Reflection Questions

1. Imagine you're having a conversation with a future colleague about whether to use the iPython Shell instead of Python's default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

The Ipython shell's code text is easier to read due to syntax highlighting by displaying different features of your code in contrasting fonts and colors. The Ipython shell auto-indents text for nested statements. It also allows you to test small chunks of code quickly and easily.

2. Python has a host of different data types that allow you to store and organize information. List 4 examples of data types that Python recognizes, briefly define them, and indicate whether they are scalar or non-scalar. Data type Definition Scalar or NonScalar?

Int(Integer): non-rational numbers from zero to infinity in both directions(negative and positive). This data type is Scalar.
Boolean: A data type of value either 'True' or 'False' which is mainly used to store output of a condition checked.
This data type is Scalar
String(str): An array of characters and symbols enclosed in either double or single quotes.
This data type is non-Scalar
Tuples: A linear array of multiple values of any data type stored.
This data type is non-Scalar

3. A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.

They are both similar as they both are arrays of stored data of any type. List elements can be modified. Tuples cannot be modified after it is defined, only reassigned.

4.**Imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language learning app beyond vocabulary memorization.**

I would use the Dictionary structure to store the information of each vocabulary in cards. With dictionaries having the advantage of storing the data in key:value pairs. It would make it easy to index/modify an attribute by using the key assigned.