Unicode Conversion in Python

Yash Khanna in

Abstract

Unicode is international standard where a mapping of individual characters and a unique number is maintained. As of May 2019, the most recent version of Unicode is 12.1 which contains over 137k characters including different scripts including English, Hindi, Chinese and Japanese, as well as emojis. These 137k characters are each represented by a unicode code point. So Unicode code points refer to actual characters that are displayed. These code points are encoded to bytes and decoded from bytes back to code points. You can download all the codes from **here**. The method explained here can be applied for different languages as well as for special characters as well.

Procedure

- Open and read the text file using open() and read() functions respectively.
- Using the split() function, get individual strings/words present in the text.
- For every individual word, we can use .encode() method on the string to generate a unicode.
- encode() will result in a sequence of bytes.
- There are various types of character encoding schemes, out of which the scheme UTF-8 is used in Python by default.
- A different encoding format can also be selected using encoding argument.
- For getting the original text back from unicode bytes, .decode() method can be used.
- Demo Code can be seen **here**