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

Ans=>slice(),str(),tuple(),sum()

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

Ans=> we can use string formatting with f string.

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

Ans=>we can replace them with replace keywords.

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

Ans=> This mode is about conversion of line endings. When reading in text mode, the platform's native line endings ( \r\n on Windows) are converted to Python's Unix-style \n line endings. When writing in text mode, the reverse happens. In binary mode, no such conversion is done

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

Ans=> file.open('unicode.rst', encoding='utf-8')

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

Ans=> unicode\_text = u'ʑʒʓʔʕʗʘʙʚʛʜʝʞ'

encoded\_unicode = unicode\_text.encode("utf8")

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

Ans=> The string qualifies as ASCII-Only

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

Ans=>no effect