| Compiler | Vs. | Interpreter |
| --- | --- | --- |
| • Scans the whole code & convert into machine code, doesn't require source code. | | • Translates line by line, machine code is not stored anywhere so need source code. |
| • Fast execution | | • Slow Execution |
| • CPU Utilization is high as needs to compile first. | | • CPU utilization is relatively low |
| • Difficult error analysis | | • Easy error analysis |
| • Cannot be changed during runtime hence less flexible. | | • Can be changed during runtime, more flexible. |
| • For e.g. C, C++, Java. | | • For e.g. Python, PERL, PHP, Ruby, JS. |

Note :-

→ The compiler and interpreter are more of method of implementation rather than any language's property, so let's say if your system has Java interpreter then you can run it like an interpreter.

→ Can interpreter be faster than compiler in future as the execution time gap seems to be lowering in both ?

Ans: Technically NO, as both runs the machine code which our CPU understand so it can be equal but not surpass it. Also, it may vary language to language but in general we can see the gap getting close to ~~null~~ null.