

# Uploading Files to an FTP Server

The other major task that you do with an FTP server is upload files to it. Python can handle this too. There are actually two methods that you can use for uploading file:

- `storlines` - Used for uploading text files (TXT, HTML, RST)
- `storbinary` - Used for uploading binary files (PDF, XLS, etc)

Let's look at an example of how we might do this:

```
import ftplib

def ftp_upload(ftp_obj, path, ftype='TXT'):
    """
    A function for uploading files to an FTP server
    @param ftp_obj: The file transfer protocol object
    @param path: The path to the file to upload
    """
    if ftype == 'TXT':
        with open(path) as fobj:
            ftp.storlines('STOR ' + path, fobj)
    else:
        with open(path, 'rb') as fobj:
            ftp.storbinary('STOR ' + path, fobj, 1024)

if __name__ == '__main__':
    ftp = ftplib.FTP('host', 'username', 'password')
    ftp.login()

    path = '/path/to/something.txt'
    ftp_upload(ftp, path)

    pdf_path = '/path/to/something.pdf'
    ftp_upload(ftp, pdf_path, ftype='PDF')

    ftp.quit()
```

In this example, we create a function for uploading files. It takes an ftp object, the path of the file we want to upload and the type of the file. Then we do a quick check on the file type to determine if we should use **storlines** or

**storbinary** for our upload process. Finally in our conditional statement at the

bottom, we connect to the FTP server, login and upload a text file and a PDF file. An easy enhancement to add to this is some logic for changing to a specific directory once we're logged in as we probably don't want to just upload files to the root location.

## Wrapping Up

At this point, you should know enough to get started using Python's `ftplib`. It has a lot of other methods that are well worth checking out in Python's documentation on the module. But you now know the basics of listing a directory, navigating the folder structure as well as downloading and uploading files.