεκτείνουν* τις δυνατότητές της

# Παραδείγματα κώδικα NumPy

## Δημιουργία arrays

```
import numpy as np
x = np.array([1,2,3])
y= np.arange(10) #όπως το range της Python, όμως επιστρέφει array
print(x, '\n', y)

[1 2 3]
[0 1 2 3 4 5 6 7 8 9]
```

## Βασικές πράξεις

```
a = np.array([1,2,3,4,5])
b = np.linspace(0,2,5) #Δημιούργησε ένα array με 5 ισαπέχοντα σημεία από το 0 έως το 2
c=a-b
print (a, '\n', a**2, '\n', b, '\n', c)

[1 2 3 4 5]
[ 1  4  9 16 25]
[0.  0.5  1.  1.5  2. ]
[1.  1.5  2.  2.5  3. ]
```

# Παραδείγματα κώδικα NumPy

## Βασικές συναρτήσεις

```
a = np.linspace(-np.pi, np.pi, 100) # Δημιούργησε ένα array με 100 ισαπέχοντα σημεία από  $-\pi$  έως  $\pi$   
b = np.sin(a) # Δημιούργησε ένα array με τις τιμές του ημιτόνου για τα παραπάνω σημεία  
c = np.cos(a) # Δημιούργησε ένα array με τις τιμές του συνημιτόνου για τα παραπάνω σημεία
```

## Γραμμική άλγεβρα

```
from numpy.linalg import solve, inv  
a = np.array([[1,2,3], [4,-5,6], [7,8,-9]]) # Δισδιάστατο array 3x3  
print('\n', a, '\n\n', inv(a)) # inv(a): αντίστροφος πίνακας του a  
b = np.array([1,2,3]) # Μονοδιάστατο array 3 θέσεων  
x= solve(a,b) # επίλυση του συστήματος ax=b  
print('\n',x)  
[[ 1  2  3]  
 [ 4 -5  6]  
 [ 7  8 -9]]  
  
[[-0.00847458  0.11864407  0.07627119]  
 [ 0.22033898 -0.08474576  0.01694915]  
 [ 0.18926554  0.01694915 -0.03672316]]  
  
[0.45762712 0.10169492 0.11299435]
```

# Παραδείγματα κώδικα NumPy

Γραμμική άλγεβρα (συνέχεια)

```
from numpy.random import rand
c= rand(3,3) * 20 # δημιούργησε έναν 3x3 πίνακα με τυχαίους αριθμούς από 0-1, πολλαπλασιασμένους επί 20
print(' \n', c)
print(' \n', np.dot(a,c)) # πολλαπλασιασμός πινάκων (ο πίνακας a, προέρχεται από το προηγούμενο παράδειγμα)
print(' \n', a @ c) #ισοδύναμος τελεστής πολλαπλασιασμού πινάκων (μετά την Python 3.5 και την NumPy 1.10)
```

```
[[13.48822205  5.68408188  2.24278296]
 [ 8.03450609 14.07614439  5.23257304]
 [ 3.30389294 16.83351101  8.43775437]]
```

```
[[ 39.46891305  84.33690369  38.02119214]
 [ 33.60371543  53.35667167  33.43479287]
 [128.95856653   0.8961291  -18.37972426]]
```

```
[[ 39.46891305  84.33690369  38.02119214]
 [ 33.60371543  53.35667167  33.43479287]
 [128.95856653   0.8961291  -18.37972426]]
```





- Αποτελεί την *βασικότερη βιβλιοθήκη της Python* για επιστημονικό προγραμματισμό
- Είναι χτισμένη πάνω στην *NumPy* από την οποία και *εξαρτάται* καθώς τα *πολυδιάστατα arrays* αποτελούν την θεμελιώδη δομή της SciPy.
- Περιέχει συναρτήσεις που υποστηρίζουν την επίλυση σειράς κατηγοριών προβλημάτων στην επιστήμη και την μηχανική όπως ενδεικτικά:
  - Βελτιστοποίηση (*optimization*)
  - Γραμμική άλγεβρα
  - Ολοκλήρωση (*Integration*)
  - Παρεμβολή /προσέγγιση (*Integration / approximation*)
  - Fast Fourier Transform (*FFT*)
  - Επεξεργασία σήματος και εικόνας (*signal & image processing*)
  - Επίλυση κανονικών διαφορικών εξισώσεων (*Ordinary Differential Equations*)
- Αποτελείται από έναν πυρήνα και σειρά από αλληλεξαρτώμενες ενότητες / πακέτα (modules) τα οποία εγκαθίστανται ταυτόχρονα.

# Διαθέσιμα πακέτα (modules) της SciPy

| Module             | Description  |
|--------------------|--|
| <i>cluster</i>     | Clustering algorithms                                      |
| <i>constants</i>   | Physical and mathematical constants                        |
| <i>fftpack</i>     | Fast Fourier Transform routines                            |
| <i>integrate</i>   | Integration and ordinary differential equation solvers     |
| <i>interpolate</i> | Interpolation and smoothing splines                        |
| <i>io</i>          | Input and Output   |
| <i>linalg</i>      | Linear algebra Περιέχει συναρτήσεις                        |
| <i>ndimage</i>     | N-dimensional image processing                             |
| <i>odr</i>         | Orthogonal distance regression                             |
| <i>optimize</i>    | Optimization and root-finding routines                     |
| <i>signal</i>      | Signal processing  |
| <i>sparse</i>      | Sparse matrices and associated routines                    |
| <i>spatial</i>     | Spatial data structures and algorithms                     |
| <i>special</i>     | Special functions such as Bessel, Gamma, Erf, Elliptic etc |
| <i>stats</i>       | Statistical distributions and functions                    |



