Assignment 1

Interpreter Vs Compiler: Difference Between Interpreter and Compiler.

Interpreter	**Compiler**
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
It takes less amount of time to analyze the source code but the overall execution time is slower.	It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Programming language like Python, Ruby use interpreters.	Programming language like C, C++ use compilers.

Important difference in python 2 and 3:

- Python 2.7 is anyway will not be supported in future and soon it will become outdated. Python 2.7 Countdown.
- print statement (2.7) and print() method (3.x).
- input() & raw_input() in 2.7 and only input() in 3.x.

- . Almost everything is generator in Python 3.x

  Most important and biggest change in Python 3.x as compared to python 2.7 is that everything has become generator. Generators in Python are having advantage of effective utilization of memory. Why waste memory with n items, when you can get one item at a time.

  In python2.7, there is range and xrange method, where xrange is a generator, and range give a list of items.

In Python 3.x, there is no xrange, range itself behaves like xrange of Python 2.7

Similarly, any call or object which was returning a list of items in python2.7 is replaced with a generator object in Python 3.x.

What is ASCII and UTF-8 ?

American Standard Code for Information Interchange (ASCII) is a character-encoding scheme and it was the first character encoding standard. It is a code for representing English characters as numbers, with each letter assigned a number from 0 to 127. Most modern character-encoding schemes are based on ASCII, though they support many additional characters.

UTF-8 - uses 1 byte to represent characters in the ASCII set, two bytes for characters in several more alphabetic blocks, and three bytes for the rest of the BMP. Supplementary characters use 4 bytes.