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

- Strings are a special type of a python class. As objects, in a class, you can call methods on string objects using the. methodName() notation. The string class is available by default in python, so you do not need an import statement to use the object interface to strings. The string comparison operator in python is used to compare two strings.
“==” operator returns Boolean True if two strings are the same and return Boolean False if two strings are not the same. “!= ” operator returns Boolean True if two strings are not the same and return Boolean False if two strings are the same.

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

- With the release of Python 3.6, we were introduced to F-strings. As their name suggests, F-strings begin with "f" followed by a string literal. We can insert values dynamically into our strings with an identifier wrapped in curly braces, just like we did in the format () method.

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

- Use str. encode () to convert a Unicode string to an ASCII string. Call str. encode (encoding, errors) with encoding as "ASCII" and errors as "ignore" to return an ASCII representation of a Unicode string str.

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

- A text file stores data in the form of alphabets, digits, and other special symbols by storing their ASCII values and are in a human-readable format. ... whereas binary file contains a sequence or a collection of bytes which are not in a human-readable format. text files follow some simple rules whereas binary files do not.

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

- Unicode is just another type of character encoding; it is still a lookup of bits -> characters. Unicode encoding schemes like UTF-8 are more efficient in how they use their bits. With UTF-8 if a character can be represented with 1 byte that is all it will use. If a character needs 4 bytes, it will get 4 bytes

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

- write () to write unicode text to a text file. Call str. encode(encoding) with encoding set to "utf8" to encode str. Call open (file, mode) to open a file with mode set to "wb".

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

- Unicode represents many characters such as letters of various languages, mathematical symbols, historical scripts, etc. ASCII represents a specific number of characters such as uppercase and lowercase letters of English language, digits, and symbols.

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

- Python strings are "immutable" which means they cannot be changed after they are created. Since strings cannot be changed, we construct *new* strings as we go to represent computed values.