Introduction - Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Applications for Python

Web and Internet Development:

Python offers many choices for web development:

- *Frameworks such as **Django** and **Pyramid.***
- *Micro-frameworks such as **Flask** and **Bottle.***
- *Advanced content management systems such as **Plone** and **django CMS.***
- *Python's standard library supports many Internet protocols:*
  HTML and XML
  JSON
  E-mail processing.
  Support for FTP, IMAP, and other Internet protocols.
  Twisted Python, a framework for asynchronous network programming.

Applications for Python

Scientific and Numeric

Python is widely used in scientific and numeric computing:

- **SciPy** and **NumPy** are a collection of packages for mathematics, science, and engineering.
- **Pandas** is a data analysis and modeling library.
- **IPython** is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.

Education

Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.

Desktop GUIs

- The **Tk GUI library** is included with most binary distributions of Python.
- Some toolkits that are usable on several platforms are available separately:

  wxWidgets

  Kivy, for writing multitouch applications.

  Qt via pyqt or pyside
- Platform-specific toolkits are also available:

  GTK+

  Microsoft Foundation Classes through the win32 extensions

Software Development

Python is often used as a support language for software developers, for build control and management, testing, and in many other ways.

- **SCons** for build control.
- **Buildbot** and **Apache Gump** for automated continuous compilation and testing.
- **Roundup** or **Trac** for bug tracking and project management.

Business Applications

Python is also used to build ERP and e-commerce systems:

- **Odoo** is an all-in-one management software that offers a range of business applications that form a complete suite of enterprise management applications.
- **Tryton** is a three-tier high-level general purpose application platform.

