Complier

Vs

Interpreter

@_codevalley

Zuhaib Asif

Compiler:

Compiler scans the entire program and translates the whole of it into machine code at once.

Interpreter:

Interpreter tanslates just one statement of the program at a time into machine code.

Compiler:

A Compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster.

Interpreter:

An Interpretert takes very less time to analyze the source code. However, the overall time to execute the process is much slower

Compiler:

A Compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.

Interpreter:

An Interpreter does not generate an intermediary code. Hence, an interpreter is highly efficient in terms of its memory.

Compiler:

Compiler generates the error message only after it scans the complete program and hence debugging is relatively harder while working with a compiler.

Interpreter:

Keeps translating the program continuously till the first error in confronted. If any error is spotted, it stops working and hence debugging becomes easy.

Compiler:

Compilers are used by programming languages like C and C++ for example.

Interpreter:

Compilers are used by programming languages like Python and Ruby for example.