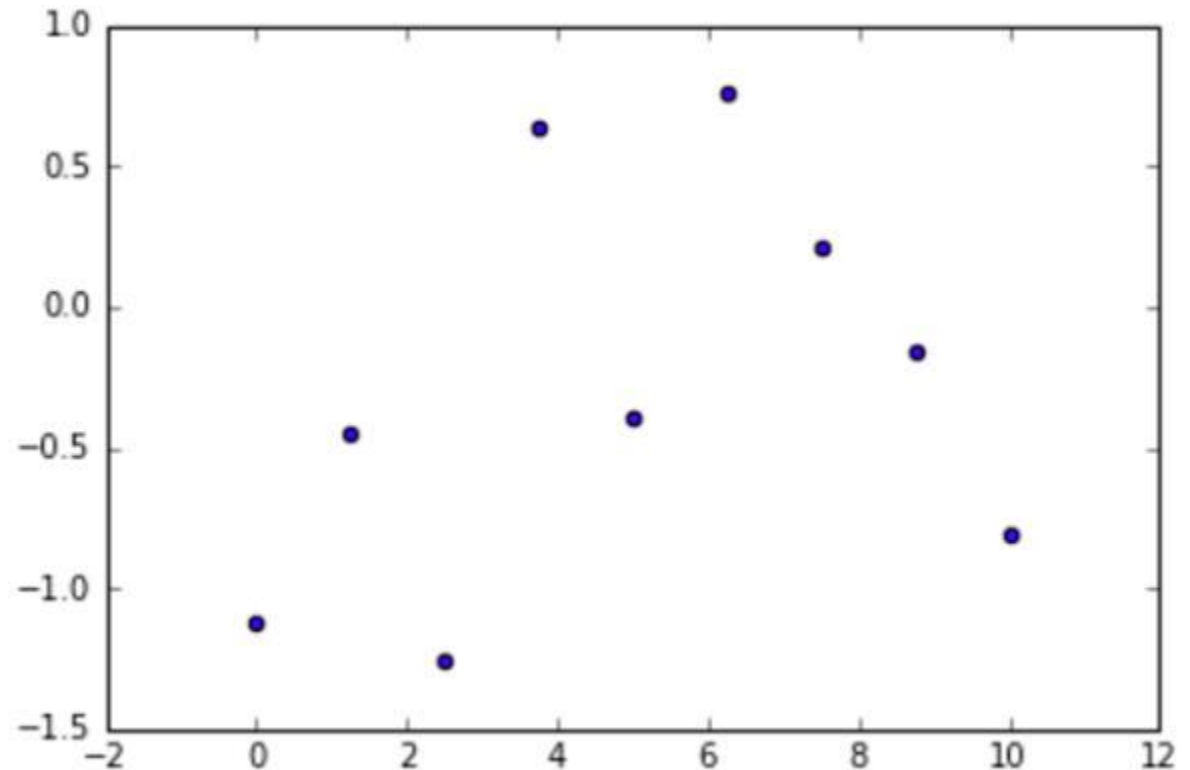
# matplotlib

- Βιβλιοθήκη *δημιουργίας γραφημάτων* (plots) της Python, εμπνευσμένη από το *MATLAB* οι όροι που χρησιμοποιεί (*Axis, Figure, Plots*) είναι παρόμοιοι με εκείνους του *MATLAB*
- Αποτελεί *φυσική επέκταση* της βιβλιοθήκης *NumPy* πάνω στην οποία είναι χτισμένη
- Καθώς χρησιμοποιείται ευρύτατα και από την βιβλιοθήκη *SciPy*, οι τρεις βιβλιοθήκες *NumPy*, *SciPy* και *matplotlib* εγκαθίστανται συνήθως μαζί και αντιμετωπίζονται ως μια αδιάσπαστη ενότητα
- Διαθέτει ειδικό κέλυφος αλληλεπίδρασης ονόματι *Pyplot* το οποίο προσομοιάζει την λειτουργικότητα του *MATLAB* (για άτομα που είναι εξοικειωμένα με την χρήση του *MATLAB*)
- Διαθέτει επίσης το δομοστοιχείο *Pylab* (κατά το *Matlab*) το οποίο συνδυάζει το κέλυφος αλληλεπίδρασης *Pyplot* και την βιβλιοθήκη *NumPy* σε *ενιαίο ονοματοχώρο*, με στόχο την κατά το δυνατόν εξομοίωση της *υποκατάστασης* του *MATLAB* εξ, ολοκλήρου από την Python.

# Pyplot

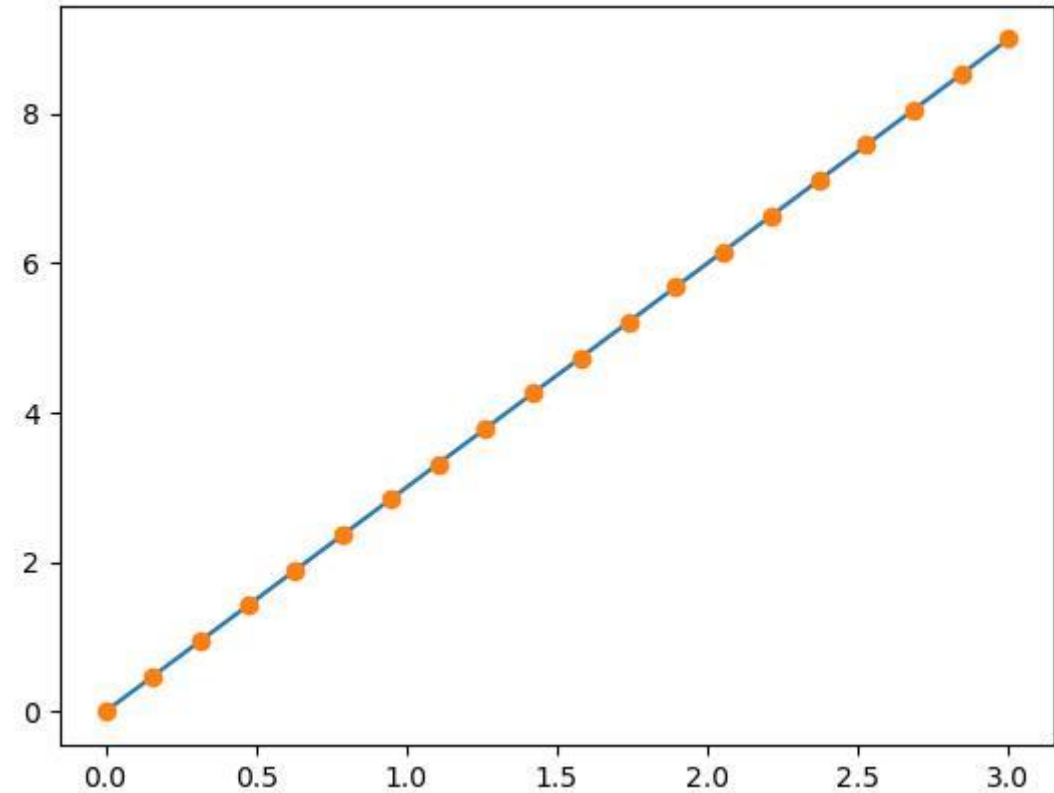
Ειδικό πακέτο-κέλυφος αλληλεπίδρασης (shell-like interface module) της βιβλιοθήκης matplotlib το οποίο προσφέρει παρόμοια λειτουργικότητα με εκείνη του MATLAB.

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = np.linspace(0,10,9)
>>> y = np.randn(9)
>>> plt.scatter(x,y)
>>> plt.show()
```



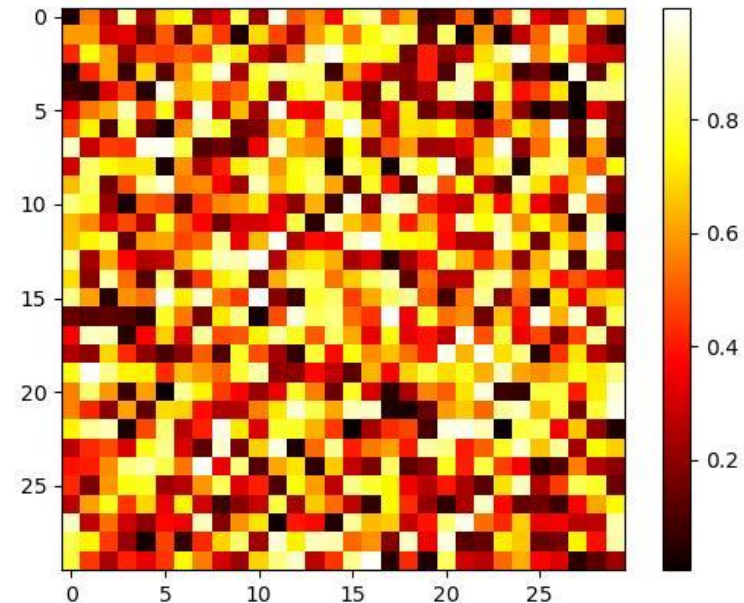
# Pyplot - Basic 1D Figure

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = np.linspace(0, 3, 20)
>>> y = np.linspace(0, 9, 20)
>>> plt.plot(x,y)
>>> plt.show()
```

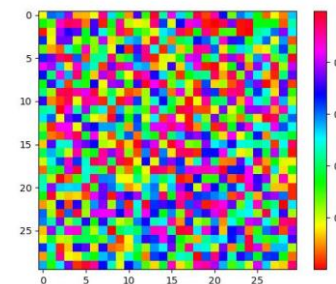


# Pyplot Basic 2D Plot

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> image = np.random.rand(30, 30)
>>> plt.imshow(image, cmap=plt.cm.hot)
>>> plt.colorbar()
>>> plt.show()
```