Knowledge Sources

https://docs.python.org/3.8/tutorial/

https://www.w3schools.com/python/default.asp

https://www.learnpython.org/

https://realpython.com/numpy-scipy-pandas-correlation-python/

https://benalexkeen.com/correlation-in-python/

Basic concepts of Python - the What's and the Why's

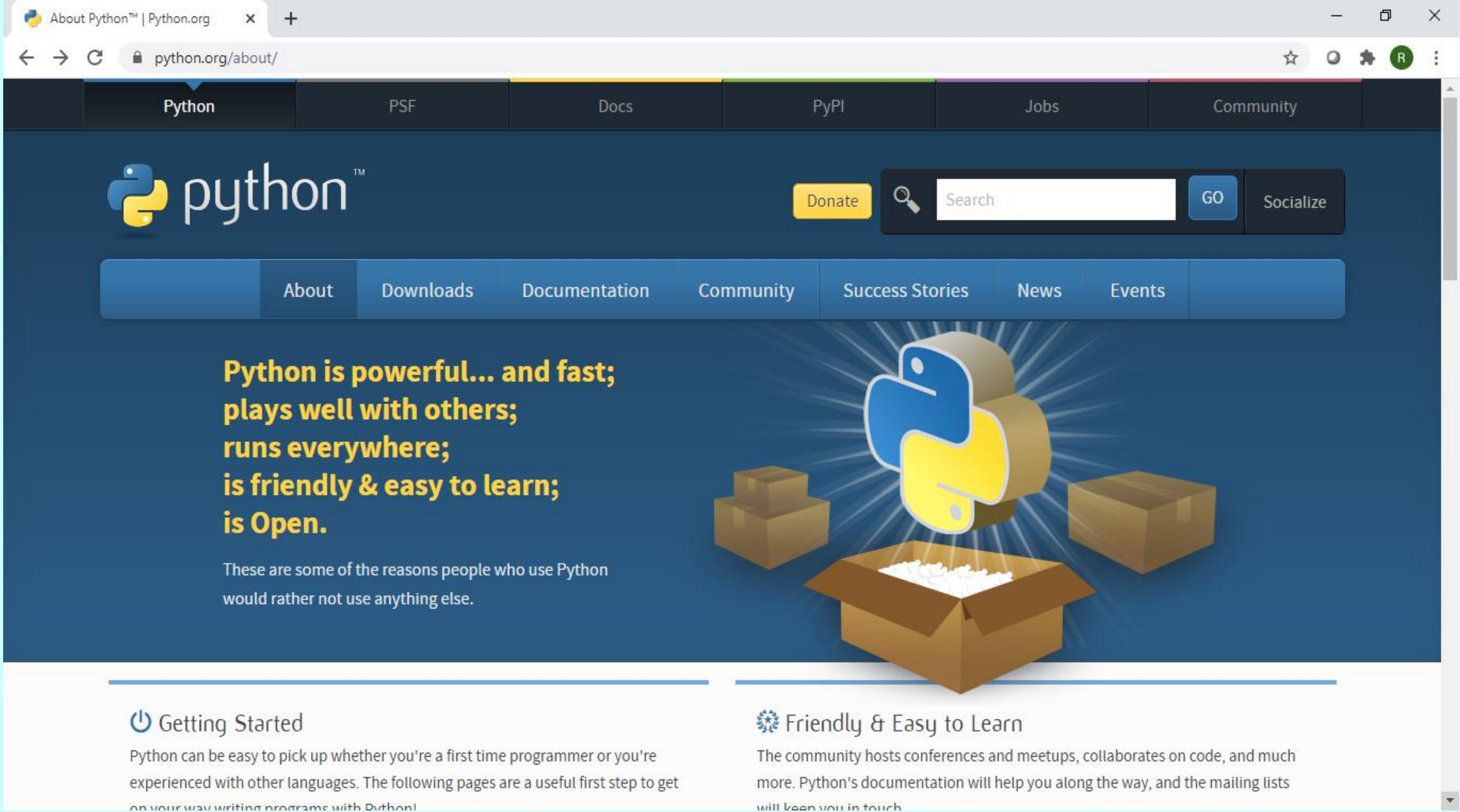
What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Basic concepts of Python - the What's and the Why's

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.



Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.

