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

str.upper() and str.lower().str.join(), str.split(), and str.replace()

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

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

s.replace('Â ', '')

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

Text mode, usually by default, and binary mode. Obviously, in text mode, the program writes data to file as text characters, and in binary mode, the program writes data to files as 0/1 bits

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

*All* text *( str ) is* Unicode *by* default*.* Encoded Unicode text *is represented as binary data ( bytes ).*

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

Use str. encode() and file. write() to write unicode text to a text file

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

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?