

**Q1. Describe the differences between text and binary files in a single paragraph.**

Ans: - Text files and binary files differ in how they store and represent data. Text files store data as ASCII characters and are organized around lines, each ending with a newline character. They are easily readable and modifiable as the content is human-readable. Binary files, on the other hand, store data in binary format, often used for numeric information. They are not easily readable or modifiable as the content looks like encrypted content<sup>1</sup>. Binary files can be more efficient in terms of memory usage as they store data as per its memory size.

**Q2. What are some scenarios where using text files will be the better option? When would you like to use binary files instead of text files?**

Ans: - Text files are a better option when the data is simple, needs to be human-readable, and doesn't require much space. They are also useful when the data needs to be easily transferable between different systems. Binary files are preferable when dealing with complex data structures or large volumes of data. They are also used when the data needs to be processed by a program rather than a human.

**Q3. What are some of the issues with using binary operations to read and write a Python integer directly to disc?**

Ans: - Using binary operations to read and write a Python integer directly to disk can lead to issues such as data corruption and difficulties in data interpretation. If the binary data is not properly formatted or if there is an error in the binary file, it can lead to incorrect results or even program crashes.

**Q4. Describe a benefit of using the with keyword instead of explicitly opening a file.**

Ans: - The benefit of using the with a keyword instead of explicitly opening a file in Python is that it ensures the file is properly closed after operations are completed, even if an error occurs within the block. This helps prevent resource leaks and makes the code cleaner and more readable.

**Q5. Does Python have the trailing newline while reading a line of text? Does Python append a newline when you write a line of text?**

Ans: - Yes, Python does include the trailing newline when reading a line of text from a file. However, Python does not automatically append a newline when you write a line of text to a file. You need to explicitly include the newline character (\n) in the string that you are writing if you want a newline to be added.

**Q6. What file operations enable for random-access operation?**

Ans: - The seek() and tell() file operations enable random-access operation. The seek() operation moves the file pointer to a specified location, allowing for reading or writing at any position in the file. The tell() operation returns the current position of the file pointer

**Q7. When do you think you'll use the struct package the most?**

Ans: - The struct package in Python is most useful when you need to convert between Python values and C structs represented as Python bytes objects. This is particularly handy when dealing with binary data for things like network connections, file manipulations, and handling of complex data structures.

**Q8. When is pickling the best option?**

Ans: - Pickling is the best option when you need to serialize and deserialize complex Python objects, especially when you need to store or transmit these objects for later use. It's important to note that pickling should not be used for long-term storage as the format is not guaranteed to be the same across different Python versions.

**Q9. When will it be best to use the shelve package?**

Ans: - The shelve package in Python is best used when you need a simple persistent storage option for Python objects when a relational database might be overkill<sup>11</sup>. It provides a dictionary-like API and can handle arbitrary Python objects.

**Q10. What is a special restriction when using the shelve package, as opposed to using other data dictionaries?**

Ans: - A special restriction when using the shelve package, as opposed to using other data dictionaries, is that only string data can be used for keys. Additionally, it's important to note that the shelve module is backed by pickle, so it is insecure to load a shelf from an untrusted source.