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

Ans. Str.islower() str.isspace() str.replace() str.isupper()

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

Ans. Islower() and isupper() checks whether a string is in lower or upper case

Lower() or upper() converts string to lower and upper case respectively

Replace() replaces given character with another character in a string

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

Ans. Using \u followed by the nos representing that Unicode

Eg. r='\u03c0'

print(r)

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

Ans. A text file consists of human readable characters, which can be opened by any text editor. On the other hand, binary files are made up of non-human readable characters and symbols, which require specific programs to access its contents.

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

Ans. Click the File > "Save As" menu. The "Save As" dialog box comes up. 3. Enter notepad\_utf-16le as the new file name and select "Unicode" option in the Encoding field.

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

Ans. . Click the File > "Save As" menu. The "Save As" dialog box comes up. 3. Enter notepad\_utf-16le as the new file name and select "Unicode" option in the Encoding field.

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

Ans.  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?

Ans. If we try to convert string data type to int or float it throws value error.