Introduction to Robotics

Manipulation and Programming

Unit 1: Introduction

INSTALL PYTHON AND PYTHON ROBOTICS

DR. ERIC CHOU

IEEE SENIOR MEMBER

Syllabus

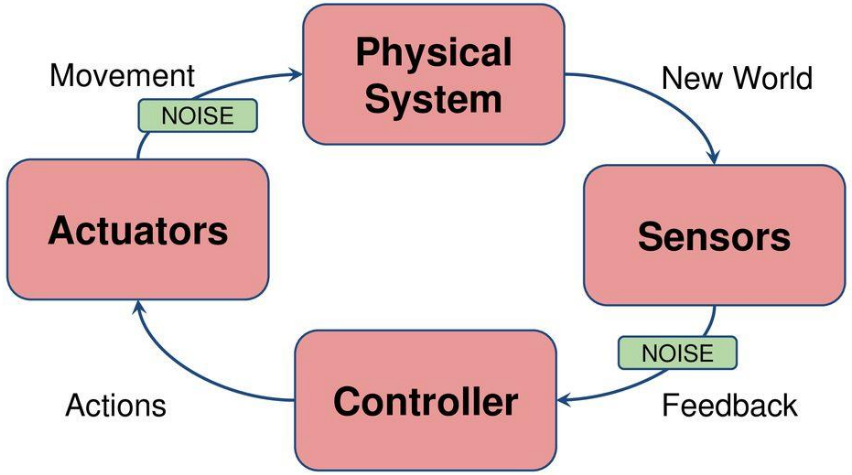
Unit 1: Introduction

Unit 2: Kinematics

Unit 3: Sensors

Unit 4: Motion Control

Control System



Course Objectives

- This course cover a wide range of robotics topics common to most robotics classes, including forward and inverse kinematics, sensors and computer vision (machine vision), and feedback motion control. Theory is paired with hands-on building and programming tasks using a kit of parts that allows you to practice as you learn, building and programming complete robots.
