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

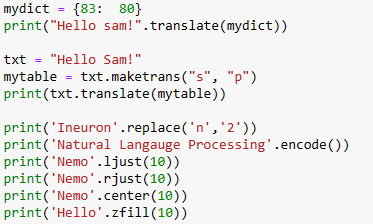
**Ans: The following are the the names and functions of string object types in python 3.X**

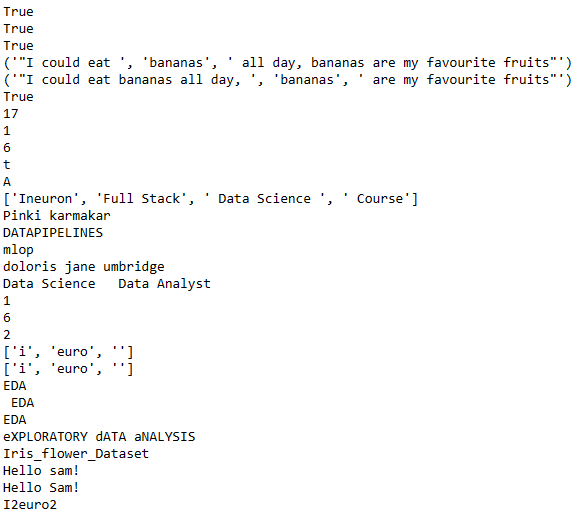
* **<string> . isdecimal ( ) → Returns True if all characters in a string are decimal.**
* **<string> . isalnum ( ) → Returns True if all characters in the string are AlphaNumeric**
* **<string> . istitle ( ) →Returns True if first character in a string is in uppercase.**
* **<string> . partition(<sub\_string>) → split string at first occurance of sub string and returns a tuple of 3 elements**
* **<string> . rpartition (<sub\_string>)→ splits string at last occurance of sub string and returns a tuple of 3 elements.**
* **<string> . isidentifier ( ) → Returns true if give string is a valid identifier name.**
* **len<string> → returns the length of the given string.**
* **<string>.index(<sub\_string>)→return the lowest index of substring if substring is found in the string.**
* **<string> .rindex(<sub\_string>)→return the higest index of substring if substring is found in the string.**
* **max(<string>)→ returns the highest Alphabetical character in the string as per ASCII**
* **min(<string>)→ returns the lowest Alphabetical character in the string as per ASCII**
* **<string> . splitlines( ) → returns a list of lines in the string.**
* **<string> . capitalize( ) →returns the string with first character capitalized.**
* **<string>. Upper()→ return the string with all characters in uppercase.**
* **<string>. Lower( ) → returns the string with all characters in lowercase**
* **<string>.casefold ( ) → returns the string in lowercase which can be used for caseless comparisions.**
* **<string> . expandtabs(no\_of\_spaces)→ replace tabs in astring with specified no of spaces default is 8**
* **<string> . find(<sub\_string>)→ returns lowest inde of substring if substring is found in the string else returns -1**
* **<string > . rfind(< sub\_string>)→returns highest index of substring if substring is found in the stringelse returns -1**
* **<string> . count(<char>)→returns the no of occurances of the char in the given string**
* **<string> . split(<sep>)→returns list of words separated by given sep else separated by whitespace.**
* **<string> . rsplit(<sep>)→ returns list of words separated by given sep else separated by whitespace scanning from end.**
* **<string> . lstrip()→returns a copy of where trailed whitespaces are removed.**
* **<string> .rstrip()→ returns a copy of where trailed whitespaces are removed.**
* **<string> .rstrip()→ returns a copy of where trailed whitespaces are removed.**
* **<string> .rstrip()→ returns a copy of where trailed whitespaces are removed.**
* **<string> .strip()→ returns a copy of where both leading and trailed whitespaces are removed.**
* **<string> . swapcase ( ) → swaps lowercase characters with uppercase and vice versa.**
* **<sep> . join(<list>) → concatenates a list or tuple of words with intervening occurrences of sep.**
* **<string> . translate(<mapping\_table>)→ translates the characters using table.**
* **<string> . maketrans(<dict>)→ creating a mapping translation table usable for**

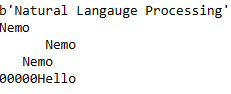
**<string> .translate(<mapping\_table>)**

* **<string> . replace(<char\_1>,<char\_2>)→ replace all occurances of char\_1 with char\_2 in string.**
* **<string> .encode ( ) → Encodes string into any encoding supported by python. Default encoding is UTF-8.**
* **<string> .ljust(<no\_of\_spaces>)→left –justify in a field of given width.**
* **<string> . rjust(<no\_of\_spaces>)→ right-justify in a field of given width.**
* **<string> . center(<no\_of\_spaces>)→ center –justify in a field of given width.**
* **<string> .zfill(<length>)→ zfill adds zero to the beginning of string until the specified length is reached.**

****

****

****

****

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

**Ans:** In python 3 default format o string is Unicode whereas in python2 we need to explicitly mention Unicode value using u.

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

**Ans**: in python 3.X **unidecode ( )** method from unidecode library can be used to put non-ASCII unicode characters in a string .

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

Ans: The major difference between these two is that 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.

When a file is opened in **text mode** , reading its data automatically decodes its content (as per the platform default or as per provided encoding), and returns it as a **str ,** writing operation takes a **str**, and automatically encodes it before transfer ring to the file. Text mode files also support universal end-of-line translation , and encoding specification arguments.

When a file is opened a **binary mode** by adding a **b** to the mode string argument in the open( ) call, reading its data does not decode it in any way, and simply returns its content raw and unchanged, as a bytes object; writing takes a bytes object and transfers it to the file unchanged. Binary mode files also accept a byte array object for the content to be written to the file.

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

**Ans:** use of encode( ) and decode ( ) method can be used to you interpret a Unicode text file containing text encoded in a different encoding than your platform’s default , by default encoding parameter is UTF-8

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

**Ans**: use str. encode ( ) and file. write ( ) to make a Unicode text file in a particular encoding format, default encoding format is UTF-18

* Call str. encode (encoding) with encoding set to utf8 to encode str.
* Call open (file, mode) to open a file with mode set to wb . wb writes to files in binary mode &preserves UTF-8 format.
* Call file. write(data) to write data to the file

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

**Ans:** Unicode represent most written language in the world. **ASCII** has its equivalent in Unicode. The difference between ASCII and Unicode is that ASCII represents lowercase letters (a-z) , uppercase letters (A-Z), digits (0-9) and symbols such as punctuation marks while Unicode represent letters of English , Arabic, Greek etc. mathematical symbols, historical scripts, emoji covering a wide range of characters than ASCII.

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

**Ans:** python 3 stores strings as Unicode by default whereas python 2 requires you to mark a string with a u if we want to store it as Unicode . Unicode strings are more versatile than ASCII strings, which are the python 3 . X default, as they can store letters from foreign languages as well as emoji and the standard Roman letters and numerals