

:mod:`ftplib` --- FTP protocol client

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 1); [backlink](#)
Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 4)
Unknown directive type "module".

```
.. module:: ftplib
   :synopsis: FTP protocol client (requires sockets).
```

Source code: `source: Lib/ftplib.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 7); [backlink](#)
Unknown interpreted text role "source".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 9)
Unknown directive type "index".

```
.. index::
   pair: FTP; protocol
   single: FTP; ftplib (standard module)
```

This module defines the class `:class:`FTP`` and a few related items. The `:class:`FTP`` class implements the client side of the FTP protocol. You can use this to write Python programs that perform a variety of automated FTP jobs, such as mirroring other FTP servers. It is also used by the module `:mod:`urllib.request`` to handle URLs that use FTP. For more information on FTP (File Transfer Protocol), see internet [RFC 959](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 15); [backlink](#)
Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 15); [backlink](#)
Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 15); [backlink](#)
Unknown interpreted text role "mod".

The default encoding is UTF-8, following [RFC 2640](#).

Here's a sample session using the `:mod:`ftplib`` module:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 24); [backlink](#)
Unknown interpreted text role "mod".

```
>>> from ftplib import FTP
>>> ftp = FTP('ftp.us.debian.org') # connect to host, default port
>>> ftp.login() # user anonymous, passwd anonymous@
'230 Login successful.'
>>> ftp.cwd('debian') # change into "debian" directory
'250 Directory successfully changed.'
>>> ftp.retrlines('LIST') # list directory contents
-rw-rw-r-- 1 1176 1176 1063 Jun 15 10:18 README
...
drwxr-xr-x 5 1176 1176 4096 Dec 19 2000 pool
drwxr-xr-x 4 1176 1176 4096 Nov 17 2008 project
drwxr-xr-x 3 1176 1176 4096 Oct 10 2012 tools
'226 Directory send OK.'
>>> with open('README', 'wb') as fp:
...     ftp.retrbinary('RETR README', fp.write)
'226 Transfer complete.'
>>> ftp.quit()
'221 Goodbye.'
```

The module defines the following items:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 48)
Invalid class attribute value for "class" directive: "FTP(host=", user=", passwd=", acct=", timeout=None, source_address=None, *, encoding='utf-8')".

```
.. class:: FTP(host='', user='', passwd='', acct='', timeout=None, source_address=None, *, encoding='utf-8')
```

Return a new instance of the `:class:`FTP`` class. When `*host*` is given, the method call `connect(host)` is made. When `*user*` is given, additionally the method call `login(user, passwd, acct)` is made (where `*passwd*` and `*acct*` default to the empty string when not given). The optional `*timeout*` parameter specifies a timeout in seconds for blocking operations like the connection attempt (if is not specified, the global default timeout setting will be used). `*source_address*` is a 2-tuple `((host, port))` for the socket to bind to as its source address before connecting. The `*encoding*` parameter specifies the encoding for directories and filenames.

The `:class:`FTP`` class supports the `:keyword:`with`` statement, e.g.:

```
>>> from ftplib import FTP
>>> with FTP("ftpl.at.proftpd.org") as ftp:
...     ftp.login()
...     ftp.dir()
... # doctest: +SKIP
'230 Anonymous login ok, restrictions apply.'
dr-xr-xr-x 9 ftp ftp 154 May 6 10:43 .
dr-xr-xr-x 9 ftp ftp 154 May 6 10:43 ..
dr-xr-xr-x 5 ftp ftp 4096 May 6 10:43 CentOS
dr-xr-xr-x 3 ftp ftp 18 Jul 10 2008 Fedora
>>>
```

```
.. versionchanged:: 3.2
    Support for the :keyword:`with` statement was added.

.. versionchanged:: 3.3
    *source_address* parameter was added.

.. versionchanged:: 3.9
    If the *timeout* parameter is set to be zero, it will raise a
    :class:`ValueError` to prevent the creation of a non-blocking socket.
    The *encoding* parameter was added, and the default was changed from
    Latin-1 to UTF-8 to follow :rfc:`2640`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 86)

Invalid class attribute value for "class" directive: "FTP_TLS(host='', user='', passwd='', acct='', keyfile=None, certfile=None, context=None, timeout=None, source_address=None, *, encoding='utf-8')".

```
.. class:: FTP_TLS(host='', user='', passwd='', acct='', keyfile=None, certfile=None, context=None, timeout=None, source_address=None, *, encoding='utf-8')
```

A :class:`FTP` subclass which adds TLS support to FTP as described in :rfc:`4217`.

Connect as usual to port 21 implicitly securing the FTP control connection before authenticating. Securing the data connection requires the user to explicitly ask for it by calling the :meth:`prot_p` method. *context* is a :class:`ssl.SSLContext` object which allows bundling SSL configuration options, certificates and private keys into a single (potentially long-lived) structure. Please read :ref:`ssl-security` for best practices.

keyfile and *certfile* are a legacy alternative to *context* -- they can point to PEM-formatted private key and certificate chain files (respectively) for the SSL connection.

