**Difference between Compiler and Interpreter**

| S.No. | Compiler | Interpreter |
| --- | --- | --- |
| 1. | Compiler scans the whole program in one go. | Translates program one statement at a time. |
| 2. | As it scans the code in one go, the errors (if any) are shown at the end together. | Considering it scans code one line at a time, errors are shown line by line. |
| 3. | Main advantage of compilers is it’s execution time. | Due to interpreters being slow in executing the object code, it is preferred less. |
| 4. | It converts the source code into object code. | It does not convert source code into object code instead it scans it line by line |
| 5 | It does not require source code for later execution. | It requires source code for later execution. |
| Eg. | C, C++, C# etc. | Python, Ruby, Perl, SNOBOL, MATLAB, etc |

1. [**Compiler**](https://www.geeksforgeeks.org/introduction-of-compiler-design/)**:**

It is a translator which takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language.

* A compiler is more intelligent than an assembler it checks all kinds of limits, ranges, errors, etc.
* But its program run time is more and occupies a larger part of memory. It has slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.



1. [Interpreter](https://www.geeksforgeeks.org/compiler-vs-interpreter-2/):

An interpreter is a program that translates a programming language into a comprehensible language. –

* It translates only one statement of the program at a time.
* Interpreters, more often than not are smaller than compilers.

