This series of scripts is written to demonstrate the use of Python’s File 'io' module.

The io module is the core of tools for working with streams.

It provides Python’s main facilities for dealing with various types of I/O.

There are three main types of I/O: text I/O, binary I/O and raw I/O.

Text I/O:

Text I/O expects and produces ‘str’ objects.

This means that whenever the backing store is natively made of bytes (such as in the case of a file), encoding and decoding of data is made transparently as well as optional translation of platform-specific newline characters.

Binary I/O:

Binary I/O (also called buffered I/O) expects bytes-like objects and produces bytes objects.

No encoding, decoding, or newline translation is performed.

This category of streams can be used for all kinds of non-text data, and also when manual control over the handling of text data is desired.

Raw I/O:

Raw I/O (also called unbuffered I/O) is generally used as a low-level building-block for binary and text streams; it is rarely useful to directly manipulate a raw stream from user code.

Nevertheless, you can create a raw stream by opening a file in binary mode with buffering disabled: