**The ftplib module**

This module contains a *File Transfer Protocol* (FTP) client implementation.

The first example shows how to log in and get a directory listing of the login directory. The **dir** function takes a callback function, which is called once for each line in the server response. The default callback provided by the **ftplib** module simply prints the response to **sys.stdout**.

Note that the format of the directory listing is server dependent (it’s usually the same as the format used by the directory listing utility on the server host platform).

**Example: Using the ftplib module to get a directory listing**

# File: [ftplib-example-1.py](http://effbot.org/librarybook/ftplib/ftplib-example-1.py)

import ftplib

ftp = ftplib.FTP("www.python.org")

ftp.login("anonymous", "ftplib-example-1")

data = []

ftp.dir(data.append)

ftp.quit()

for line in data:

print "-", line

$ python ftplib-example-1.py

- total 34

- drwxrwxr-x 11 root 4127 512 Sep 14 14:18 .

- drwxrwxr-x 11 root 4127 512 Sep 14 14:18 ..

- drwxrwxr-x 2 root 4127 512 Sep 13 15:18 RCS

- lrwxrwxrwx 1 root bin 11 Jun 29 14:34 README -> welcome.msg

- drwxr-xr-x 3 root wheel 512 May 19 1998 bin

- drwxr-sr-x 3 root 1400 512 Jun 9 1997 dev

- drwxrwxr-- 2 root 4127 512 Feb 8 1998 dup

- drwxr-xr-x 3 root wheel 512 May 19 1998 etc

...

Downloading files is easy; just use the appropriate **retr** function. Note that when you download a text file, you have to add line endings yourself. The following function uses a **lambda** expression to do that on the fly.

**Example: Using the ftplib module to retrieve files**

# File: [ftplib-example-2.py](http://effbot.org/librarybook/ftplib/ftplib-example-2.py)

import ftplib

import sys

def gettext(ftp, filename, outfile=None):

# fetch a text file

if outfile is None:

outfile = sys.stdout

# use a lambda to add newlines to the lines read from the server

ftp.retrlines("RETR " + filename, lambda s, w=outfile.write: w(s+"\n"))

def getbinary(ftp, filename, outfile=None):

# fetch a binary file

if outfile is None:

outfile = sys.stdout

ftp.retrbinary("RETR " + filename, outfile.write)

ftp = ftplib.FTP("www.python.org")

ftp.login("anonymous", "ftplib-example-2")

gettext(ftp, "README")

getbinary(ftp, "welcome.msg")

$ python ftplib-example-2.py

WELCOME to python.org, the Python programming language home site.

You are number %N of %M allowed users. Ni!

Python Web site: http://www.python.org/

CONFUSED FTP CLIENT? Try begining your login password with '-' dash.

This turns off continuation messages that may be confusing your client.

...

Finally, here’s a simple example that copies files to the FTP server. This script uses the file extension to figure out if the file is a text file or a binary file:

**Example: Using the ftplib module to store files**

# File: [ftplib-example-3.py](http://effbot.org/librarybook/ftplib/ftplib-example-3.py)

import ftplib

import os

def upload(ftp, file):

ext = os.path.splitext(file)[1]

if ext in (".txt", ".htm", ".html"):

ftp.storlines("STOR " + file, open(file))

else:

ftp.storbinary("STOR " + file, open(file, "rb"), 1024)

ftp = ftplib.FTP("ftp.fbi.gov")

ftp.login("mulder", "trustno1")

upload(ftp, "trixie.zip")

upload(ftp, "file.txt")

upload(ftp, "sightings.jpg")