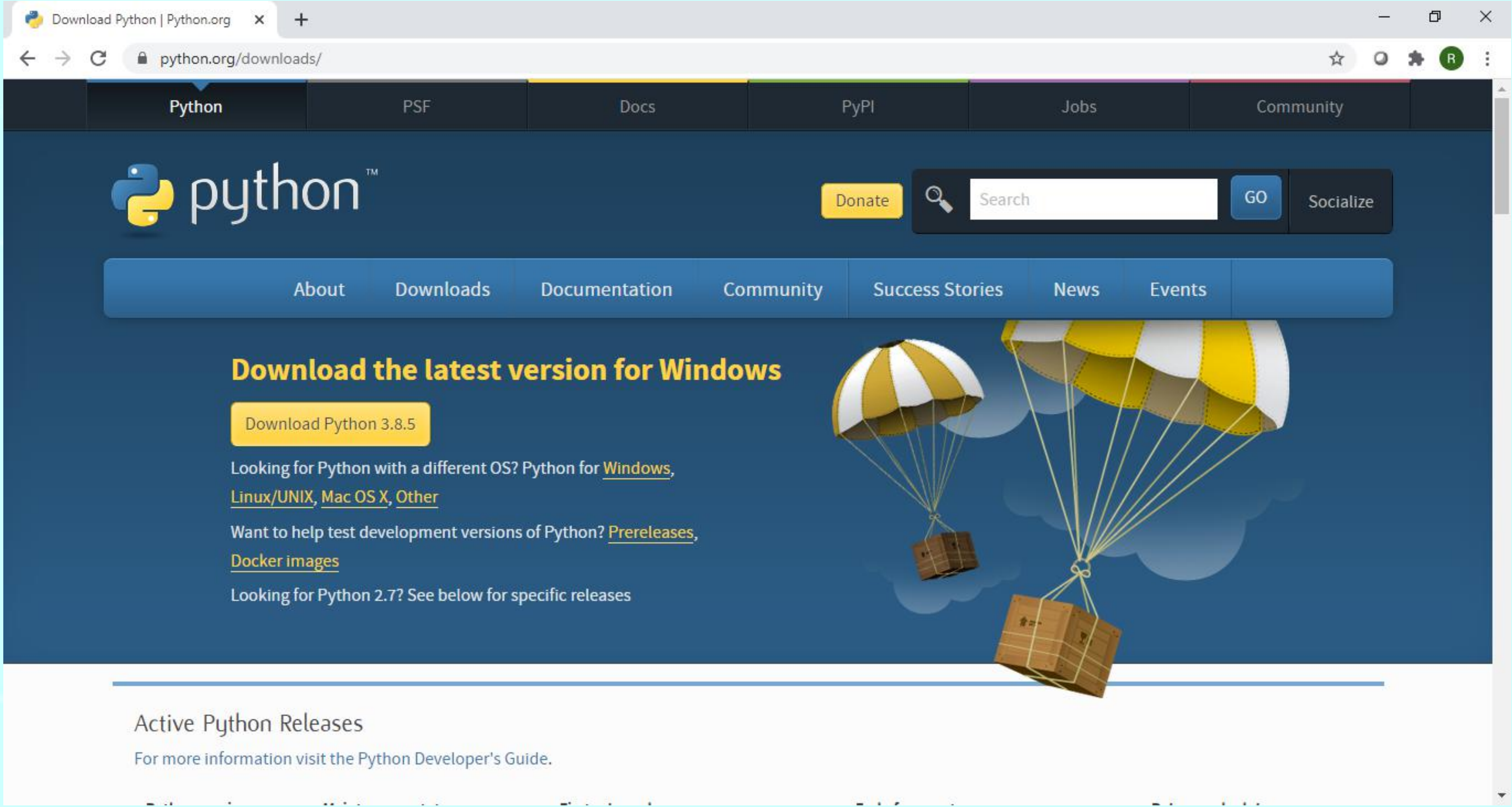
These are some of the reasons people who use Python would rather not use anything else.

Getting Started

Python can be easy to pick up whether you're a first time programmer or you're experienced with other languages. The following pages are a useful first step to get on your way writing programs with Python!

Friendly & Easy to Learn

The community hosts conferences and meetups, collaborates on code, and much more. Python's documentation will help you along the way, and the mailing lists will keep you in touch.



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Active Python Releases

For more information visit the Python Developer's Guide.

Python and Statistics/Scientific & Numeric Computing

Numerical Computing defines an area of computer science and mathematics dealing with algorithms for numerical approximations of problems from mathematical or numerical analysis,

In other words: Algorithms solving problems involving continuous variables.

Numerical analysis is used to solve science and engineering problems.

Tools for Scientific & Numeric Computing

Tools: Libraries and Software

In addition to standart tools like editors, compilers, or debuggers, there is a lot of (commercial or public domain) support available:

Modelling:

Computer algebra programs like Mathematica, Maple,Axiom, or Reduce support derivations and proofs of theorems via symbolic means.

Numerics:

Mathematica, Maple, or MATLAB support the development, testing, and analysis of (numerical) algorithms and allow an efficient prototyping.

Tools for Scientific & Numeric Computing

Implementation:
A zoo of (numerical) libraries provide up-to-date modules for standard tasks (numerical linear algebra etc.), tailored to specific target architectures.

Visualization:
Packages like IDL, IRIS Explorer, or AVS/Express offer (nearly) all you want.