```
>>> plt.imshow(image, cmap=plt.cm.hsv)
>>> plt.colorbar()
>>> plt.show()
```



# Pyplot – Data Fitting Example

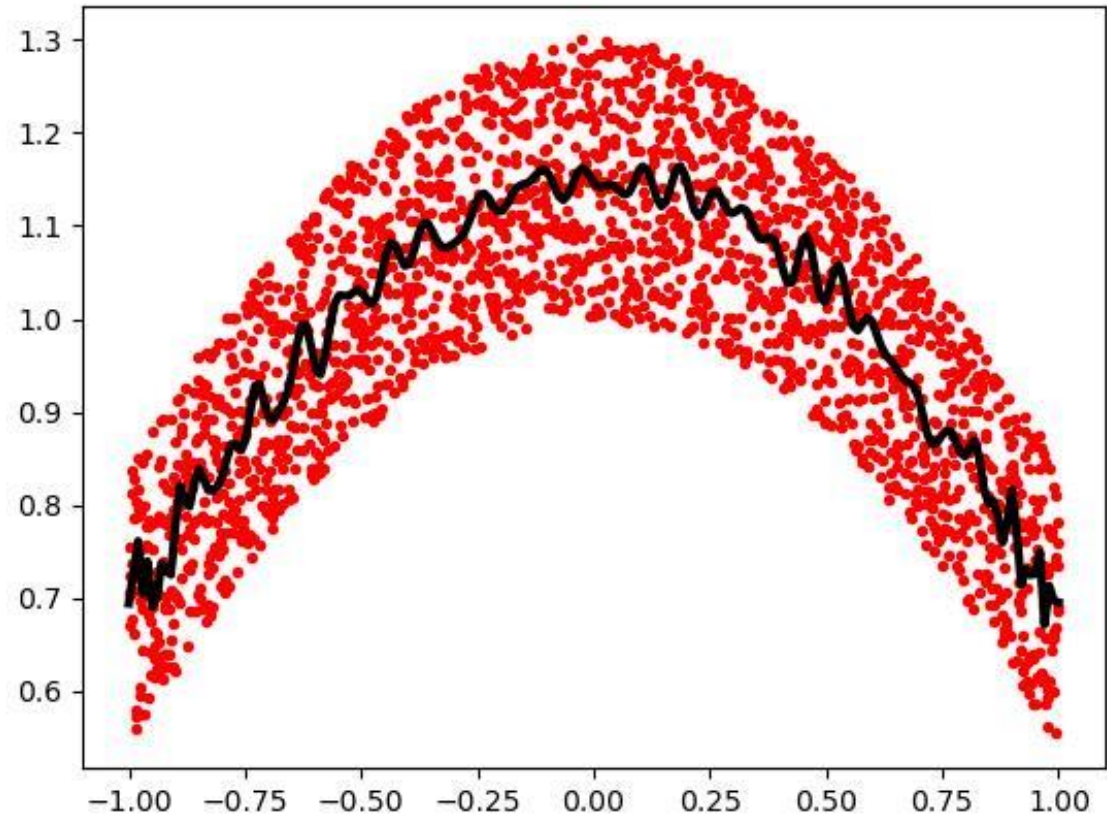
```
#Plot noisy data and their polynomial fit in a Chebyshev basis
```

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

np.random.seed(0)

x = np.linspace(-1, 1, 2000)
y = np.cos(x) + 0.3*np.random.rand(2000)
p = np.polynomial.Chebyshev.fit(x, y, 90)

t = np.linspace(-1, 1, 200)
plt.plot(x, y, 'r.')
plt.plot(t, p(t), 'k-', lw=3)
plt.show()
```



# Pyplot - Read and write images

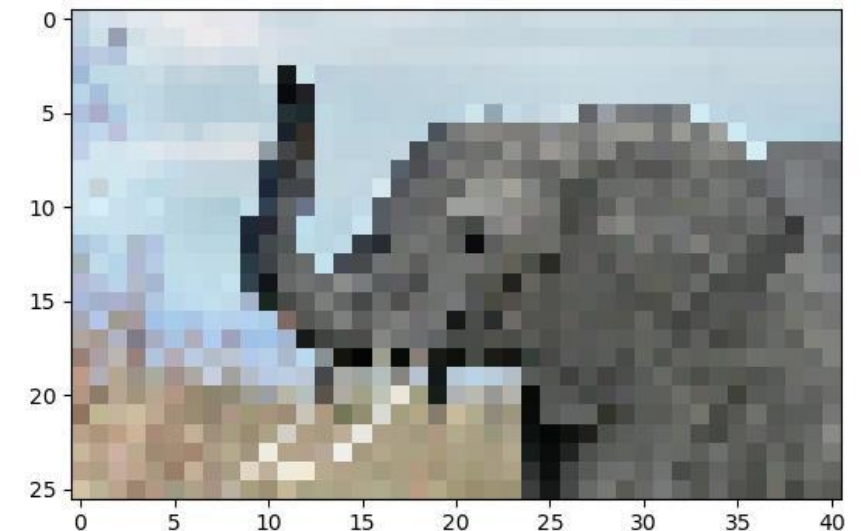
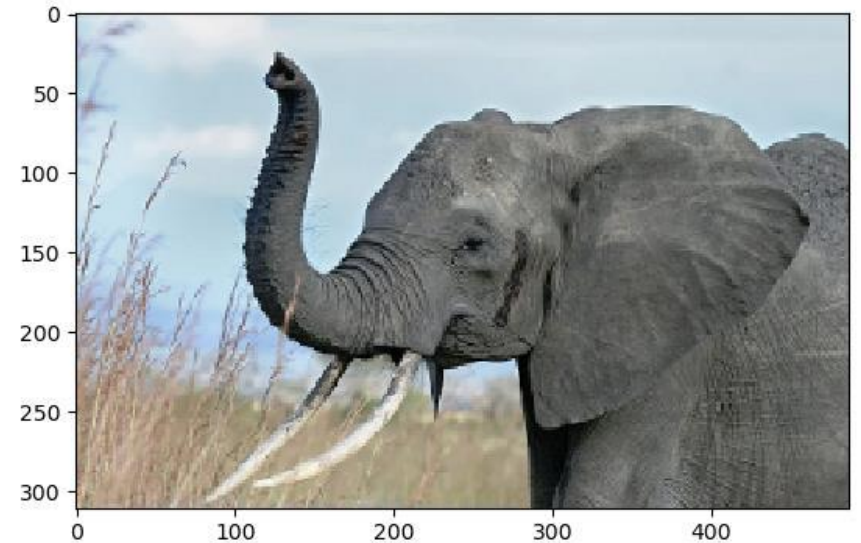
```
#Read and write images

import numpy as np
import matplotlib.pyplot as plt

# original figure
plt.figure()
img = plt.imread('elephant.png')
plt.imshow(img)

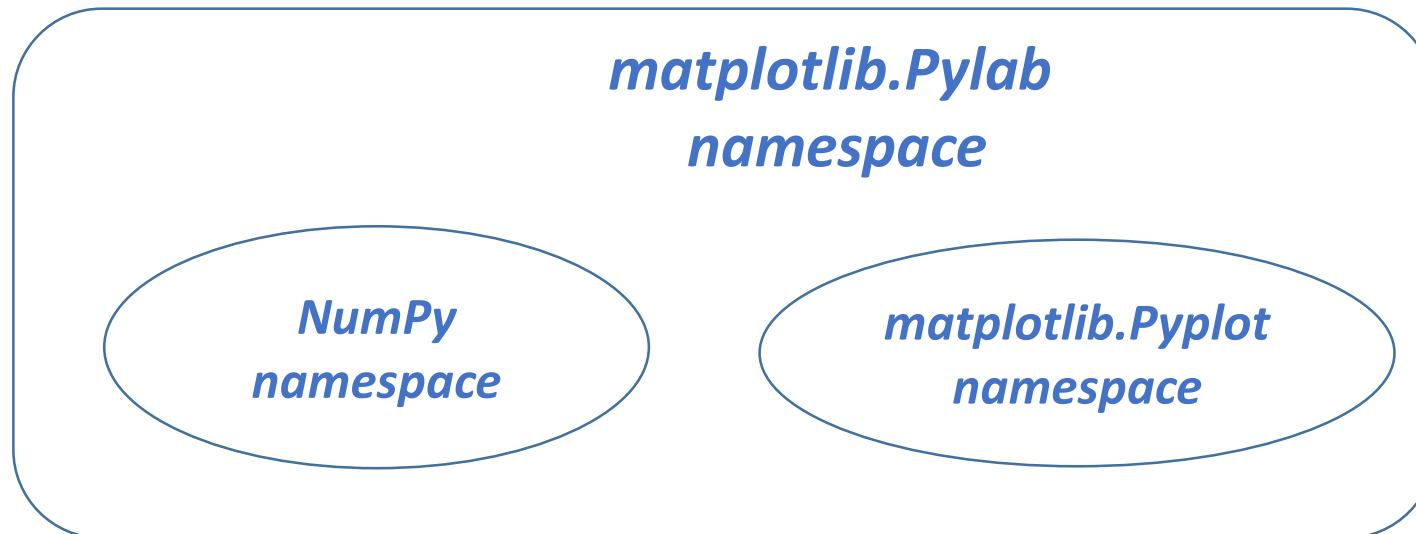
# lower resolution
plt.figure()
img_tiny = img[::12, ::12]
plt.imshow(img_tiny, interpolation='nearest')

plt.show()
```