```
.. versionadded:: 3.2

.. versionchanged:: 3.3
    *source_address* parameter was added.

.. versionchanged:: 3.4
    The class now supports hostname check with
    :attr:`ssl.SSLContext.check_hostname` and *Server Name Indication* (see
    :data:`ssl.HAS_SNI`).

.. deprecated:: 3.6

    *keyfile* and *certfile* are deprecated in favor of *context*.
    Please use :meth:`ssl.SSLContext.load_cert_chain` instead, or let
    :func:`ssl.create_default_context` select the system's trusted CA
    certificates for you.

.. versionchanged:: 3.9
    If the *timeout* parameter is set to be zero, it will raise a
    :class:`ValueError` to prevent the creation of a non-blocking socket.
    The *encoding* parameter was added, and the default was changed from
    Latin-1 to UTF-8 to follow :rfc:`2640`.
```

Here's a sample session using the :class:`FTP_TLS` class::

```
>>> ftps = FTP_TLS('ftp.pureftpd.org')
>>> ftps.login()
'230 Anonymous user logged in'
>>> ftps.prot_p()
'200 Data protection level set to "private"'
>>> ftps.nlst()
['6jack', 'OpenBSD', 'antilink', 'blogbench', 'bsdcam', 'clockspeed', 'djb dns-jedi', 'docs', 'eaccelerator-jedi', 'favicon.i
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 135)

Unknown directive type "exception".

```
.. exception:: error_reply

    Exception raised when an unexpected reply is received from the server.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 140)

Unknown directive type "exception".

```
.. exception:: error_temp

    Exception raised when an error code signifying a temporary error (response
    codes in the range 400--499) is received.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 146)

Unknown directive type "exception".

```
.. exception:: error_perm

    Exception raised when an error code signifying a permanent error (response
    codes in the range 500--599) is received.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 152)

Unknown directive type "exception".

```
.. exception:: error_proto

    Exception raised when a reply is received from the server that does not fit
    the response specifications of the File Transfer Protocol, i.e. begin with a
    digit in the range 1--5.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (Library) ftplib.rst, line 159)

Unknown directive type "data".

```
.. data:: all_errors
```

```
The set of all exceptions (as a tuple) that methods of :class:`FTP` instances may raise as a result of problems with the FTP connection (as opposed to programming errors made by the caller). This set includes the four exceptions listed above as well as :exc:`OSError` and :exc:`EOFError`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 167)

Unknown directive type "seealso".

```
.. seealso::

    Module :mod:`netrc`
        Parser for the :file:`.netrc` file format. The file :file:`.netrc` is typically used by FTP clients to load user authentication information before prompting the user.
```

FTP Objects

Several methods are available in two flavors: one for handling text files and another for binary files. These are named for the command which is used followed by `lines` for the text version or `binary` for the binary version.

`class:FTP` instances have the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 184); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 187)

Unknown directive type "method".

```
.. method:: FTP.set_debuglevel(level)

    Set the instance's debugging level. This controls the amount of debugging output printed. The default, ``0``, produces no debugging output. A value of ``1`` produces a moderate amount of debugging output, generally a single line per request. A value of ``2`` or higher produces the maximum amount of debugging output, logging each line sent and received on the control connection.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 196)

Unknown directive type "method".

```
.. method:: FTP.connect(host='', port=0, timeout=None, source_address=None)

    Connect to the given host and port. The default port number is ``21``, as specified by the FTP protocol specification. It is rarely needed to specify a different port number. This function should be called only once for each instance; it should not be called at all if a host was given when the instance was created. All other methods can only be used after a connection has been made.

    The optional *timeout* parameter specifies a timeout in seconds for the connection attempt. If no *timeout* is passed, the global default timeout setting will be used.

    *source_address* is a 2-tuple ``(host, port)`` for the socket to bind to as its source address before connecting.

    .. audit-event:: ftplib.connect self,host,port ftplib.FTP.connect

    .. versionchanged:: 3.3
        *source_address* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 216)

Unknown directive type "method".

```
.. method:: FTP.getwelcome()

    Return the welcome message sent by the server in reply to the initial connection. (This message sometimes contains disclaimers or help information that may be relevant to the user.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 223)

Unknown directive type "method".

```
.. method:: FTP.login(user='anonymous', passwd='', acct='')

    Log in as the given *user*. The *passwd* and *acct* parameters are optional and default to the empty string. If no *user* is specified, it defaults to ``'anonymous'``. If *user* is ``'anonymous'``, the default *passwd* is ``'anonymous@'``. This function should be called only once for each instance, after a connection has been established; it should not be called at all if a host and user were given when the instance was created. Most FTP commands are only allowed after the client has logged in. The *acct* parameter supplies "accounting information"; few systems implement this.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 235)

Unknown directive type "method".