- The course involves programming in C and Python. It is helpful if you have some experience already with matrix math (adding/subtracting and multiplying matrices and vectors) and trigonometry.

Course Objectives

- All of the topics in the course lead to building and programming a complete, working, 3-degree-of-freedom SCARA pick-and-place manipulator by understanding and practicing all of the theory underneath. This course use XARM as an example robot model.

Unit 1

- Install Python and Python Robotics
- Assemble and Manipulate XARM
- Basic Python Mathematics
- Introduction to Arduino Sensor Kit

Objectives

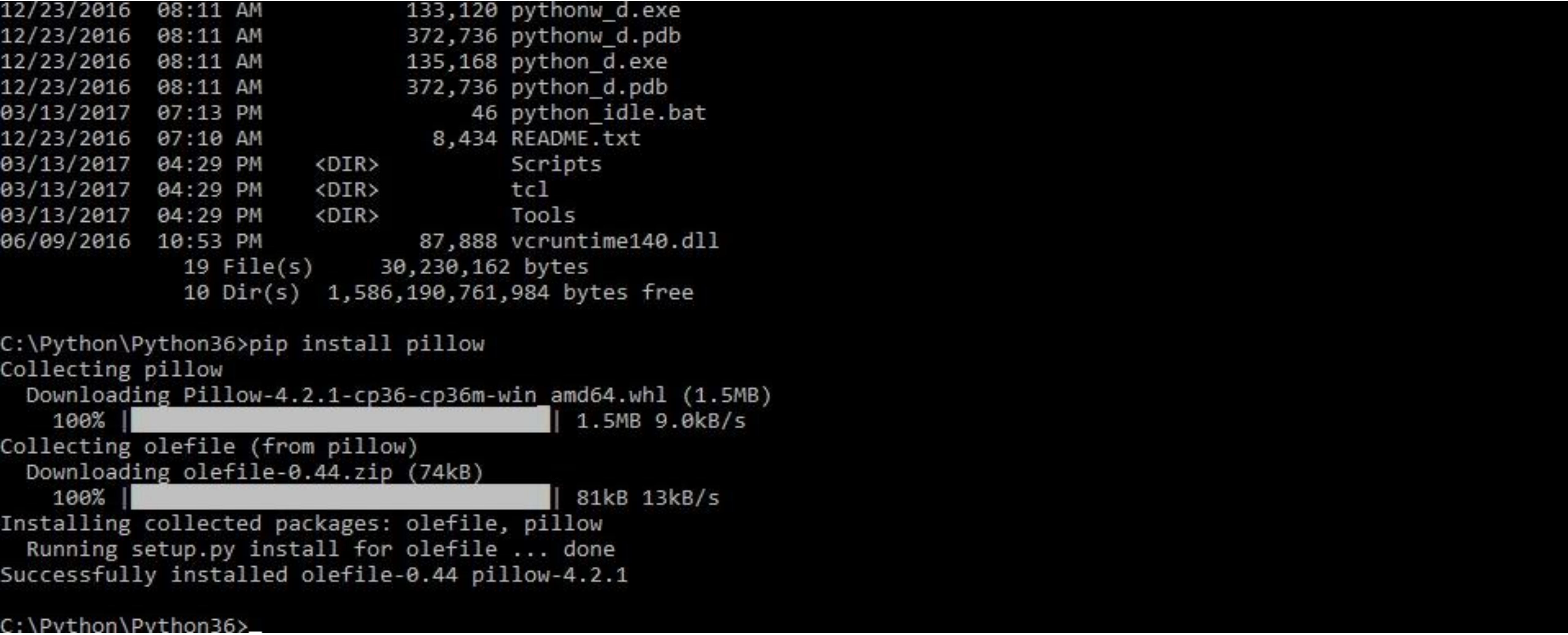
- Install Python Interpreter and IDLE
- Python Graphics Frameworks (Pillow, piglet, PyGame)
- Python Scientific Packages
- PythonRobotics

