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

Python String casefold() Method.

Python String center() Method.

Python String count() Method.

Python Strings encode() method.

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

**string assignment, string repetition, string slicing, string concatenation, string comparison, string formatting, membership, escape sequence**

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

**Python requires explicit encoding and decoding of strings into Unicode**. In IBM® SPSS® Modeler, Python scripts are assumed to be encoded in UTF-8, which is a standard Unicode encoding that supports non-ASCII characters.

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

**a text file contains textual information in the form of alphabets, digits and special characters or symbols.** **On the other hand, a binary file contains bytes or a compiled version of a text file**.

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

To store text as binary data, you must specify an encoding for that text. The process of converting from a sequence of bytes (i.e. binary data) to a sequence of code points (i.e. text data) is decoding, while the reverse process is encoding.

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

encode method does, and the result of encoding a unicode string is a bytestring (a str type.) You should either **use normal open() and encode the unicode yourself**, or (usually a better idea) use codecs. open() and not encode the data yourself

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

For backward compatibility, **the first 128 Unicode characters point to ASCII characters**. And since UTF-8 encodes each of those characters using 1-byte. ASCII is essentially just UTF-8, or we can say that ASCII is a subset of Unicode.

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

One of the most noticeable changes in Python 3.0 is the mutation of string object types. In a nutshell, 2.X's str and unicode types have morphed into 3.X's bytes and str types, and a new mutable bytearray type has been added. Especially if you process data that is either Unicode or binary in nature, this can have substantial impacts on your code.