

INSTALLING THIRD-PARTY MODULES IN PYTHON

WHAT ARE MODULES?

Modules are basically files which contain definitions that provide helpful functionality. Theoretically, all of you could write any program you wanted to write on your own, without any modules or outside help. But why reinvent the wheel if you don't have to? This is the general gist behind modules. You all have experience using modules such as `tkinter` and `csv`. These modules are part of a library of standard modules that came packaged with your install which the developers have deemed useful or necessary for your daily python usage. However, they can't give you everything. This is where third-party modules come in. When you come across a problem that the standard modules can't solve, there's a good chance somebody else had the same problem before you and created his own module to solve it. The good news? We can take that module (if the developer is kind enough to publish it) and install it for our own use. Cool, right? The bad news, however, is that it's a little more complicated than your average one-click install...

More reading: <http://docs.python.org/3/tutorial/modules.html>

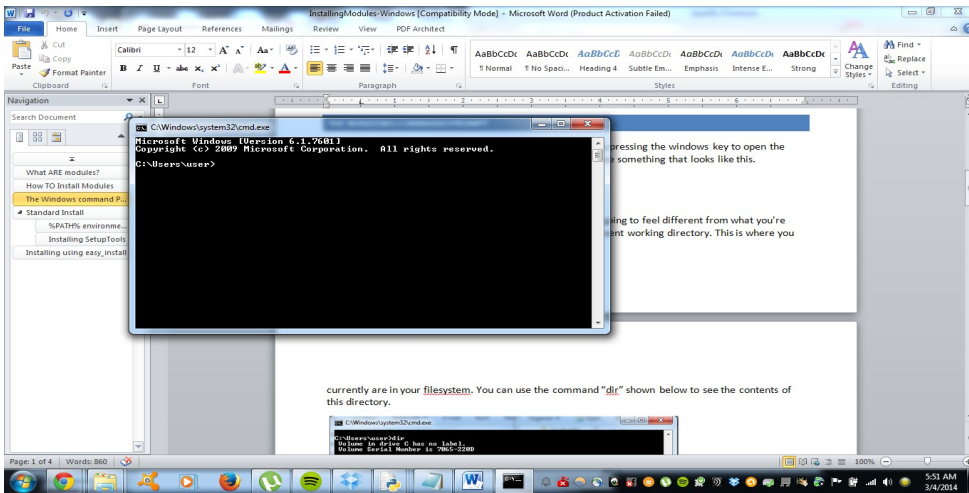
HOW TO INSTALL MODULES

Before we get into the details of actually installing `pymysql`, it'll be good to learn a little more about installing packages to Python in general. This will be useful to know if you ever start using Python more seriously as there are many packages out there which provide you with some functionality you couldn't otherwise get. In this class, we will be working with `pymysql`, a library written purely in Python, to interact with a MySQL database.

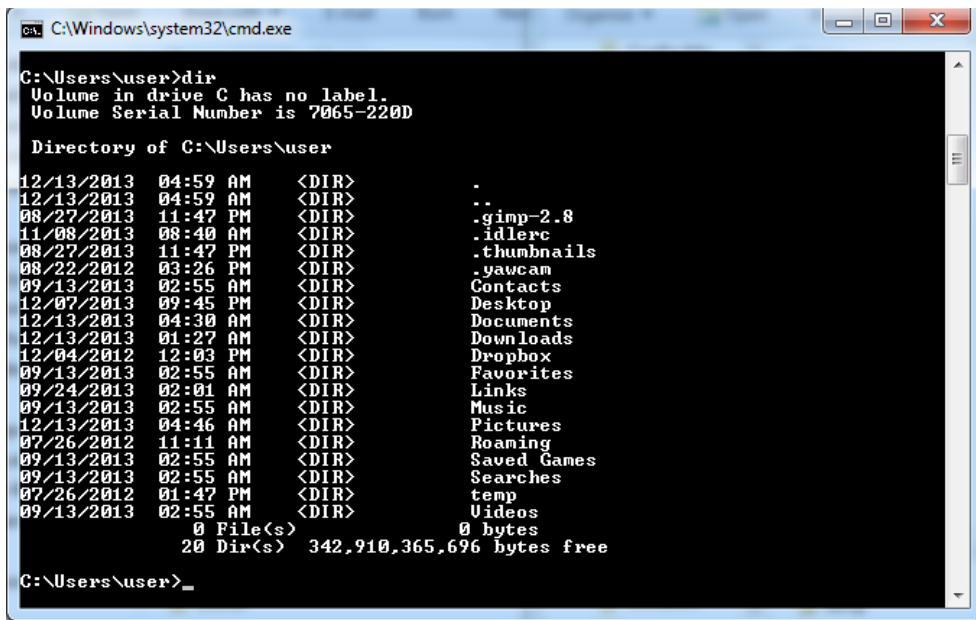
The traditional way to install `pymysql` in this class has been to download the source and run its `setup.py` script to install the package on our system. However, in the interest of giving you the know-how to install other third party libraries should you need them; we will be looking at a more versatile method using the `setuptools` module. You will also need to know the very basics of using the command line.