Installation

SECTION 1

Basic GUI package

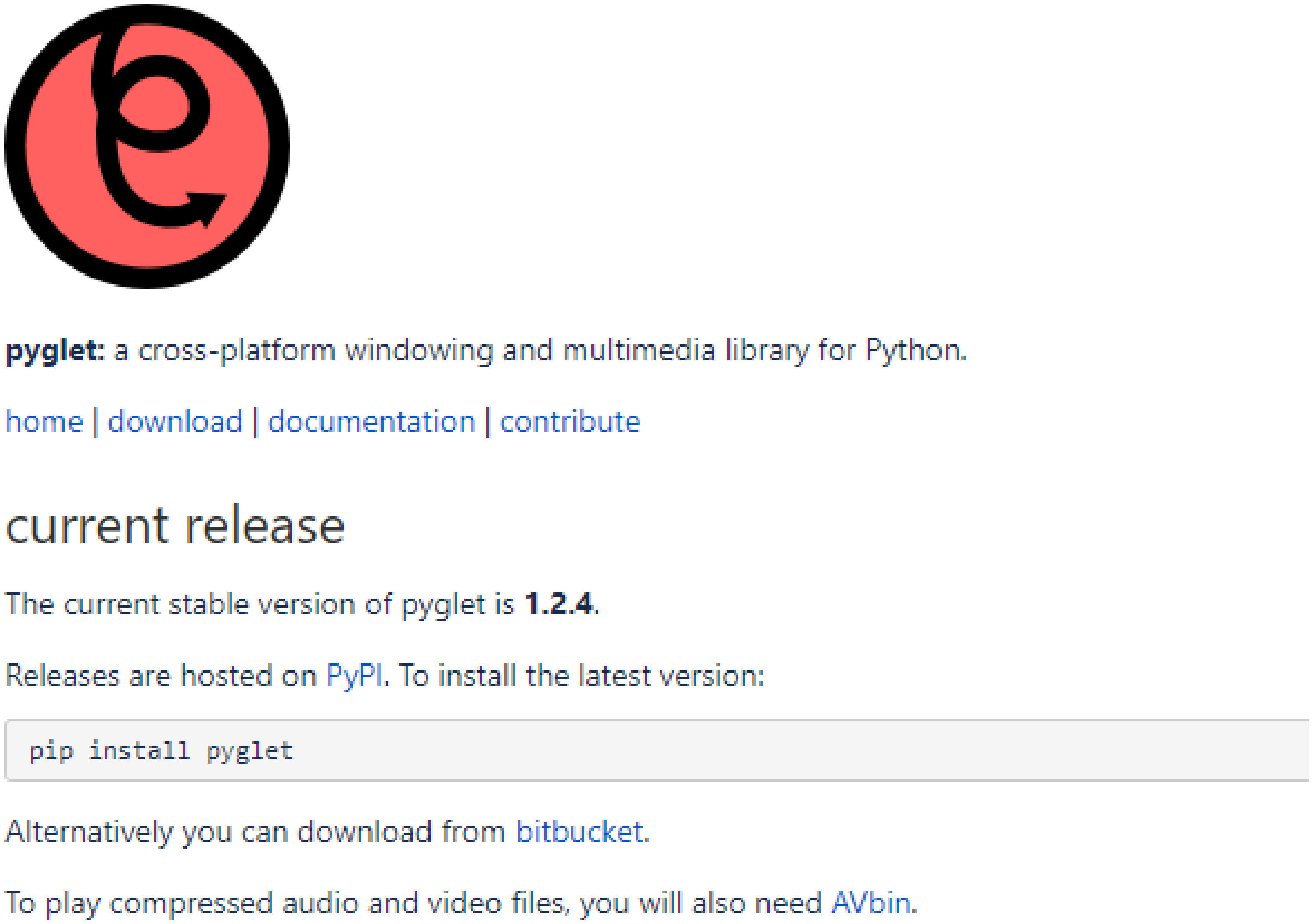
- Tkinter and (TTK) included in Python 3.x
- Pillow (Advanced PIL)
- PyGame
- Pyglet



```
12/23/2016  08:11 AM           133,120 pythonw_d.exe
12/23/2016  08:11 AM           372,736 pythonw_d.pdb
12/23/2016  08:11 AM           135,168 python_d.exe
12/23/2016  08:11 AM           372,736 python_d.pdb
03/13/2017  07:13 PM                46 python_idle.bat
12/23/2016  07:10 AM             8,434 README.txt
03/13/2017  04:29 PM    <DIR>          Scripts
03/13/2017  04:29 PM    <DIR>          tcl
03/13/2017  04:29 PM    <DIR>          Tools
06/09/2016  10:53 PM            87,888 vcruntime140.dll
              19 File(s)     30,230,162 bytes
              10 Dir(s)  1,586,190,761,984 bytes free

C:\Python\Python36>pip install pillow
Collecting pillow
  Downloading Pillow-4.2.1-cp36-cp36m-win_amd64.whl (1.5MB)
    100% |████████████████████████████████| 1.5MB 9.0kB/s
Collecting olefile (from pillow)
  Downloading olefile-0.44.zip (74kB)
    100% |████████████████████████████████| 81kB 13kB/s
Installing collected packages: olefile, pillow
  Running setup.py install for olefile ... done
Successfully installed olefile-0.44 pillow-4.2.1

C:\Python\Python36>_
```



pyglet: a cross-platform windowing and multimedia library for Python.

