Difference between interpreter and compiler

Interpreter	Compiler
(1) Interpreter translates just one statement of the program at a time into machine code	(1) Compiler scans the entire program and translates the whole of it into machine code at once
(2) At interpreter takes very less time to analyze the source code.	(2) A compiler takes a lot of time to analyze the source code.
(3) However, the overall time to execute the process is much slower	(3) However, the overall time taken to execute the process is much faster.
(4) An interpreter does not generate an intermediary code. Hence, an interpreter is highly efficient in terms of its memory	(4) A compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.
(5) Keeps translating the program continuously till the first error is confronted. If any error is spotted, it stops working and hence debugging becomes easy	(5) A compiler generates the error message only after it scans the complete program and hence debugging is relatively harder while working with a compiler.

(6) Interpreters are used by programming languages like Ruby and python for example.	(6) compilers are used by programming language like c and c++ for example.
(7) It takes a single line of code.	(7) It takes an entire program
(8) It is best suited for the development environment	(8) It is best suited for the productive Environment
(9) Interpreted languages support Dynamic type	(9) Difficult to implement as compilers can not predict what happens at turn time.
(10) The interpreter exists in the memory during interpretation	(10) Target program executed independently and do not require the compiler in the memory.