```
.. method:: FTP.abort()

    Abort a file transfer that is in progress. Using this does not always work, but it's worth a try.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) ftplib.rst, line 241)

Unknown directive type "method".

```
.. method:: FTP.sendcmd(cmd)

    Send a simple command string to the server and return the response string.

.. audit-event:: ftplib.sendcmd self,cmd ftplib.FTP.sendcmd
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 248)

Unknown directive type "method".

```
.. method:: FTP.voidcmd(cmd)

    Send a simple command string to the server and handle the response. Return
    nothing if a response code corresponding to success (codes in the range
    200--299) is received. Raise :exc:`error_reply` otherwise.

.. audit-event:: ftplib.sendcmd self,cmd ftplib.FTP.voidcmd
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 257)

Unknown directive type "method".

```
.. method:: FTP.retrbinary(cmd, callback, blocksize=8192, rest=None)

    Retrieve a file in binary transfer mode. *cmd* should be an appropriate
    ``RETR`` command: ``RETR filename``. The *callback* function is called for
    each block of data received, with a single bytes argument giving the data
    block. The optional *blocksize* argument specifies the maximum chunk size to
    read on the low-level socket object created to do the actual transfer (which
    will also be the largest size of the data blocks passed to *callback*). A
    reasonable default is chosen. *rest* means the same thing as in the
    :meth:`transfercmd` method.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 269)

Unknown directive type "method".

```
.. method:: FTP.retrlines(cmd, callback=None)

    Retrieve a file or directory listing in the encoding specified by the
    *encoding* parameter at initialization.
    *cmd* should be an appropriate ``RETR`` command (see :meth:`retrbinary`) or
    a command such as ``LIST`` or ``NLST`` (usually just the string ``LIST``).
    ``LIST`` retrieves a list of files and information about those files.
    ``NLST`` retrieves a list of file names.
    The *callback* function is called for each line with a string argument
    containing the line with the trailing CRLF stripped. The default *callback*
    prints the line to ``sys.stdout``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 282)

Unknown directive type "method".

```
.. method:: FTP.set_pasv(val)

    Enable "passive" mode if *val* is true, otherwise disable passive mode.
    Passive mode is on by default.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 288)

Unknown directive type "method".

```
.. method:: FTP.storbinary(cmd, fp, blocksize=8192, callback=None, rest=None)

    Store a file in binary transfer mode. *cmd* should be an appropriate
    ``STOR`` command: ``STOR filename``. *fp* is a :term:`file object`
    (opened in binary mode) which is read until EOF using its :meth:`~io.IOBase.read`
    method in blocks of size *blocksize* to provide the data to be stored.
    The *blocksize* argument defaults to 8192. *callback* is an optional single
    parameter callable that is called on each block of data after it is sent.
    *rest* means the same thing as in the :meth:`transfercmd` method.

.. versionchanged:: 3.2
    *rest* parameter added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 302)

Unknown directive type "method".

```
.. method:: FTP.storlines(cmd, fp, callback=None)

    Store a file in line mode. *cmd* should be an appropriate
    ``STOR`` command (see :meth:`storbinary`). Lines are read until EOF from the
    :term:`file object` *fp* (opened in binary mode) using its :meth:`~io.IOBase.readline`
    method to provide the data to be stored. *callback* is an optional single
    parameter callable that is called on each line after it is sent.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 311)

Unknown directive type "method".

```
.. method:: FTP.transfercmd(cmd, rest=None)

    Initiate a transfer over the data connection. If the transfer is active, send an
    ``EPRT`` or ``PORT`` command and the transfer command specified by *cmd*, and
    accept the connection. If the server is passive, send an ``EPSV`` or ``PASV``
    command, connect to it, and start the transfer command. Either way, return the
    socket for the connection.

    If optional *rest* is given, a ``REST`` command is sent to the server, passing
```

`*rest*` as an argument. `*rest*` is usually a byte offset into the requested file, telling the server to restart sending the file's bytes at the requested offset, skipping over the initial bytes. Note however that the `:meth:'transfercmd'` method converts `*rest*` to a string with the `*encoding*` parameter specified at initialization, but no check is performed on the string's contents. If the server does not recognize the ```REST``` command, an `:exc:'error_reply'` exception will be raised. If this happens, simply call `:meth:'transfercmd'` without a `*rest*` argument.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 330)

Unknown directive type "method".

```
.. method:: FTP.ntransfercmd(cmd, rest=None)
```

Like `:meth:'transfercmd'`, but returns a tuple of the data connection and the expected size of the data. If the expected size could not be computed, ```None``` will be returned as the expected size. `*cmd*` and `*rest*` means the same thing as in `:meth:'transfercmd'`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 338)

Unknown directive type "method".