# PyLab

- Δομοστοιχείο της matplotlib το *οποίο δεν προσθέτει νέες λειτουργικότητες*, απλά διευκολύνει την χρήση του Pyplot.
- Προέκυψε από την ιδέα συνδυασμού των ονοματοχώρων (namespaces) της βιβλιοθήκης *NumPy* και του δομοστοιχείου *Pyplot* της βιβλιοθήκης *matplotlib* και μερικών ακόμα στοιχείων σε ένα κοινό ονοματοχώρο.



- Η χρήση του PyLab θεωρείται κακή πρακτική και δεν ενθαρρύνεται πλέον.

<https://matplotlib.org/api/index.html?highlight=pylab#the-pylab-api-disapproved>

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

Επιστημονικός προγραμματισμός  
/ Αριθμητική ανάλυση  
*Scientific /Numeric Computing*

*NumPy, SciPy, Matplotlib*

Ανάπτυξη διεπαφών χρήστη-υπολογιστή  
*GUI Programming*

*Tkinter, wxPython*

Συγκομιδή περιεχομένου ιστοσελίδων  
*Web Scraping*

*BeautifulSoup, Requests, Urllib3*

Ανάπτυξη εφαρμογών Web  
*Web Application Development*

*Flask, Django*

Επιστήμη Δεδομένων & Μηχανική Μάθηση  
*Data Science & Machine Learning*

*scikit-learn, Tensorflow , Keras*

Επεξεργασία και ανάλυση δεδομένων  
*Data manipulation and analysis*

*pandas*

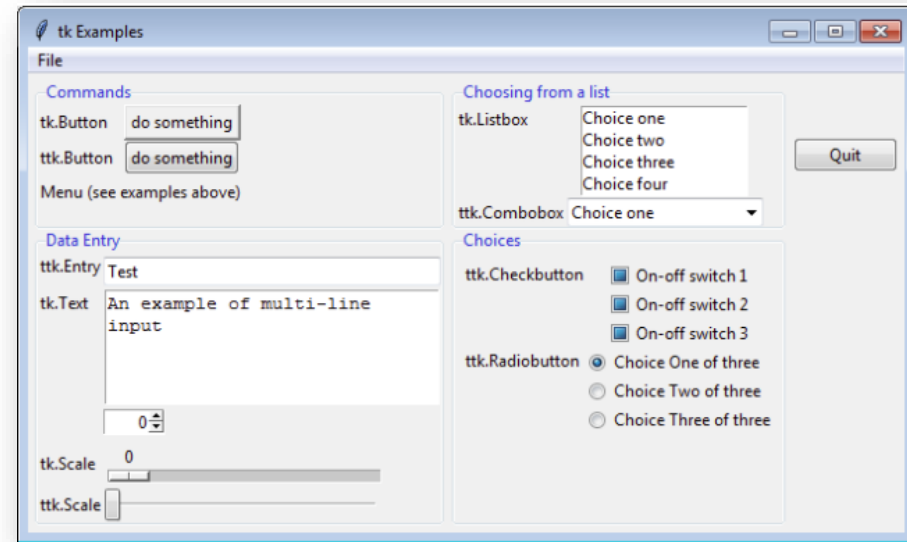
Ανάπτυξη παιχνιδιών  
*Game Programming*

*PyGame, Pyglet , Panda3D*

# Tkinter

- Αποτελεί την *κατ' εξοχήν στάνταρντ βιβλιοθήκη* για δημιουργία εφαρμογών *γραφικής επικοινωνίας* (GUI) σε Python

- Μέρος του στάνταρντ πακέτου που εγκαθίσταται σε όλες τις εκδόσεις της Python για περιβάλλοντα Linux, Windows και Mac OS X



- Περιέχει μια *ποικιλία από στοιχεία GUI* (*widgets*) όπως *ετικέτες*, *πλαίσια κειμένου*, *λίστες*, *πλήκτρα* κλπ. τα οποία μπορούν να τοποθετηθούν σε *επιλεγμένα σημεία ενός "καμβά"* ή ενός πλέγματος μέσα σ' ένα παράθυρο

# Tkinter

Για *κάθε ένα* από αυτά τα *στοιχεία* που τοποθετούμε πάνω στον *καμβά* ή το πλέγμα μπορούμε να *τροποποιήσουμε*

- την εμφάνισή του (θέση, μέγεθος, τύπος και χρώμα φόντου, τύπος, μέγεθος και χρώμα κειμένου κλπ.)
- το περιεχόμενό του (κείμενο , εικόνα κλπ.)
- την συμπεριφορά του για διάφορες καταστάσεις (*αλλαγή περιεχομένου* ή άλλων ιδιοτήτων του, επιλογή με *κλικ*, *διπλό-κλικ* του ποντικιού, πέρασμα ή *σύρσιμο* (drag) του ποντικιού από πάνω του κλπ.)



# Tkinter widgets 1/2

- *Button*: used to display buttons in application
- *Canvas*: used to draw shapes, such as lines, ovals, polygons and rectangles, in application
- *Checkbutton*: used to display a number of options as checkboxes. The user can select multiple options at a time
- *Entry*: used to display a single-line text field for accepting values from a user
- *Frame*: used as a container widget to organize other widgets
- *Label*: used to provide a single-line caption for other widgets. It can also contain images
- *Listbox*: used to provide a list of options to a user
- *Menubutton*: used to display menus in application
- *Menu*: used to provide various commands to a user. These commands are contained inside Menubutton
- *Message*: used to display multiline text fields for accepting values from a user

# Tkinter widgets 2/2

- *Radiobutton*: used to display a number of options as radio buttons. The user can select only one option at a time
- *Scale*: used to provide a slider widget
- *Scrollbar*: used to add scrolling capability to various widgets, such as list boxes
- *Text*: used to display text in multiple lines
- *Toplevel*: used to provide a separate window container
- *Spinbox*: variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values
- *PanedWindow*: a container widget that may contain any number of panes, arranged horizontally or vertically
- *LabelFrame*: simple container widget to act as a spacer or container for complex window layouts
- *tkMessageBox*: used to display message boxes in applications