Tools for Scientific & Numeric Computing

Libraries and Collections

GAMS - Guide to Available Mathematical Software
http://gams.nist.gov/

Matrix Market - http://math.nist.gov/MatrixMarket/

NETLIB - http://www.netlib.org/

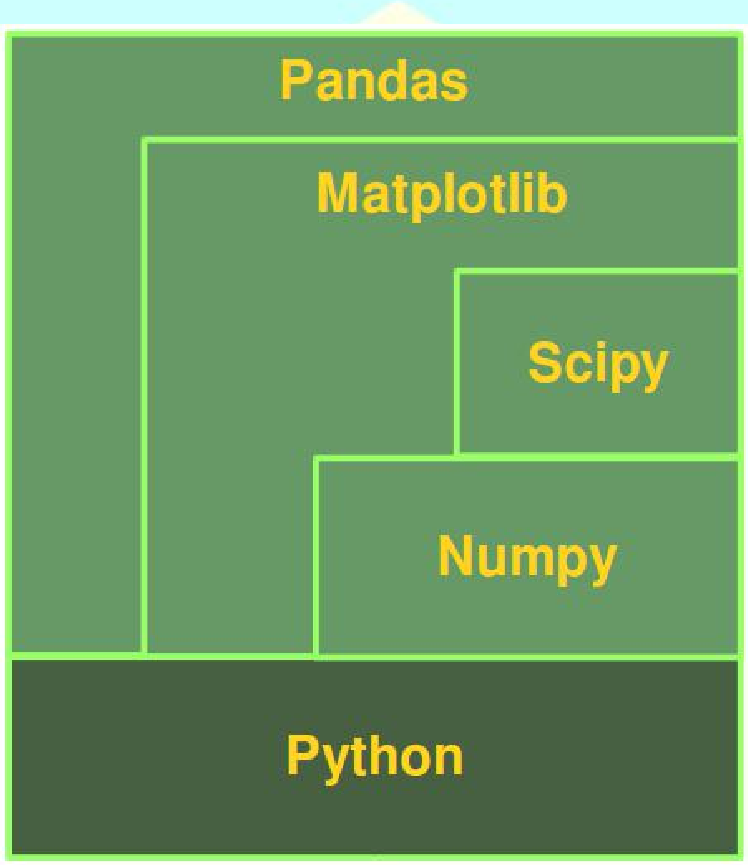
Diffpack - http://www.nobjects.com

and many more

Tools for Scientific & Numeric Computing

Numpy is a module which provides the basic data structures, implementing **Multi-dimensional arrays and matrices**.

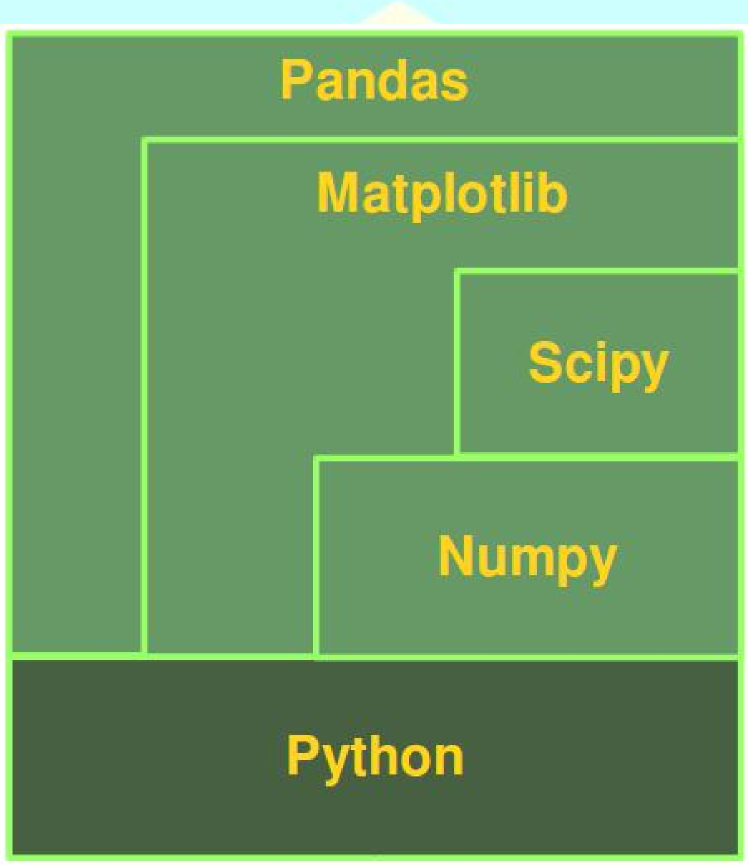
SciPy is based on top of Numpy, i.e. it uses the data structures provided by NumPy. It extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others.



