Find 3 differences between a compiler and an interpreter.

Interpreter	Compiler
Doesn’t require program previously to have been compiled into a machine language program.	translates instructions written in high-level programming language into the equivalent machine language
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
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.

Find the difference between Python 2 and 3

1. The print function

Python 2 doesn’t have a problem with additional parentheses, but in contrast, Python 3 would raise a Syntax Error if we called the print function the Python 2-way without the parentheses.

2. Parsing user inputs via input()

Fortunately, the input() function was fixed in Python 3 so that it always stores the user inputs as str objects. In order to avoid the dangerous behavior in Python 2 to read in other types than strings, we have to use raw_input() instead.

3. Differences mostly in texts (syntaxes)
4. Expressions
5. Unequal operations

6. Range
7. Automated migration
8. Performance issues
9. Some major housekeeping changes

What is ASCII and UTF-8?

ASCII : In ASCII, every letter, digits, and symbols that mattered (a-z, A-Z, 0–9, +, -, /, “, ! etc.) were represented as a number between 32 and 127**.**

UTF-8 (8-bit Unicode Transformation Format)is is a method for encoding Unicode characters using 8-bit sequences. Unicode is a standard for representing a great variety of characters from many languages. Other benefits of UTF-8 meant that nothing changed from the ASCII so far as the basic English character-set was considered.

Why UTF-8 is necessary?

Well, our computers use binary to store data. So inside a computer information is a sequence of 0’s and 1’s. When you are writing a text file on your computer the computer needs to store that in binary code (in 0’s and 1’s). But what would the character 'a' represent in binary? Short answer is whatever you want it to be. That is why we have UTF8 (and ASCII before it), it provides a standard that says the letter 'a' will take the value of 01100010 in binary. It allows us to say this file is stored with the UTF8 encoding, so the binary code must be interpreted with that in mind.UTF8 has a special structure in that it can represent different characters using 1, 2, 3, or 4 bytes. But how would the computer be able to tell if a UTF8 code was only one byte and the next byte was a separate character or if both bytes represented one character? Well, each byte in UTF8 is has the first couple of bits to tell the computer whether it is part of a 1, 2, 3, 4 byte sequence.