Q1. In Python 3.X, what are the names and functions of string object types?

Ans1- In Python 3.X, the names and functions of string object types are:

str: Represents Unicode text strings.

bytes: Represents sequences of raw binary data.

bytearray: Mutable version of bytes.

Q2. How do the string forms in Python 3.X vary in terms of operations?

Ans2- The string forms in Python 3.X vary in terms of operations:

str supports text operations like encoding, decoding, and string manipulation.

bytes and bytearray support binary data operations like indexing and slicing.

Q3. In 3.X, how do you put non-ASCII Unicode characters in a string?

Ans3- In Python 3.X, you can include non-ASCII Unicode characters in a string by simply including them directly in the string, as Python 3.X strings are Unicode by default. For example, my\_string = "Café".

Q4. In Python 3.X, what are the key differences between text-mode and binary-mode files?

Ans4- In Python 3.X, the key differences between text-mode and binary-mode files are:

Text-mode files (opened with 't' flag) perform newline translation, while binary-mode files (opened with 'b' flag) do not.

Text-mode files use Unicode encoding/decoding, whereas binary-mode files treat data as raw bytes.

Q5. How can you interpret a Unicode text file containing text encoded in a different encoding than your platform's default?

Ans5- To interpret a Unicode text file containing text encoded in a different encoding than your platform's default, you can specify the desired encoding when opening the file using the open() function. For example: with open('file.txt', 'r', encoding='utf-8') as file.

Q6. What is the best way to make a Unicode text file in a particular encoding format?

Ans6- The best way to create a Unicode text file in a particular encoding format is to specify the encoding when opening the file for writing. For example: with open('file.txt', 'w', encoding='utf-8') as file.

Q7. What qualifies ASCII text as a form of Unicode text?

Ans7- ASCII text qualifies as a form of Unicode text when all the characters in the text fall within the ASCII character set, which is a subset of Unicode. ASCII characters are represented in the Unicode standard from U+0000 to U+007F.

Q8. How much of an effect does the change in string types in Python 3.X have on your code?

Ans8- The change in string types in Python 3.X can have a significant effect on your code, especially if you're working with text data and need to handle encodings explicitly. Code that relies on implicit conversions between bytes and strings may need adjustments to ensure compatibility with Python 3.X's stricter separation between text and binary data.