Tools for Scientific & Numeric Computing

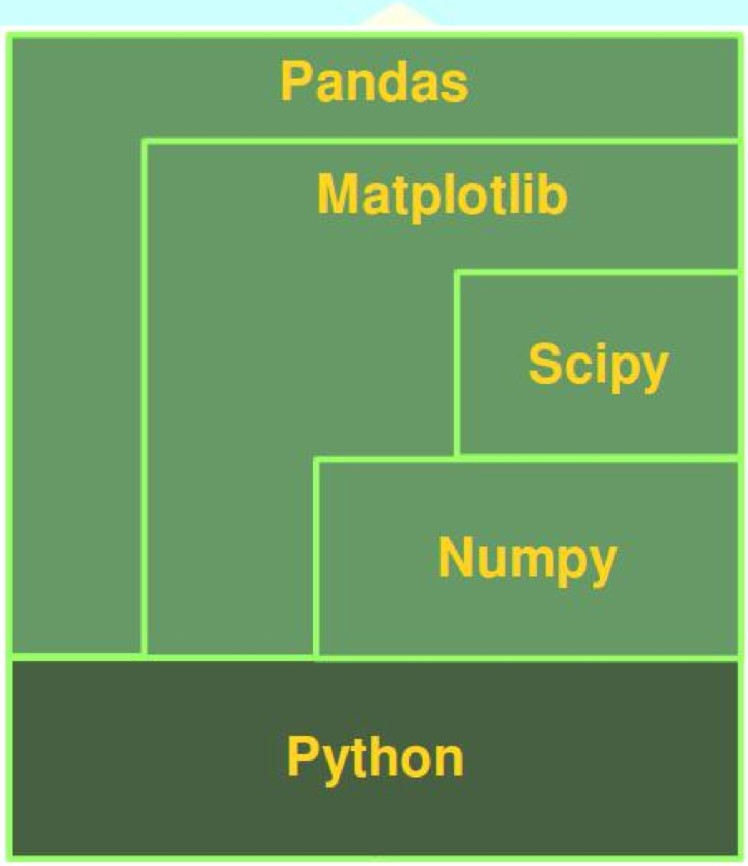
Matplotlib is a plotting library for the Python programming language and the numerically oriented modules like *NumPy* and *SciPy*.

Pandas is using all of the previously mentioned modules. It's build on top of them to provide a module for the Python language, which is also capable of data manipulation and analysis.

