**Graphical Interface**

The Python Standard Library comes with TkInter, but you can take your GUIs to the next level using an external Python module.

[**wxPython**](https://wxpython.org/): Create truly native user interfaces for their Python applications that run with little or no modifications on Windows, Mac and Linux or other Unix-like systems.

[**PyGObject**](https://pygobject.readthedocs.io/en/latest/): Python package which provides bindings for GObject based libraries such as GTK, GStreamer, WebKitGTK, GLib, GIO, and many more.

[**Pmw**](http://pmw.sourceforge.net/): Toolkit for building high-level compound widgets in Python using the Tkinter module.

[**WCK**](http://effbot.org/zone/wck.htm) : Extension API that allows you to implement all sorts of custom widgets, in pure Python.

[**Tix**](http://tix.sourceforge.net/): A powerful set of user interface components that expands the capabilities of your Tcl/Tk and Python applications. Using Tix together with Tk will greatly enhance the appearance and functionality of your application.

**Databases**

The following open-source modules will allow you to easily access data stored in databases.

[**MySQLdb**](https://sourceforge.net/projects/mysql-python/): Python DB API-2.0-compliant interface for accessing MySQL databases.

[**PyGreSQL**](https://www.pygresql.org/): An open-source module that interfaces to a PostgreSQL database. It embeds the PostgreSQL query library to allow easy use of the powerful PostgreSQL features from a Python script.

[**Gadfly**](http://gadfly.sourceforge.net/): A simple relational database system implemented in Python based on the SQL Structured Query Language.

[**SQLAlchemy**](https://www.sqlalchemy.org/): A Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

[**KInterbasDB**](http://kinterbasdb.sourceforge.net/): A Python extension package that implements Python Database API 2.0-compliant support for the open source relational database Firebird and some versions of its proprietary cousin Borland Interbase.

**Web Development**

Python is a popular language for web development. Each Python package listed below will make your web developer life much easier.

[**Beautiful Soup**](https://www.crummy.com/software/BeautifulSoup/): A Python library designed for quick turnaround projects like screen-scraping.

[**scrape**](http://zesty.ca/python/scrape.html): A Python module for web browsing and scraping.

[**mechanize**](http://wwwsearch.sourceforge.net/mechanize/): Stateful programmatic web browsing in Python.

[**libgmail**](http://libgmail.sourceforge.net/): A pure Python binding to provide access to Google’s Gmail web-mail service.

[**Google Maps**](https://pypi.org/project/googlemaps/): This library brings the Google Maps Platform Web Services to your Python application.

[**Requests**](https://2.python-requests.org/en/master/) allows you to send HTTP/1.1 requests extremely easily.

[**Selenium**](https://www.selenium.dev/): With this module, developers can programmatically open webpages, enter fields, click buttons, and submit forms.

[**pyquery**](https://pythonhosted.org/pyquery/) allows you to make jQuery queries on XML documents. The API is as much as possible the similar to jQuery. Pyquery uses LXML for fast XML and HTML manipulation.

**Image and Video Manipulation**

Python is a very powerful language that can accomplish many tasks such as image manipulation. The Standard Library doesn’t provide any image manipulation built-in module, but the following will definitely help.

[**Python Imaging Library (PIL)**](http://www.pythonware.com/products/pil/): PIL adds image processing capabilities to your Python interpreter. This library supports many file formats, and provides powerful image processing and graphics capabilities.

[**GDmodule**](http://newcenturycomputers.net/projects/gdmodule.html): An interface to the GD library written by Thomas Boutell.

[**VideoCapture**](http://videocapture.sourceforge.net/): A Win32 Python extension for accessing video devices such as USB WebCams and TV cards.

[**MoviePy**](https://pypi.org/project/moviepy/): A Python library for video editing: cutting, concatenations, title insertions, video compositing (a.k.a. non-linear editing), video processing, and creation of custom effects.

[**pyscreenshot**](https://pypi.org/project/pyscreenshot/): A cross-platform module that allows to take screenshots without installing 3rd party libraries.