```
.. method:: FTP.mlsd(path="", facts=[])
```

List a directory in a standardized format by using ```MLSD``` command (`:rfc:'3659'`). If `*path*` is omitted the current directory is assumed. `*facts*` is a list of strings representing the type of information desired (e.g. ```["type", "size", "perm"]```). Return a generator object yielding a tuple of two elements for every file found in path. First element is the file name, the second one is a dictionary containing facts about the file name. Content of this dictionary might be limited by the `*facts*` argument but server is not guaranteed to return all requested facts.

```
.. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 352)

Unknown directive type "method".

```
.. method:: FTP.nlst(argument[, ...])
```

Return a list of file names as returned by the ```NLST``` command. The optional `*argument*` is a directory to list (default is the current server directory). Multiple arguments can be used to pass non-standard options to the ```NLST``` command.

```
.. note:: If your server supports the command, :meth:'mlsd' offers a better API.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 362)

Unknown directive type "method".

```
.. method:: FTP.dir(argument[, ...])
```

Produce a directory listing as returned by the ```LIST``` command, printing it to standard output. The optional `*argument*` is a directory to list (default is the current server directory). Multiple arguments can be used to pass non-standard options to the ```LIST``` command. If the last argument is a function, it is used as a `*callback*` function as for `:meth:'retrlines'`; the default prints to ```sys.stdout```. This method returns ```None```.

```
.. note:: If your server supports the command, :meth:'mlsd' offers a better API.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 374)

Unknown directive type "method".

```
.. method:: FTP.rename(fromname, toname)
```

Rename file `*fromname*` on the server to `*tname*`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 379)

Unknown directive type "method".

```
.. method:: FTP.delete(filename)
```

Remove the file named `*filename*` from the server. If successful, returns the text of the response, otherwise raises `:exc:'error_perm'` on permission errors or `:exc:'error_reply'` on other errors.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 386)

Unknown directive type "method".

```
.. method:: FTP.cwd(pathname)
```

Set the current directory on the server.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 391)

Unknown directive type "method".

```
.. method:: FTP.mkd(pathname)
```

Create a new directory on the server.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 396)

Unknown directive type "method".

```
.. method:: FTP.pwd()

    Return the pathname of the current directory on the server.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 401)

Unknown directive type "method".

```
.. method:: FTP.rmd(dirname)

    Remove the directory named *dirname* on the server.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 406)

Unknown directive type "method".

```
.. method:: FTP.size(filename)

    Request the size of the file named *filename* on the server. On success, the
    size of the file is returned as an integer, otherwise ``None`` is returned.
    Note that the ``SIZE`` command is not standardized, but is supported by many
    common server implementations.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 414)

Unknown directive type "method".

```
.. method:: FTP.quit()

    Send a ``QUIT`` command to the server and close the connection. This is the
    "polite" way to close a connection, but it may raise an exception if the server
    responds with an error to the ``QUIT`` command. This implies a call to the
    :meth:`close` method which renders the :class:`FTP` instance useless for
    subsequent calls (see below).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 423)

Unknown directive type "method".

```
.. method:: FTP.close()

    Close the connection unilaterally. This should not be applied to an already
    closed connection such as after a successful call to :meth:`~FTP.quit`.
    After this call the :class:`FTP` instance should not be used any more (after
    a call to :meth:`close` or :meth:`~FTP.quit` you cannot reopen the
    connection by issuing another :meth:`login` method).
```

FTP_TLS Objects

:class:`FTP_TLS` class inherits from :class:`FTP`, defining these additional objects:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 435); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 435); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 437)

Unknown directive type "attribute".

```
.. attribute:: FTP_TLS.ssl_version

    The SSL version to use (defaults to :attr:`ssl.PROTOCOL_SSLv23`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 441)

Unknown directive type "method".

```
.. method:: FTP_TLS.auth()

    Set up a secure control connection by using TLS or SSL, depending on what
    is specified in the :attr:`ssl_version` attribute.

    .. versionchanged:: 3.4
        The method now supports hostname check with
        :attr:`ssl.SSLContext.check_hostname` and *Server Name Indication* (see
        :data:`ssl.HAS_SNI`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 451)

Unknown directive type "method".

```
.. method:: FTP_TLS.ccc()

    Revert control channel back to plaintext. This can be useful to take
    advantage of firewalls that know how to handle NAT with non-secure FTP
    without opening fixed ports.
```

```
.. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 459)

Unknown directive type "method".

```
.. method:: FTP_TLS.prot_p()
```

Set up secure data connection.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) (Doc) (library) ftplib.rst, line 463)

Unknown directive type "method".

```
.. method:: FTP_TLS.prot_c()
```

Set up clear text data connection.