# Παράδειγμα απλής εφαρμογής σε Tkinter

```
from tkinter import *
from tkinter import messagebox
from tkinter import simpledialog

window = Tk()

window.title("Το πρώτο μου πρόγραμμα Python σε Windows")

window.geometry('600x400')

lbl = Label(window, text="Hello")
lbl.place(relx=0.5, rely=0.5, relwidth=0.2, anchor="center")

txt = Entry(window,width=30, fg="blue", bg="yellow" )
txt.place(relx=0.5, rely=0.7, relwidth=0.2, anchor="center")

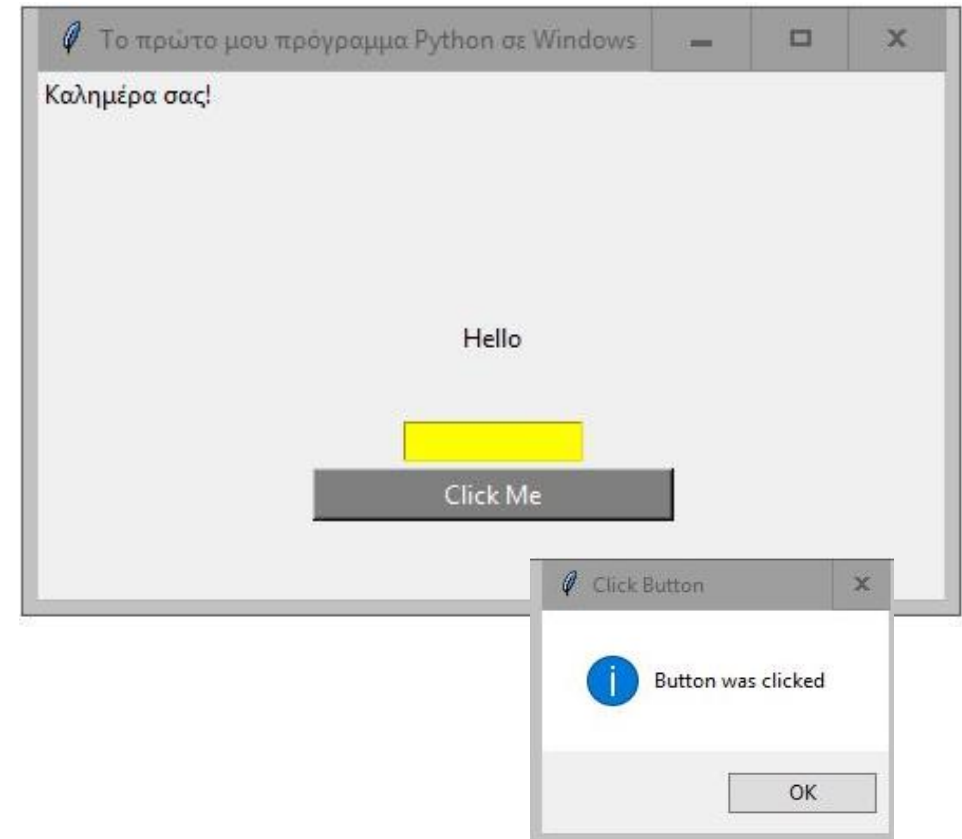
l1=Label(window,text="Καλημέρα σας!")
l1.grid(column=1)

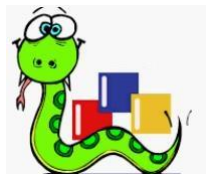
def clicked():

    lbl.configure(text="Button was clicked !!")
    info = messagebox.showinfo('Click Button', 'Button was clicked', parent=window)

btn = Button(window, text="Click Me", command=clicked, bg="gray50", fg="white", relief=RAISED)
btn.place(relx=0.5, rely=0.8, anchor="center", relwidth=0.4, relheight=0.1)

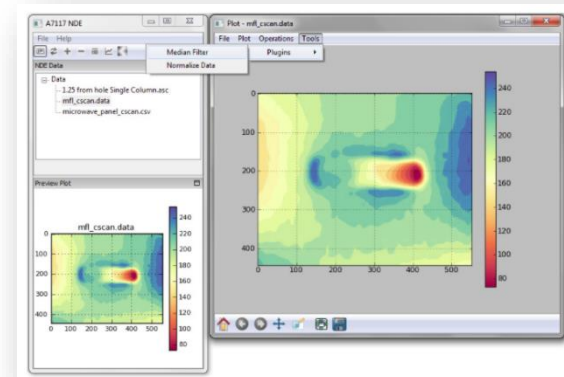
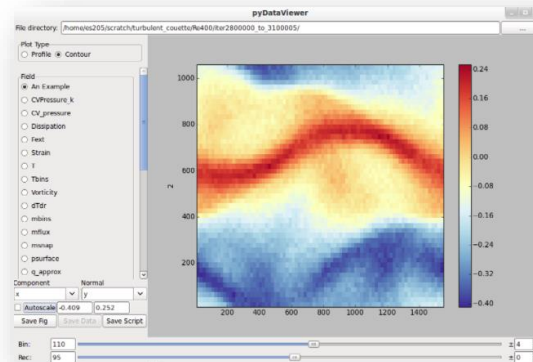
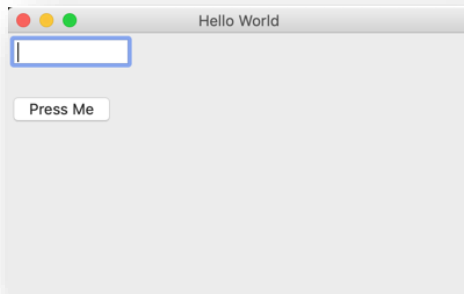
window.mainloop()
```

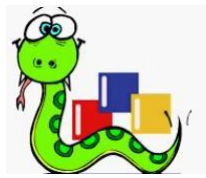




# wxPython

- Εναλλακτική βιβλιοθήκη για δημιουργία εφαρμογών *γραφικής επικοινωνίας* (GUI) σε Python βασισμένη στην εργαλειοθήκη *wxwidgets* της C++
- Προσφέρει περισσότερη *ευελιξία* και *επιλογές* από την Tkinter, όμως είναι πιο *απαιτητική* στην *εκμάθηση*
- Αποτελεί πρώτη επιλογή για προγραμματιστές που είναι ήδη εξοικειωμένοι με την εργαλειοθήκη *wxwidgets* της C++
- Η *μορφή* που έχουν τα *παράθυρα* και τα *widgets* της wxPython ακολουθεί την "*αισθητική*" του *λειτουργικού συστήματος* που τρέχει το πρόγραμμα (*Windows macOS, Linux*), σε αντίθεση με την *Tkinter*, όπου *όλες οι εφαρμογές έχουν την ίδια τυποποιημένη γραφική εικόνα* όπου και αν τρέξουν





# wxPython

Παράδειγμα δημιουργίας ενός απλού παράθυρου με wxPython

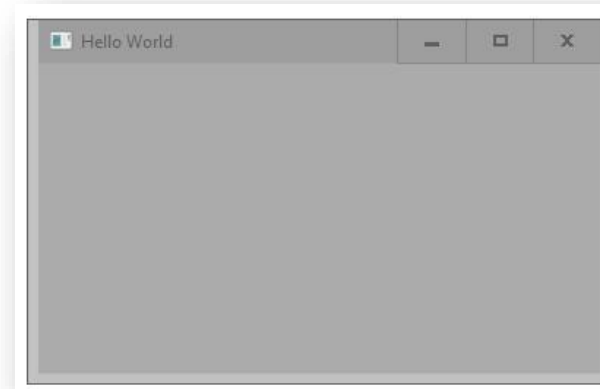
```
import wx

# Next, create an application object.
app = wx.App()

# Then a frame.
frm = wx.Frame(None, title="Hello World")

# Show it.
frm.Show()

# Start the event loop.
app.MainLoop()
```



Linux



MacOS

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

Επιστημονικός προγραμματισμός  
/ Αριθμητική ανάλυση  
*Scientific /Numeric Computing*

*NumPy, SciPy, Matplotlib*

Ανάπτυξη διεπαφών χρήστη-υπολογιστή  
*GUI Programming*

*Tkinter, wxPython*

Συγκομιδή περιεχομένου ιστοσελίδων  
*Web Scraping*

*BeautifulSoup, Requests, Urllib3*

Ανάπτυξη εφαρμογών Web  
*Web Application Development*

*Flask, Django*

Επιστήμη Δεδομένων & Μηχανική Μάθηση  
*Data Science & Machine Learning*

*scikit-learn, Tensorflow , Keras*

Επεξεργασία και ανάλυση δεδομένων  
*Data manipulation and analysis*

*pandas*

Ανάπτυξη παιχνιδιών  
*Game Programming*

*PyGame, Pyglet , Panda3D*



# Requests

- Βιβλιοθήκη συναρτήσεων για αποστολή αιτήσεων HTTP σε διακομιστές, με εξαιρετικά απλό και φιλικό τρόπο
- Χρησιμοποιείται στην θέση των built-ins βιβλιοθηκών της Python [urllib](#) και [urllib2](#), οι οποίες παρέχουν τις βασικές συναρτήσεις πρόσβασης σε σελίδες διακομιστών.