**Data Science and Maths**

Python features many extensions that can be used for scientific needs such as maths, data science, and engineering.

[**SciPy**](https://www.scipy.org/): SciPy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering.

[**Matplotlib**](https://matplotlib.org/): 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 interpreter, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

[**Pandas**](https://pandas.pydata.org/): A fast, powerful, flexible and easy-to-use open source data analysis and manipulation tool.

[**Numpy**](https://numpy.org/): A library adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

**Game Development**

Python is a versatile language that allows programmers to create many different types of apps, including video games.

[**Pygame**](https://www.pygame.org/news): A set of modules designed for writing video games. Pygame adds functionality on top of the excellent SDL library. This allows you to create fully featured games and multimedia programs in the Python language.

[**Pyglet**](http://pyglet.org/): A powerful, yet easy to use Python library for developing games and other visually-rich applications on Windows, Mac OS X, and Linux.

[**pyOpenGL**](http://pyopengl.sourceforge.net/): The most common cross platform Python binding to OpenGL and related APIs.

**Sound**

Sound manipulation is easily done in Python, thanks to a few very useful modules.

[**pySonic**](http://pysonic.sourceforge.net/): A Python wrapper around the high performance, cross platform FMOD sound library.

[**PyMedia**](http://pymedia.org/): A Python module for WAV, MP3, Ogg, AVI, DivX, DVD, CD-DA etc. file manipulations. It allows you to parse, demultiplex, multiplex, decode and encode all supported formats. It can be compiled for Windows, Linux, and Cygwin.

[**PMIDI**](http://www.cs.unc.edu/Research/assist/developer.shtml): The PMIDI library wraps the Windows MIDI Streams library for use in Python. Using PMIDI, developers can generate synthesized musical sequences on the fly in their code for playback to users.

[**Mutagen**](https://github.com/quodlibet/mutagen): A module to handle audio metadata. It supports FLAC, M4A, Musepack, MP3, Ogg FLAC, Ogg Speex, Ogg Theora, Ogg Vorbis, True Audio, and WavPack audio files.

## Microsoft Windows

If you are developing applications for Microsoft Windows, the following modules can help make your app better integrated with the OS.

[**pywin32**](https://github.com/mhammond/pywin32): A wrapper of Python that allows us to interact with COM objects and automate Windows applications with Python.

[**PyRTF**](http://pyrtf.sourceforge.net/): A set of Python classes that make it possible to produce RTF documents from Python programs.

[**WMI**](http://timgolden.me.uk/python/wmi/index.html): The Python WMI module is a lightweight wrapper on top of the pywin32 extensions, and hides some of the messy plumbing needed to get Python to talk to the WMI API.

[**Py2exe**](https://www.py2exe.org/) converts Python scripts into executable Windows programs, able to run without requiring a Python installation.

## Mac OS

Python integrates very well with Mac OS. The following modules are very helpful if you are developing for Apple’s OS.

[**py2app**](https://github.com/ronaldoussoren/py2app): A Python setuptools command that will allow you to make standalone Mac OS X application bundles and plugins from Python scripts.

[**PyObjC**](https://pypi.org/project/pyobjc/): PyObjC is a bridge between Python and Objective-C. It allows full featured Cocoa applications to be written in pure Python.

## USB and Serial Ports

Did you know that using Python, you can access your computer’s USB and Serial ports? The following modules will help when you need to accomplish such tasks.

[**PyUSB**](https://pyusb.github.io/pyusb/) aims to be an easy-to-use Python module to access USB devices. PyUSB relies on a native system library for USB access. Currently, it works out of the box with libusb 0.1, libusb 1.0, libusbx, libusb-win32 and OpenUSB, and works with any Python version starting at 2.4, including Python 3 releases.

[**PySerial**](https://github.com/pyserial/pyserial): Python serial port access library.

[**USPP**](https://sites.google.com/site/ibarona/uspp) is a multi-platform Python module to access serial ports. At the moment, it only works in Windows.