home | download | documentation | contribute

current release

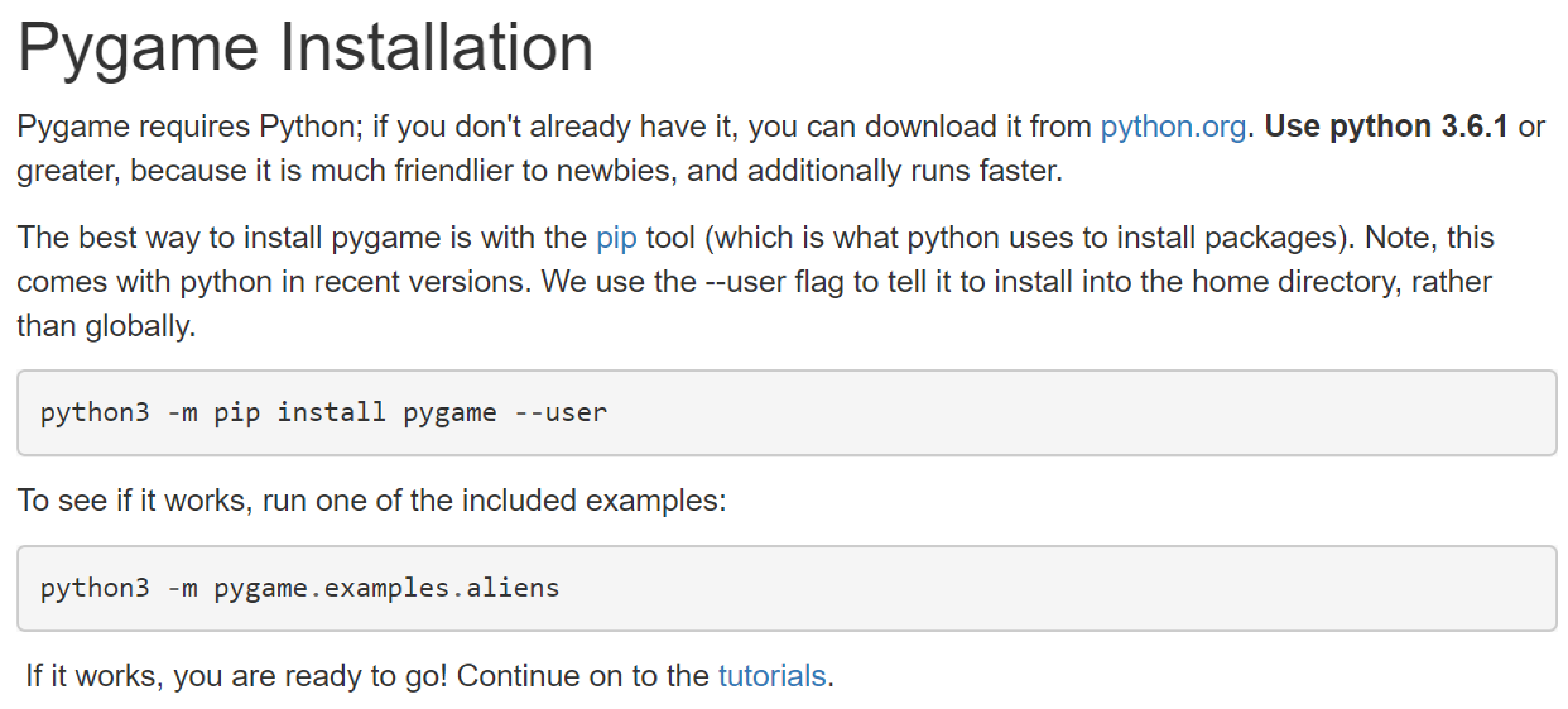
The current stable version of pyglet is **1.2.4**.

Releases are hosted on PyPI. To install the latest version:

```
pip install pyglet
```

Alternatively you can download from bitbucket.

To play compressed audio and video files, you will also need AVbin.



Pygame Installation

Pygame requires Python; if you don't already have it, you can download it from python.org. **Use python 3.6.1** or greater, because it is much friendlier to newbies, and additionally runs faster.

The best way to install pygame is with the pip tool (which is what python uses to install packages). Note, this comes with python in recent versions. We use the --user flag to tell it to install into the home directory, rather than globally.

```
python3 -m pip install pygame --user
```

To see if it works, run one of the included examples:

```
python3 -m pygame.examples.aliens
```

If it works, you are ready to go! Continue on to the tutorials.