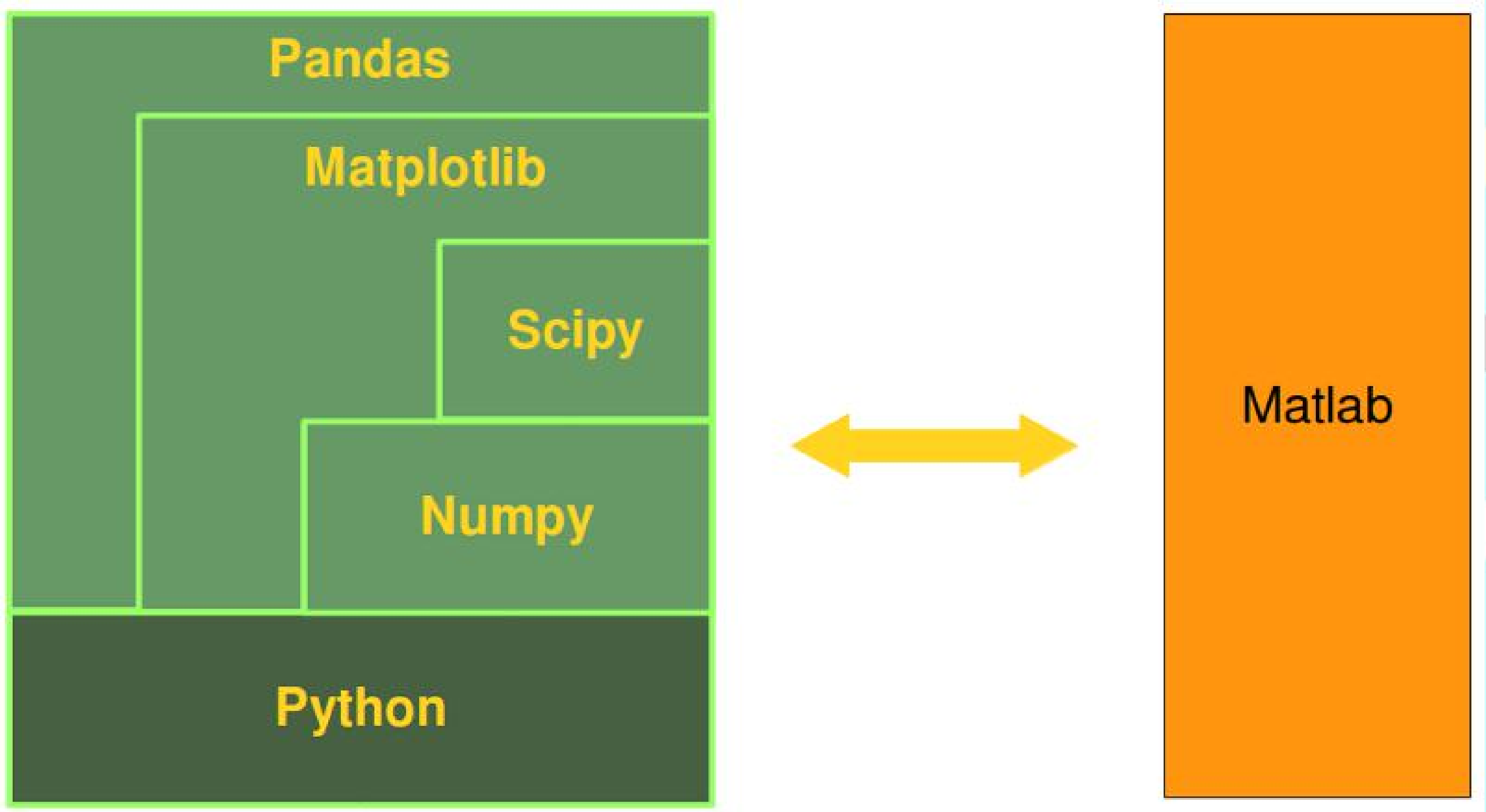


Tools for Scientific & Numeric Computing

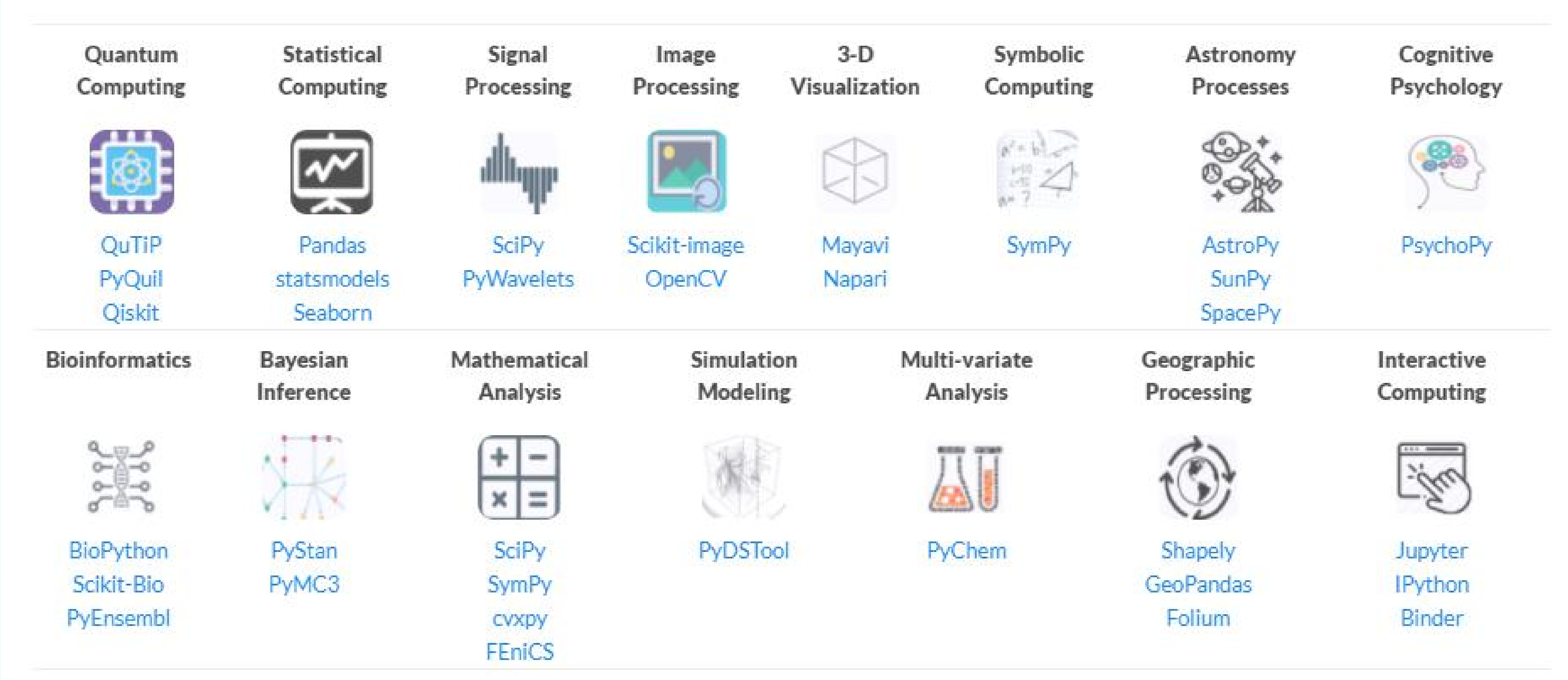
The special focus of Pandas consists in offering data structures and operations for manipulating numerical tables and time series. The name is derived from the term "*panel data*". Pandas is well suited for working with tabular data as it is known from spread sheet programming like Excel.



Python - An Alternative of Matlab



NumPy brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use. With this power comes simplicity: a solution in NumPy is often clear and elegant.



Quantum Computing	Statistical Computing	Signal Processing	Image Processing	3-D Visualization	Symbolic Computing	Astronomy Processes	Cognitive Psychology
QuTiP PyQuil Qiskit	Pandas statsmodels Seaborn	SciPy PyWavelets	Scikit-image OpenCV	Mayavi Napari	SymPy	AstroPy SunPy SpacePy	PsychoPy

Bioinformatics	Bayesian Inference	Mathematical Analysis	Simulation Modeling	Multi-variate Analysis	Geographic Processing	Interactive Computing
BioPython Scikit-Bio PyEnsembl	PyStan PyMC3	SciPy SymPy cvxpy FEniCS	PyDSTool	PyChem	Shapely GeoPandas Folium	Jupyter IPython Binder