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type":"User"...'
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

*JSON: Java Script Object Notation Μεταγλώσσα για αναπαράσταση δομημένων δεδομένων που βασίζεται στην JavaScript*

# BeautifulSoup



- Φτιαγμένη αποκλειστικά για "συγκομιδή" ιστοσελίδων web scraping
- Μπορεί να εξάγει δεδομένα από περιεχόμενο html είτε XML
- Το περιεχόμενο αποθηκεύεται σε *δομημένη μορφή* (δεντρική) ώστε να παρέχει δυνατότητες εύρεσης και επιλογής συγκεκριμένων στοιχείων της δομής (κλάσεις, σύνδεσμοι, ιδιότητες, απλό κείμενο κλπ.) με βάση κριτήρια του χρήστη

## Παράδειγμα χρήσης

```
import requests
```

```
from bs4 import BeautifulSoup
```

```
html_page = requests.get('https://ece.hmu.gr/news_gr/').text
```

```
soup = BeautifulSoup(html_page, 'html.parser')
```

Το soup είναι ένα αντικείμενο τύπου BeautifulSoup το οποίο περιέχει *με δομημένο τρόπο* όλο το *περιεχόμενο της ιστοσελίδας* για εξαγωγή κάθε είδους πληροφορίας (κείμενο, τμήματα κώδικα, σύνδεσμοι κλπ.)

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

Επιστημονικός προγραμματισμός  
/ Αριθμητική ανάλυση  
*Scientific /Numeric Computing*

*NumPy, SciPy, Matplotlib*

Ανάπτυξη διεπαφών χρήστη-υπολογιστή  
*GUI Programming*

*Tkinter, wxPython*

Συγκομιδή περιεχομένου ιστοσελίδων  
*Web Scraping*

*BeautifulSoup, Requests, Urllib3*

Ανάπτυξη εφαρμογών Web  
*Web Application Development*

*Flask, Django*

Επιστήμη Δεδομένων & Μηχανική Μάθηση  
*Data Science & Machine Learning*

*scikit-learn, Tensorflow , Keras*

Επεξεργασία και ανάλυση δεδομένων  
*Data manipulation and analysis*

*pandas*

Ανάπτυξη παιχνιδιών  
*Game Programming*

*PyGame, Pyglet , Panda3D*



# Flask

- *micro*-περιβάλλον για την *ανάπτυξη διαδικτυακών εφαρμογών* με βάση τη γλώσσα Python βασισμένο σε ένα *απλό πυρήνα* (micro web framework) ο οποίος μπορεί να *επεκτείνεται* κατά τις ανάγκες της εφαρμογής (easy-to-extend philosophy)
- *micro* σημαίνει ότι *περιλαμβάνει τις βασικές λειτουργικότητες* και αφήνει πιο σύνθετες λειτουργίες όπως η αυθεντικοποίηση σε πρόσθετα (επεκτάσεις) σε αντίθεση με πιο σύνθετα περιβάλλοντα όπως το *Django* που διαθέτουν ένα *ολοκληρωμένο σύνολο λειτουργιών* χωρίς να απαιτούνται πρόσθετα

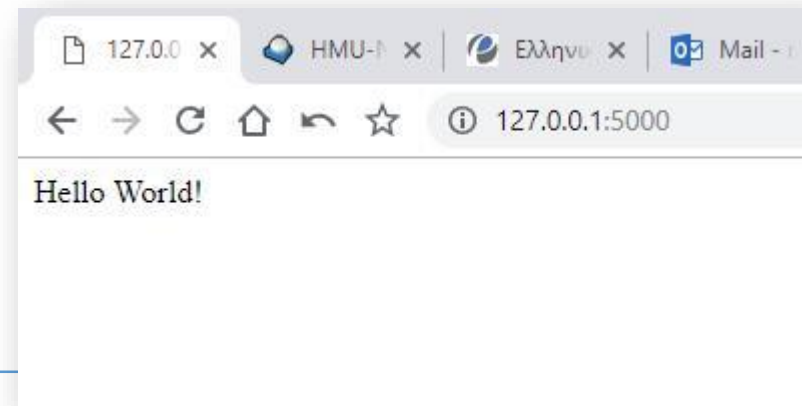
hello.py

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

```
python hello.py
Running on http://localhost:5000/
```



# Django

- Ολοκληρωμένο περιβάλλον ανάπτυξης *απαιτητικών διαδικτυακών εφαρμογών*
- Έχει πρωταρχικό στόχο την *διευκόλυνση δημιουργίας πολύπλοκων δικτυακών τόπων* που βασίζονται σε *βάσεις δεδομένων*.
- Δυνατά σημεία του η *επαναχρησιμοποίηση* και η "*δυνατότητα σύνδεσης*" (**pluggability**) των επί μέρους στοιχείων της εφαρμογής, η *ελαχιστοποίηση του κώδικα* και η *ταχεία ανάπτυξη πολύπλοκων εφαρμογών*
- Στον πυρήνα του περιλαμβάνει μεταξύ άλλων έναν *ελαφρύ και αυτόνομο web server* για την *δοκιμαστική λειτουργία των εφαρμογών που αναπτύσσονται* και ένα *σύστημα διεθνοποίησης*, (**internationalization**) που περιλαμβάνει *μεταφράσεις των συνιστωσών του Django* σε διάφορες γλώσσες
- Δέχεται σειρά επεκτάσεων, οι οποίες μεταξύ άλλων προσφέρουν
  - δυνατότητα *χειρισμού πολλαπλών websites*, καθεμιά με το δικό της *περιεχόμενο* και τις δικές της *εφαρμογές*
  - δυνατότητα δημιουργίας *συστημάτων αυθεντικοποίησης*
  - ένα πλαίσιο για τη δημιουργία *εφαρμογών GIS*
  - εργαλεία για τη δημιουργία *Google Sitemaps*

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

|   |   |
|---|---|
| Επιστημονικός προγραμματισμός<br>/ Αριθμητική ανάλυση<br><i>Scientific /Numeric Computing</i> | <i>NumPy, SciPy, Matplotlib</i>         |
| Ανάπτυξη διεπαφών χρήστη-υπολογιστή<br><i>GUI Programming</i>                                 | <i>Tkinter, wxPython</i>                |
| Συγκομιδή περιεχομένου ιστοσελίδων<br><i>Web Scraping</i>                                     | <i>BeautifulSoup, Requests, Urllib3</i> |
| Ανάπτυξη εφαρμογών Web<br><i>Web Application Development</i>                                  | <i>Flask, Django</i>                    |
| Επιστήμη Δεδομένων & Μηχανική Μάθηση<br><i>Data Science &amp; Machine Learning</i>            | <i>scikit-learn, Tensorflow , Keras</i> |
| Επεξεργασία και ανάλυση δεδομένων<br><i>Data manipulation and analysis</i>                    | <i>pandas</i>                           |
| Ανάπτυξη παιχνιδιών<br><i>Game Programming</i>  | <i>PyGame, Pyglet , Panda3D</i>         |