Numerical Packages
NumPy/SciPy/Matplotlib

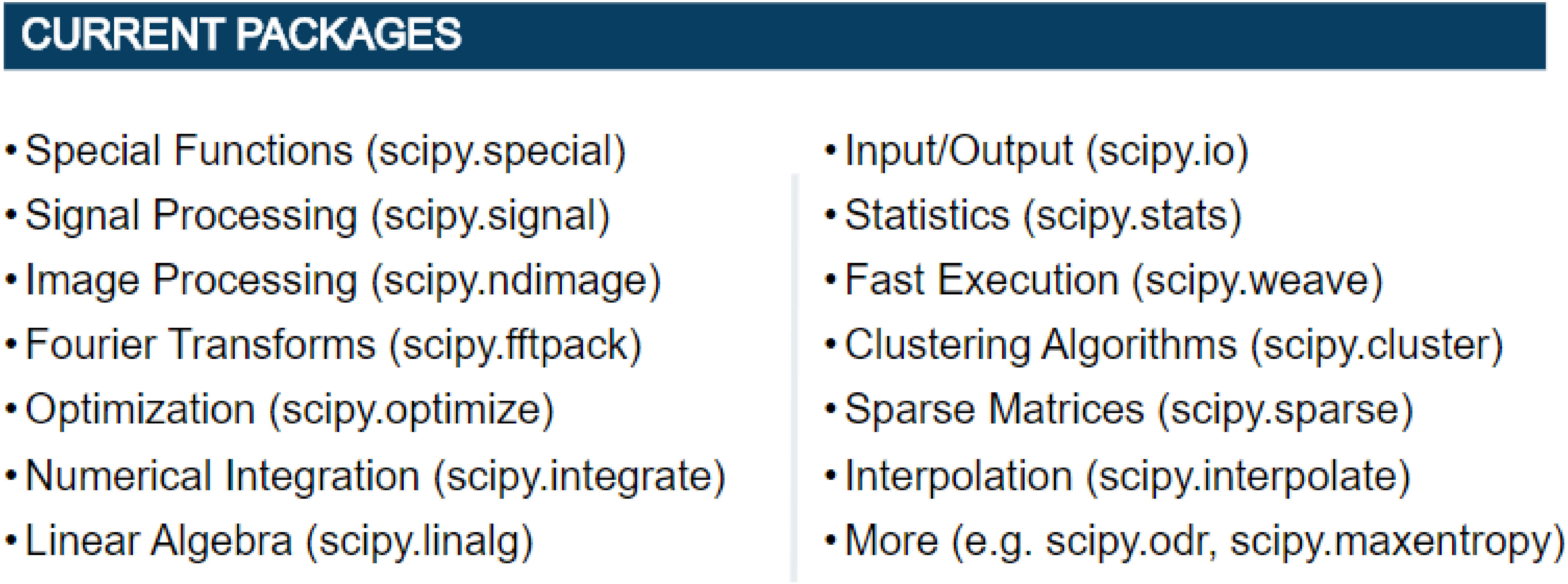
HTTP://WWW.NUMPY.ORG/

NumPy

- **NumPy** is the fundamental package for scientific computing with **Python**. It contains among other things:
  - a powerful N-dimensional array object
  - sophisticated (broadcasting) functions
  - tools for integrating C/C++ and Fortran code
  - useful linear algebra, Fourier transform, and random number capabilities

- Besides its obvious scientific uses, **NumPy** can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows **NumPy** to seamlessly and speedily integrate with a wide variety of databases.

SciPy

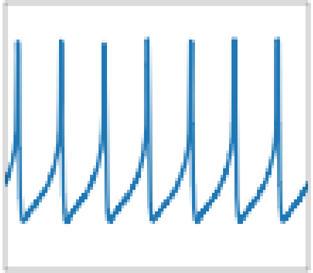


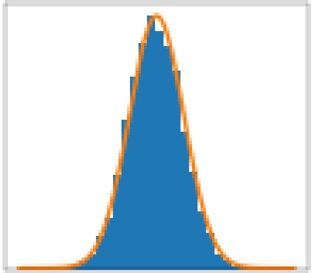
CURRENT PACKAGES

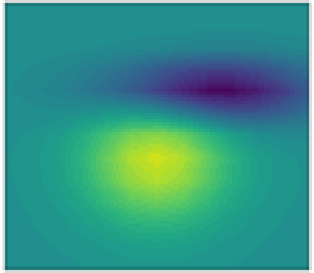
- Special Functions (scipy.special)
- Signal Processing (scipy.signal)
- Image Processing (scipy.ndimage)
- Fourier Transforms (scipy.fftpack)
- Optimization (scipy.optimize)
- Numerical Integration (scipy.integrate)
- Linear Algebra (scipy.linalg)
- Input/Output (scipy.io)
- Statistics (scipy.stats)
- Fast Execution (scipy.weave)
- Clustering Algorithms (scipy.cluster)
- Sparse Matrices (scipy.sparse)
- Interpolation (scipy.interpolate)
- More (e.g. scipy.odr, scipy.maxentropy)

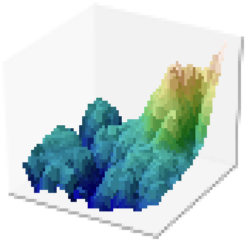
Matplotlib

- Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
- Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.



