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

int , float , complex : the numerical types.

bool : the boolean type. True and False are the only boolean-type objects.

NoneType : the “null” type; None is the only object that belongs to this type.

str : the string type.

list : the list type.

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

capitalize()=Converts the first character to upper case

casefold()=Converts string into lower case

center()=Returns a centered string

count()=Returns the number of times a specified value occurs in a string

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

The line ustring = u'A unicode \u018e string \xf1' creates a Unicode string with 20 characters.

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

The bits in text files represent characters, while the bits in binary files represent custom data. Binary files typically contain a sequence of bytes or ordered groupings of eight bits.

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

decode() is a method specified in Strings in Python. It accepts the encoding of the encoded string to decode it and returns the original string.

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

First we use write() to write unicode text to a text file. We need to call str. encode(encoding) with encoding set to "utf8" to encode str. We need to call open(file, mode) to open a file with mode set to "wb".

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

UTF-8 encodes each of these characters with a single byte, any ASCII text is also a UTF-8 text. Unicode is a superset of ASCII.

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

Strings are immutable, thus strings cannot be changed once assigned.