

Python Files, Persistence & Binary I/O — Q&A; (Set 18)

Q1. Differences between text and binary files (one paragraph)

Text files handle str with automatic encoding/decoding and newline translation. Binary files handle raw bytes with no encoding or newline adjustments, giving exact byte control. Offsets are reliable in binary, less so in text due to encodings and newline translations.

Q2. When prefer text vs binary?

Text: human-readable configs, logs, CSV, JSON, YAML, reports. Binary: images, audio, video, protocol frames, fixed-width records, high-performance I/O, serialized objects.

Q3. Issues with writing Python int directly to binary

Python ints are arbitrary precision. Direct writing requires choices: byte length, endianness, signedness, padding. Naïve binary writes hurt portability. Use `int.to_bytes/from_bytes` or `struct.pack/unpack`.

Q4. Benefit of with vs manual open/close

`with open(...)` ensures deterministic cleanup (file closed even on exceptions), avoids leaks, and is canonical.

Q5. Newlines when reading/writing text

Reading: `readline()` includes trailing `\n` (unless EOF without newline). Writing: `write()` does not append newline; must add `\n` manually or use `print(file=...)`.

Q6. File ops for random access

Use `f.seek(offset, whence)` and `f.tell()` for positioning. Binary mode ('rb') recommended. For large/mapped files, use `mmap`. `os.lseek` is a low-level alternative.

Q7. When you'll use struct

When packing/unpacking C-style binary records: parsing headers (PNG, WAV), network packets, sensor data, or cross-language binary formats.

Q8. When is pickling best?

For quick Python-to-Python persistence of trusted data (object graphs, caches, checkpoints). Never unpickle untrusted data; portability across Python versions is limited.

Q9. When to use *shelve*

When you want a dictionary-like persistent store: *shelve* maps string keys to pickled Python objects. Good for simple state or metadata DBs without schema design.

Q10. Special restriction with *shelve* vs *dicts*

Keys must be strings; values must be picklable. Mutating retrieved mutable values not auto-saved unless reassigned or opened with `writeback=True`.