

Python modules on PWF Linux

This is an non-exhaustive list of Python modules on PWF Linux. Most are standard modules for the version of Python we run (2.6) and would be found on any similar Python installation. Those that have been installed specially for PWF Linux are marked with an asterisk.

An alphabetic list of every module shipped with Python can be found at

<http://docs.python.org/modindex.html>

though the quality of the documentation varies wildly.

Please note that these modules are not in alphabetic order, but grouped thematically. This is not meant to be a reference for you to look up what a module does (use the URL above for that) but rather a quick skim read to see what there is.

Name	*	Description
os		access to operating system-specific functions (see the course “Python: Operating System Access” for some of the ways this module can be used)
sys		access to common system functionality
platform		access to underlying platform’s identifying data
subprocess		call external commands and get access to their standard input, standard output, standard error and return code (available in Python 2.4 and later; see the course “Python: Operating System Access” for details)
tempfile		securely generate temporary files and directories (see the course “Python: Further Topics” for details)
getopt		parsing command lines, with <code>--verbosity=4</code> , <code>--verbose</code> and <code>-v</code> style options
math		access to the set of (floating point) mathematical functions defined by the C standard
cmath		the complex equivalent of <code>math</code>
random		pseudo-random number generators for various distributions
numpy	*	A set of functions and types suitable for numerical processing of arrays of various sorts of numbers (integers, floats, complex). (See the course “Python: Interoperation with Fortran” for examples of how this module might be used.) For more details on NumPy, see: http://numpy.scipy.org/ and http://www.scipy.org/Documentation
scipy	*	A scientific computing package built on top of NumPy. For more details on SciPy, see: http://www.scipy.org/Documentation
re		regular expressions (see the course “Python: Regular Expressions” for details)
csv		encoding and decoding of data in comma separated value format as commonly used by spreadsheets and relational databases (see the course “Python: Further Topics” for details)
base64		encoding and decoding for Base64 encoded data, a format commonly used to transfer data files in email
binhex		encoding and decoding for files in binhex4 format, a format allowing representation of Macintosh files in ASCII
uu		encoding and decoding files in uuencode format, allowing arbitrary binary data to be transferred over ASCII-only connections

Name	*	Description
<code>pickle</code>		Serializing and de-serializing Python object structures for storage. (See the course “Python: Checkpointing” for details.)
<code>cPickle</code>		An implementation of the <code>pickle</code> module written in C rather than Python for improved performance. (Both modules provide equivalent structures.) The <code>pickle</code> module is better for testing and debugging; the <code>cPickle</code> module is better at run time. (See the course “Python: Checkpointing” for details.)
<code>Gnuplot</code>	*	Provides an interface to the <code>gnuplot</code> data and function plotting package, allowing the use of <code>gnuplot</code> from within Python. (See the course “Python: Further Topics” for examples of how this module might be used.) For more details on this module, see: http://gnuplot-py.sourceforge.net/
<code>ploticus</code>	*	Provides an interface to the <code>ploticus</code> API, allowing the use of <code>ploticus</code> -related functions from within Python. <code>ploticus</code> is an alternative to <code>gnuplot</code> for producing plots, charts and graphics from data. (See the course “Python: Further Topics” for examples of how this module might be used.) For more details on this module, see: http://www.srcc.lsu.edu/pyploticus.html
<code>matplotlib</code>	*	A plotting library that allows MATLAB®-style plotting in Python. (See the course “Python: Further Topics” for examples of how this module might be used.) For more details on this module, see: http://matplotlib.sourceforge.net/
<code>Image</code>	*	Main module of the Python Imaging Library (PIL). PIL provides fairly powerful image processing capabilities and supports a large number of image file formats. For more details on PIL, see: http://www.pythonware.com/products/pil/index.htm
<code>sqlite3</code>		modules for interfacing to a simple SQL database built around local files (shipped with Python 2.5 and later)
<code>anydbm</code>		generic interface to variants of the DBM database
<code>bz2</code>		interface for the <code>bz2</code> compression library
<code>gzip</code>		interface for the <code>zlib</code> compression library for reading and writing <code>gzip</code> files
<code>zlib</code>		interface for the <code>zlib</code> compression library
<code>zipfile</code>		work with <code>zip</code> files
<code>unittest</code>		The Python unit testing framework. (See the course “Python: Unit Testing” for details.)