# Scikit-learn

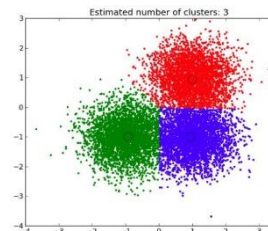
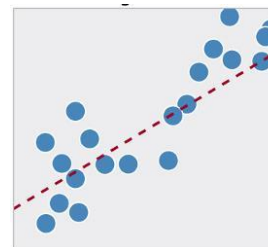
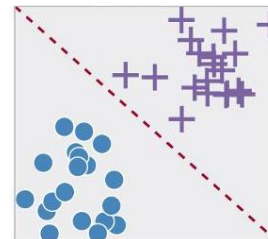
- Βιβλιοθήκη ανοιχτού κώδικα για *μηχανική μάθηση* (**machine learning**) .
- Παρέχει διάφορα εργαλεία για *προσαρμογή μοντέλου* (**model fitting**) *προεπεξεργασία δεδομένων* (**data preprocessing**), *επιλογή και αξιολόγηση μοντέλων* (**model selection and evaluation**) και πολλά άλλα βοηθητικά προγράμματα.
- Υποστηρίζει τόσο *επιβλεπόμενη* όσο και *μη επιβλεπόμενη μάθηση* (**supervised /unsupervised learning**).

## Μηχανική μάθηση

- Πεδίο της επιστήμης των υπολογιστών που δίνει στους υπολογιστές την ικανότητα *να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί*
- Διερευνά τη μελέτη και την κατασκευή *αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα* και να *κάνουν προβλέψεις* ή να *βγάλουν αποφάσεις* βασιζόμενες σε αυτά.
- Η μηχανική μάθηση μπορεί να είναι είτε *επιβλεπόμενη* (τα δεδομένα που δίνονται για εκπαίδευση των αλγορίθμων έχουν αναλυθεί από πριν και περιλαμβάνουν ενδείξεις για το τι αντιπροσωπεύουν) είτε *μη επιβλεπόμενη* (τα δεδομένα δεν περιλαμβάνουν ενδείξεις για το τι αντιπροσωπεύουν)
- Η *μηχανική μάθηση* και η *αναγνώριση προτύπων* μπορούν να θεωρηθούν ως "δυσόψεις του ίδιου νομίσματος"

# Βασικές εφαρμογές Scikit-learn

- Ταξινόμηση (Classification)
  - Προσδιορισμός της κατηγορίας που ανήκει ένα αντικείμενο
- Αναγωγή (Regression)
  - Εκτίμηση σχέσεων μεταξύ των μεταβλητών
- Ομαδοποίηση (Clustering)
  - Αυτόματη ομαδοποίηση όμοιων αντικειμένων σε ξεχωριστά σύνολα



# Tensorflow / Keras



## TensorFlow

- Πλατφόρμα *ανοιχτού κώδικα* δημιουργίας μοντέλων *μηχανικής μάθησης*
- Υποστηρίζει *ανάπτυξη και εκπαίδευση μοντέλων* σε Python, JavaScript ή Swift<sup>(\*)</sup>
- Μπορεί να τρέχει εύκολα στον *υπολογιστή*, σε *browser*, ή ακόμα και στο *cloud*



## Keras

- Διεπαφή *υψηλού επιπέδου* (*high level interface*) για *μηχανική μάθηση* γραμμένη σε Python
- Έμφαση στη δυνατότητα εύκολης και άμεσης δημιουργίας πειραματικών μοντέλων
- Ιδιαίτερη βαρύτητα στη λεγόμενη *βαθιά μάθηση* (*deep learning*)
- Τρέχει πάνω από την πλατφόρμα μηχανικής μάθησης TensorFlow
- Παλαιότερα υποστήριζε και άλλες πλατφόρμες (Microsoft Cognitive Toolkit, R, Theano) όμως μετά την έκδοση 2.4 (Ιούνιος 2020) υποστηρίζει αποκλειστικά την πλατφόρμα TensorFlow

<sup>(\*)</sup> σχετικά καινούρια (2014) γλώσσα προγραμματισμού που ξεκίνησε να αναπτύσσεται από την Apple το 2014

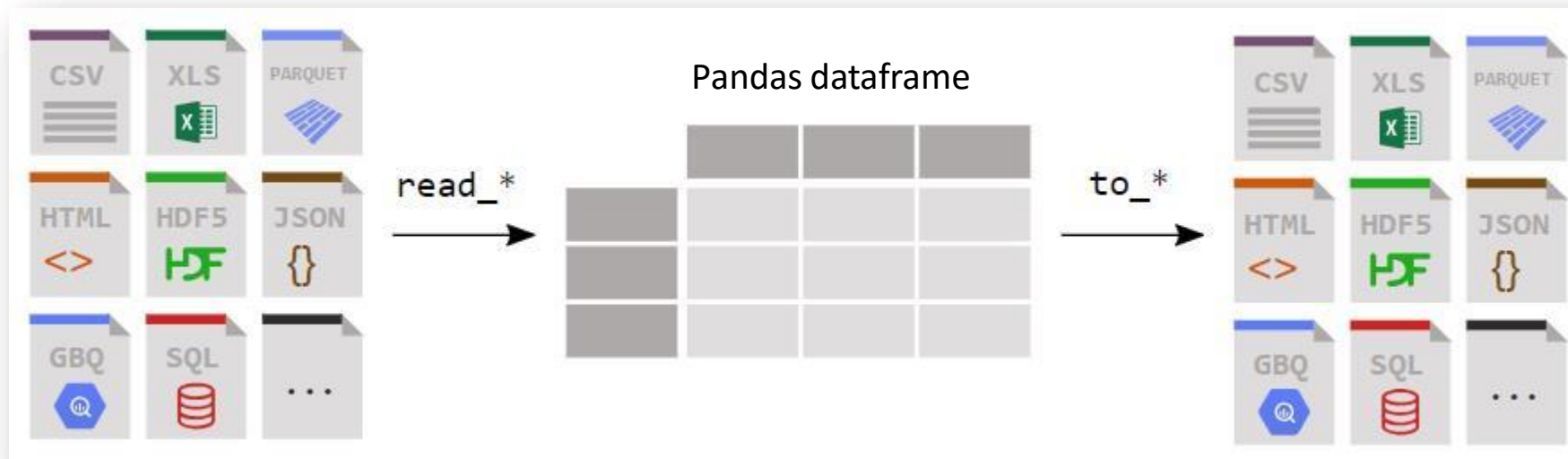
# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

|   |   |
|---|---|
| Επιστημονικός προγραμματισμός<br>/ Αριθμητική ανάλυση<br><i>Scientific /Numeric Computing</i> | <i>NumPy, SciPy, Matplotlib</i>         |
| Ανάπτυξη διεπαφών χρήστη-υπολογιστή<br><i>GUI Programming</i>                                 | <i>Tkinter, wxPython</i>                |
| Συγκομιδή περιεχομένου ιστοσελίδων<br><i>Web Scraping</i>                                     | <i>BeautifulSoup, Requests, Urllib3</i> |
| Ανάπτυξη εφαρμογών Web<br><i>Web Application Development</i>                                  | <i>Flask, Django</i>                    |
| Επιστήμη Δεδομένων & Μηχανική Μάθηση<br><i>Data Science &amp; Machine Learning</i>            | <i>scikit-learn, Tensorflow , Keras</i> |
| Επεξεργασία και ανάλυση δεδομένων<br><i>Data manipulation and analysis</i>                    | <i>pandas</i>                           |
| Ανάπτυξη παιχνιδιών<br><i>Game Programming</i>  | <i>PyGame, Pyglet , Panda3D</i>         |

# Pandas

Βιβλιοθήκη της Python για ανάλυση και χειρισμό δεδομένων (βασίζεται πάνω στην NumPy)

- Περιέχει/ορίζει *ειδικές δισδιάστατες δομές δεδομένων* που ονομάζονται *dataframes*
- Τα dataframes μπορούν να αποθηκεύσουν δεδομένα διαφορετικών τύπων (χαρακτήρες, ακέραιους, δεκαδικούς...) σε στήλες, *παρόμοια με τα υπολογιστικά φύλλα Excel*
- Υποστηρίζει την αυτόματη *εισαγωγή* και μετατροπή δεδομένων από *διαφορετικά μοντέλα μορφοποίησης* (xls, html, SQL, json) σε dataframes για περαιτέρω επεξεργασία, καθώς και την *επανεξαγωγή* τους μετά το πέρας της επεξεργασίας.



