The following scripts are written to demonstrate Python Internet communication protocols. The following modules are explored.

‘mailbox’: manipulate mailboxes in various formats.

This module defines two classes, Mailbox and Message, for accessing and manipulating on-disk mailboxes and the messages they contain. Mailbox offers a dictionary-like mapping from keys to messages. Message extends the email.message module’s Message class with format-specific state and behavior. Supported mailbox formats are Maildir, mbox, MH, Babyl, and MMDF.

‘smtplib’: SMTP protocol client.

The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

‘poplib’: POP3 protocol client.

This module defines a class, POP3, which encapsulates a connection to a POP3 server and implements the protocol as defined in RFC 1939. The POP3 class supports both the minimal and optional command sets from RFC 1939. The POP3 class also supports the STLS command introduced in RFC 2595 to enable encrypted communication on an already established connection. Additionally, this module provides a class POP3\_SSL, which provides support for connecting to POP3 servers that use SSL as an underlying protocol layer.

‘imaplib’: IMAP4 protocol client.

This module defines three classes, IMAP4, IMAP4\_SSL and IMAP4\_stream, which encapsulate a connection to an IMAP4 server and implement a large subset of the IMAP4rev1 client protocol as defined in RFC 2060. It is backward compatible with IMAP4 (RFC 1730) servers, but note that the STATUS command is not supported in IMAP4.

‘nntplib’: NNTP protocol client.

This module defines the class NNTP which implements the client side of the Network News Transfer Protocol. It can be used to implement a news reader or poster, or automated news processors. It is compatible with RFC 3977 as well as the older RFC 977 and RFC 2980.

Compiled and presented by Vakindu Philliam.