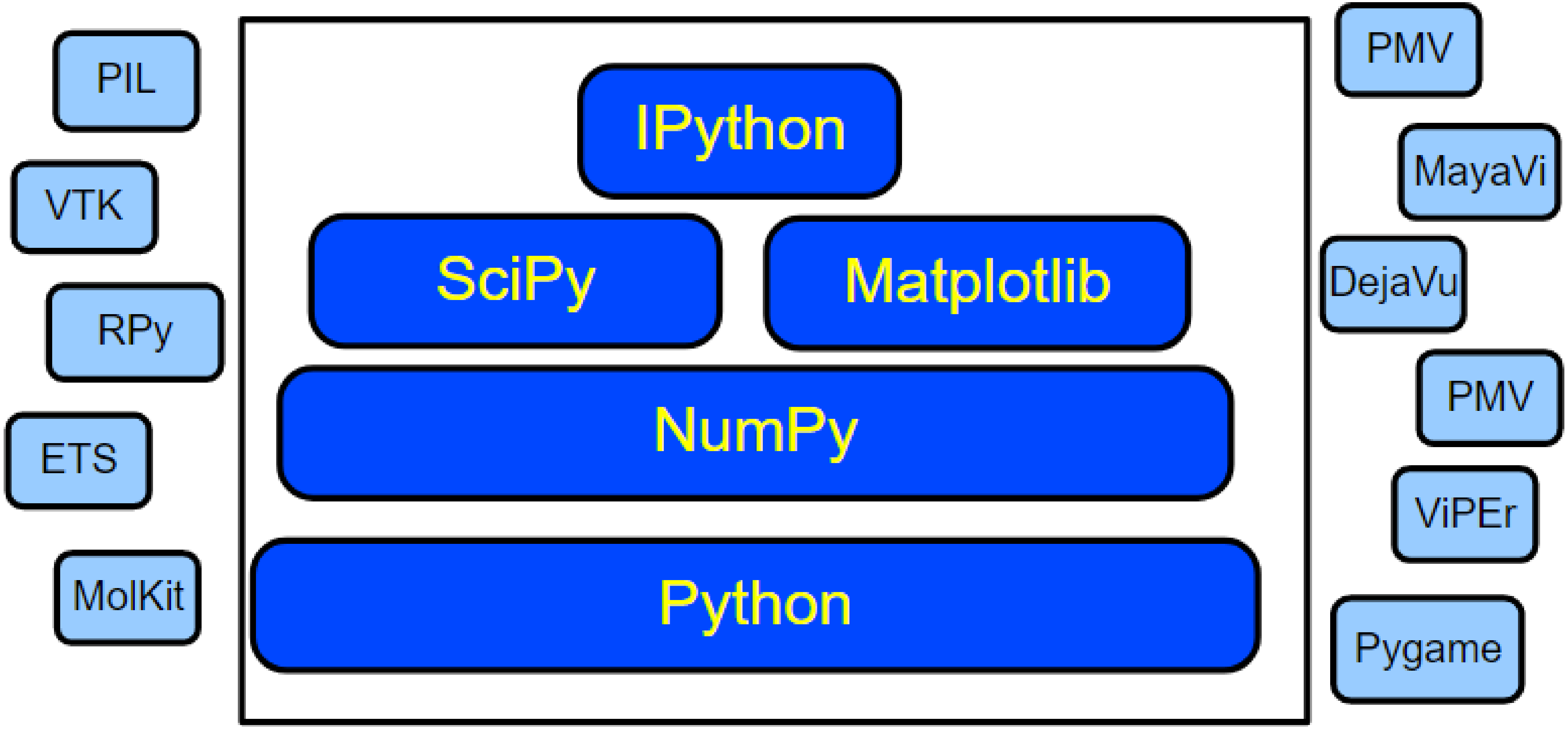






Matplotlib

- Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For a sampling, see the screenshots, thumbnail gallery, and examples directory
- For simple plotting the **pyplot** module provides a MATLAB-like interface, particularly when combined with **IPython**.
- For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.


PIL
VTK
RPy
ETS
MolKit
IPython
SciPy
Matplotlib
NumPy
Python
PMV
MayaVi
DejaVu
PMV
ViPEr
Pygame

Installing via pip

- Python comes with an inbuilt package management system, pip. Pip can install, update, or delete any official package.
- You can install packages via the command line by entering:

```
pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose
```

Installing via pip

- Python comes with an inbuilt package management system, pip. Pip can install, update, or delete any official package.
- You can install packages via the command line by entering:

```
pip install cvxpy
```