THE WINDOWS COMMAND PROMPT

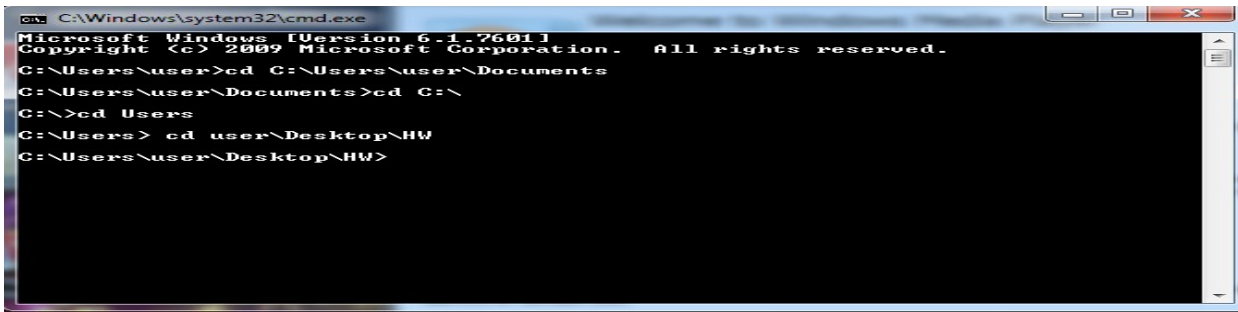
1. Bring up the command prompt. This can be easily done by pressing the windows key to open the start menu, typing "cmd", then pressing enter. You should see something that looks like this.



2. Navigating your file system through the command line is going to feel different from what you're used to. If you look at your prompt, you should see your present working directory. This is where you currently are in your filesystem. You can use the command “dir” shown below to see the contents of this directory.



3. The command “cd” is what you will use to change the present working directory. In order to use it, type “cd directory_path” and hit enter. If you want to cd into a sub-directory you do not need to input the whole path, but can just type “cd directory_name.” The following demonstrates the “cd” command:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\Documents
C:\Users\user\Documents>cd C:\
C:\>cd Users
C:\Users> cd user\Desktop\HW
C:\Users\user\Desktop\HW>
```

STANDARD INSTALL

DOWNLOADING THE MODULES SOURCE DISTRIBUTION

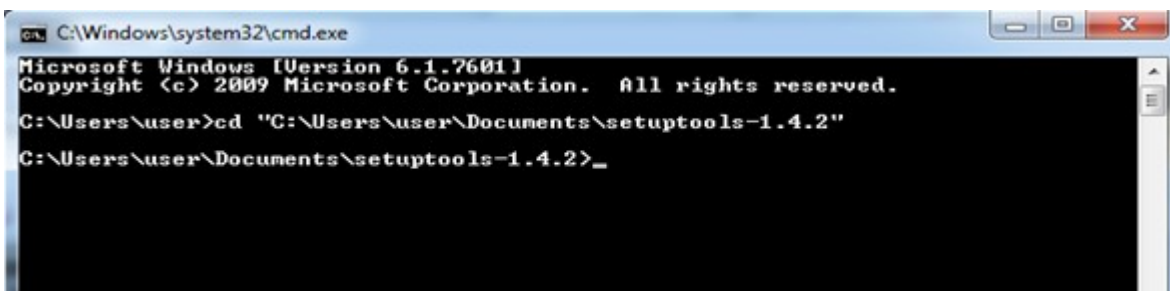
Before you can install the pymysql module, you must download the module's source distribution. Most of the time, module distribution's would be located in a tar archive, which would require a special software (e.g. winrar or 7zip) to extract. We have, however, provided it in a zip file located on the class webpage. Please download and unzip the file.

%PATH% ENVIRONMENT VARIABLE

First off, let's begin by adding the Python bin directory into your computer's PATH environment variable. The PATH is the system variable that your operating system uses to locate needed executables from the command line or Terminal window. In other words, if we type in the name of an executable (in this case, Python), the computer will automatically search for that executable in any directory contained in your system path, so that we don't have to type out Python's file path every time we want to use it. You can use the command prompt to append directories to the PATH variable; however, Python has a script written specifically to add the Python directory for you. Open Python and run the script win_add2path.py located in C:\PythonXX\Tools\Scripts and you're done.

INSTALLING SETUPTOOLS

Now to install setuptools we must first open the command prompt and change the current working directory to the setuptools folder you extracted. You can copy the path easily by holding shift as you right-click the folder and selecting 'copy as path'.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>cd "C:\Users\user\Documents\setuptools-1.4.2"
C:\Users\user\Documents\setuptools-1.4.2>_
```

The command to install the module is "python setup.py install". Type that into the prompt so that it resembles the following and hit enter.

```
C:\Users\user\Documents\setuptools-1.4.2>python setup.py install_
```

Setuptools should now be installed. Run the command “easy_install --version” to make sure. If you get an output resembling the following, setuptools has been installed.

```
C:\Users\user>easy_install --version  
setuptools 2.0
```

INSTALLING USING EASY_INSTALL

That took a lot of work, didn't it? First you had to download the module's source distribution and extract it, then change the directory in the command prompt to the directory you extracted it to, and, only after that was done, could you run the installation command. Now that you have setuptools installed you will be able to download any third party library in the Python Packaging Index: <https://pypi.python.org/pypi> and install it with one simple command. First, re-run the win_add2path.py script from above, close and re-open the command prompt. Then, in order to install pymysql, just run the following command.

```
C:\Users\user>easy_install pymysql_
```

That's it! If everything worked properly, you should be able to pull up a shell and type in import pymysql without error statements. If you're having trouble, ask the nearest TA.

More Info on Easy Install: http://pythonhosted.org//setuptools/easy_install.html