Random note about

if \_\_name\_\_ == "\_\_main\_\_":

(read http://stackoverflow.com/questions/419163/what-does-if-name-main-do)

Expanding a bit on Harley's answer...

When the Python interpreter reads a source file, it executes all of the code found in it. Before executing the code, it will define a few special variables. For example, if the python interpreter is running that module (the source file) as the main program, it sets the special \_\_name\_\_ variable to have a value "\_\_main\_\_". If this file is being imported from another module, \_\_name\_\_ will be set to the module's name.

In the case of your script, let's assume that it's executing as the main function, e.g. you said something like

python threading\_example.py

on the command line. After setting up the special variables, it will execute the import statement and load those modules. It will then evaluate the def block, creating a function object and creating a variable called myfunction that points to the function object. It will then read the if statement and see that \_\_name\_\_ does equal "\_\_main\_\_", so it will execute the block shown there.

One of the reasons for doing this is that sometimes you write a module (a .py file) where it can be executed directly. Alternatively, it can also be imported and used in another module. By doing the main check, you can have that code only execute when you want to run the module as a program and not have it execute when someone just wants to import your module and call your functions themselves.

See [this page](http://ibiblio.org/g2swap/byteofpython/read/module-name.html) for some extra details.