# Παράδειγμα χρήσης dataframe

```
import pandas as pd
df = pd.DataFrame({
    "Στήλη_1": [1, 4, 8, 7, 9],
    "Στήλη_2": ['μια', 'στήλη', 'με', 'λίγο', 'κείμενο'],
    "Στήλη_3": [1.23, 23.5, 45.6, 32.1234, 89.453],
    "Στήλη_4": [True, False, True, False, True]
})
print(df)
```

|   | Στήλη_1 | Στήλη_2 | Στήλη_3 | Στήλη_4 |
|---|---------|---------|---------|---------|
| 0 | 1       | μια     | 1.2300  | True    |
| 1 | 4       | στήλη   | 23.5000 | False   |
| 2 | 8       | με      | 45.6000 | True    |
| 3 | 7       | λίγο    | 32.1234 | False   |
| 4 | 9       | κείμενο | 89.4530 | True    |

```
import pandas as pd
s = pd.Series({
    'a':1,
    'b':2,
    'c':3})
print(s)
```

|   |   |
|---|---|
| a | 1 |
| b | 2 |
| c | 3 |

```
print(df.loc[2])
```

|         |      |
|---------|------|
| Στήλη_1 | 8    |
| Στήλη_2 | με   |
| Στήλη_3 | 45.6 |
| Στήλη_4 | True |

# Βασικές βιβλιοθήκες ανά περιοχή εφαρμογών

|   |   |
|---|---|
| Επιστημονικός προγραμματισμός<br>/ Αριθμητική ανάλυση<br><i>Scientific /Numeric Computing</i> | <i>NumPy, SciPy, Matplotlib</i>         |
| Ανάπτυξη διεπαφών χρήστη-υπολογιστή<br><i>GUI Programming</i>                                 | <i>Tkinter, wxPython</i>                |
| Συγκομιδή περιεχομένου ιστοσελίδων<br><i>Web Scraping</i>                                     | <i>BeautifulSoup, Requests, Urllib3</i> |
| Ανάπτυξη εφαρμογών Web<br><i>Web Application Development</i>                                  | <i>Flask, Django</i>                    |
| Επιστήμη Δεδομένων & Μηχανική Μάθηση<br><i>Data Science &amp; Machine Learning</i>            | <i>scikit-learn, Tensorflow , Keras</i> |
| Επεξεργασία και ανάλυση δεδομένων<br><i>Data manipulation and analysis</i>                    | <i>pandas</i>                           |
| Ανάπτυξη παιχνιδιών<br><i>Game Programming</i>  | <i>PyGame, Pyglet , Panda3D</i>         |



- Βιβλιοθήκη της Python για την ανάπτυξη παιχνιδιών, χτισμένη πάνω στο πρότυπο SDL (Simple DirectMedia Layer)
- Απλή και εύκολη στην εκμάθηση και την χρήση
- Υποστηρίζει όλες τις βασικές πλατφόρμες λογισμικού (Windows, Mac OS X, Linux, iOS, Android)
- Για αμεσότητα απόκρισης, κομμάτια του πυρήνα της είναι γραμμένα σε C και Assembly

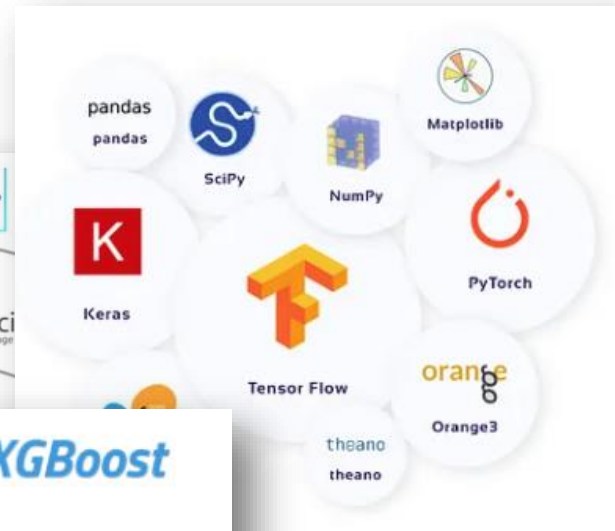
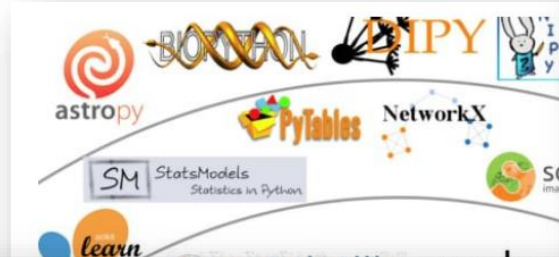


- Βιβλιοθήκη της Python για την ανάπτυξη παιχνιδιών και πολυμεσικών εφαρμογών
- Γραμμένη εξ' ολοκλήρου σε Python, δεν χρειάζεται εγκατάσταση άλλων πακέτων
- Υποστηρίζει τις πλατφόρμες λογισμικού Windows, Mac OS X και Linux
- Υποστηρίζει εκτέλεση σε παράθυρο, πλήρη οθόνη και πολλαπλές οθόνες (μόνιτορς)



- *Μηχανή ανάπτυξης παιχνιδιών (game engine)* που περιλαμβάνει γραφικά, ήχο, χειρισμό I/O , *ανίχνευση συγκρούσεων αντικειμένων (collision detection)* και άλλες δυνατότητες για τη δημιουργία *τριδιάστατων παιχνιδιών*
- *Σοβαρή εφαρμογή*, δεν απευθύνεται σε αρχάριους χρήστες που θέλουν να "παίξουν"
- *Πυρήνας σε C++* για ταχύτητα, *διεπαφή σε Python* για όλα τα υπόλοιπα πλεονεκτήματα  
*Ένας προγραμματιστής που χρησιμοποιεί το Panda3D συνήθως γράφει κώδικα σε Python, αλλά είναι επίσης δυνατό να έχει άμεση πρόσβαση στον πυρήνα του χρησιμοποιώντας κώδικα C ++*
- Χρησιμοποιείται κυρίως για *ανάπτυξη παιχνιδιών για εμπορική χρήση* εξ' αιτίας κυρίως τεσσάρων βασικών πλεονεκτημάτων της
  - ✓ Πληρότητα εργαλείων ανάπτυξης
  - ✓ Ταχύτητα ανταπόκρισης (πολύ σημαντική σε διαδραστικά παιχνίδια)
  - ✓ Παροχή εργαλείων υποστήριξης και βελτιστοποίησης
  - ✓ Ανοχή στα λάθη (δεν "χτυπάει" σε ακραίες ή απρόβλεπτες συνθήκες)

# Φυσικά, υπάρχουν και πάρα πολλές άλλες βιβλιοθήκες ....



οι περισσότερες από αυτές είναι *ανοικτού-κώδικα* και *υποστηρίζονται ενεργά* από την κοινότητα της Python

# Τέλος Διάλεξης

## Ερωτήσεις;

*Τμήματα αυτής της διάλεξης περιέχουν στοιχεία από πηγές που είναι ελεύθερες στο διαδίκτυο όπως η Βικιπαιδεια καθώς και ανοιχτές σημειώσεις παρεμφερών διαλέξεων*