

Binary Files

Binary files do not store data as text, but as binary-encoded values. This is usually done to save space, particularly when human-readability is unimportant. Binary-encoded files are typically not viewable in a standard text editor.

Opening a binary file for reading or writing is done by appending “b” as an argument to either “r” or “w”. To read n bytes, use `read(n)`. Alternatively, calling `read()` with no arguments will read the entire file into a bytes object for later processing.

To write to a file, call `write(b)` where b is some sequence of bytes (expressed in hexadecimal). To convert an existing list of integers, L , to a bytes object, use `bytes(L)`. While encoded as bytes, referencing a specific byte via its index will treat it as an integer. A string of bytes can also be specified manually by prefixing the string with `b`, as in `byte_string = b'xyz'`. An existing string can be encoded as a sequence of bytes by specifying an encoding, such as `utf8`. When printed to the screen, each byte will be displayed as either a printable ASCII character for integer values less than 128, or as its hexadecimal value for non-printable characters or those greater than 128.

All bytes objects are immutable. If there is a need to change individual bytes, a bytearray can be used instead by calling `bytearray(B)` where B is a sequence of bytes. Bytearrays are mutable, so things like slicing, removing, appending and so on will work.

Common Commands For Working With Binary Files

Description	Command
Open a binary file for reading.	<code>f = open("FILENAME", "rb")</code>
Open a binary file for writing.	<code>f = open("FILENAME", "wb")</code>
Open a binary file for appending.	<code>f = open("FILENAME", "ab")</code>
Read N bytes of data from a binary file.	<code>data = f.read(N)</code>
Write some bytes of data to a binary file.	<code>f.write(BYTES)</code>
Specify a string, s , as a sequence of bytes.	<code>s = b'string'</code>
Encode an existing string, s , as a sequence of bytes.	<code>B = bytes(s, ENCODING)</code>
Convert a list of integers, L , to a sequence of bytes.	<code>B = bytes(L)</code>
Create a mutable sequence of bytes from a list, L .	<code>B = bytearray(L)</code>
Close a file when finished.	<code>f.close()</code>

Binary Files

Answer the following questions.

1. Describe a scenario where it would be better to write data to a binary file, rather than a text file.
2. For the sequence of bytes below, what is the integer value for $B[i]$, where $0 \leq i \leq 3$?

`B = b'\x07\x1dC$'`

Write programs that accomplish each task. Use proper conventions for variable names, input prompts, output statements, and program structure. You may assume that all files contain data in the formats specified. You should download the file `binary_files.zip` for use with your programs.

3. A Portable Network Graphics (PNG) file always begins with a specific sequence of four bytes in its header: 0x89, followed by 0x50, 0x4E and 0x47 (the letters 'P', 'N' and 'G'). Download a PNG file, then read the first four bytes to verify that it is valid.
4. Read a file containing the binary representation of 1 000 positive integers between 1-100, then determine the number of values between 20 and 50 inclusive.
5. Modify your password program from the Writing To Text Files handout so that the encrypted password is stored in a binary-encoded format instead. For example, the password 'D5b', which is encrypted as 'Äµâ', would be encoded as `\xc4\xb5\xe2` using cp1252 (Windows) encoding, or as `\xc3\x84\xc2\xb5\xc3\xa2` using utf8 (*nix) encoding.
6. **Digital forensics** is a specialized field within forensic science that is concerned with recovering data found in digital devices. For example, investigators often salvage images and messages that were supposedly deleted. Given the file `data.bin`, write a program that extracts a JPEG image from the binary data within. All JPEG images begin with the byte sequence `\xff\xd8\xff\xe0`, and end with the sequence `\xff\xd9`. Your task is to find these sequences, then create a new binary file containing them and all data in between. Your resulting file should be an image that can be viewed in any image-viewing application. Hint: all string operations, such as the `find` method and slicing, can be used with sequences of bytes.
7. A 16x16 icon is encoded in a binary file. The colour of each pixel is specified by an RGB triplet (R, G, B), where R, G and B are values 0-255. For example, (182, 114, 181) produces a light purple. Write a program that reads information from a file representing an icon encoded in this fashion, and draws it to the terminal. Use the Unicode character `\u2588` (see the page https://en.wikipedia.org/wiki/Block_Elements) to draw a box for each pixel, and set the colour of this box using the `fg` method of the 'sty' module (see <https://pypi.org/project/sty>) or a similar one. Test your program using the sample icon provided. Note: you will probably need to launch python from the command line (terminal) to see the colours.