Python

Python

| compiled or interpreted? |

In many books or sources it is written that python is interpreted language. But it is half true. Python code is first compiled and then interpreted.

The complication part is done first. And a byte code is generated. But the compiled part of code gets deleted by the python as soon as we execute our code just to ensure that programmer do not get confused.

So finally python is compiled and interpreted language.

Compiler and Interpreter are two different ways to translate a program from programming or scripting language to machine language.

A compiler takes entire program and converts it into object code which is typically stored in a file. The object code is also referred as binary code and can be directly executed by the machine after linking.

An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.

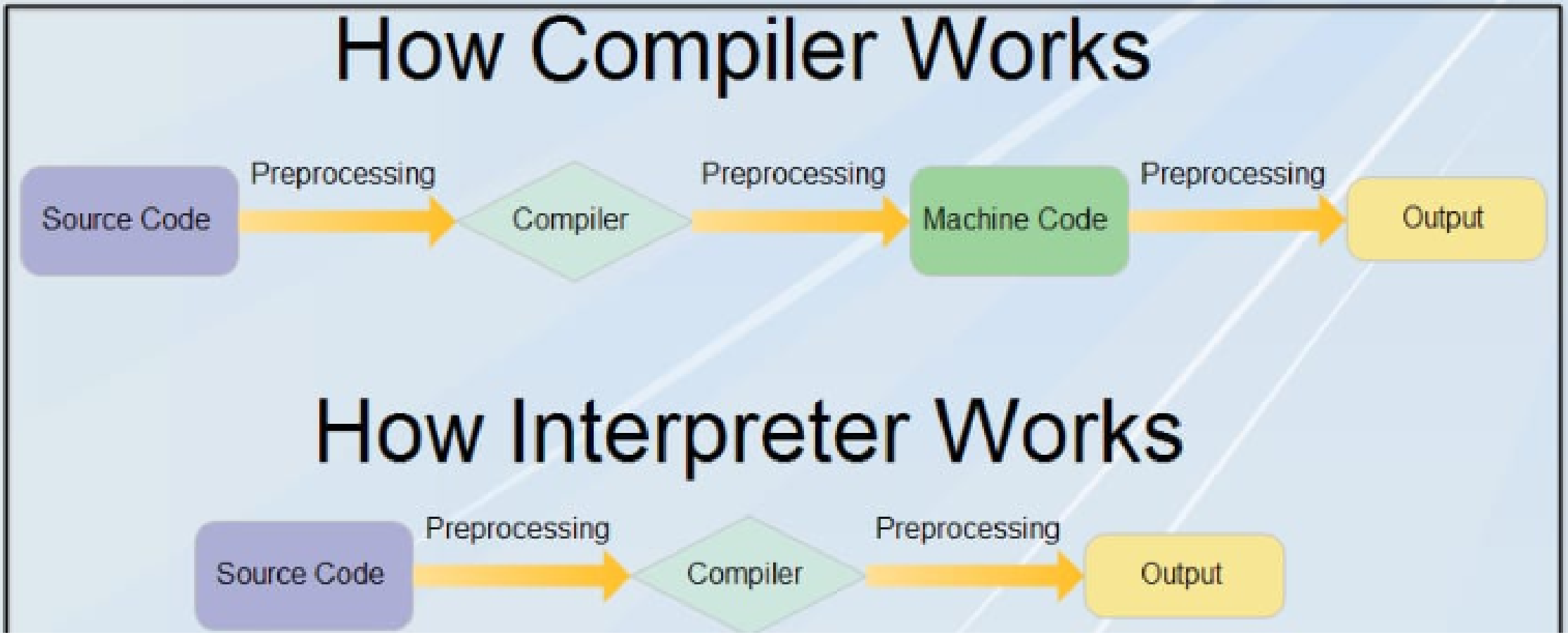
Following are some interesting facts about interpreters and compilers.

1) Both compilers and interpreters convert source code (text files) into tokens, both may generate a parse tree, and both may generate immediate instructions. The basic difference is that a compiler system, including a (built in or separate) linker, generates a stand alone machine code program, while an interpreter system instead performs the actions described by the high level program.

2) Once a program is compiled, its source code is not useful for running the code. For interpreted programs, the source code is needed to run the program every time.

3) In general, interpreted programs run slower than the compiled programs.

4) Java programs are first compiled to an intermediate form, then interpreted by the interpreter.


How Compiler Works
Source Code
Preprocessing
Compiler
Preprocessing
Machine Code
Preprocessing
Output
How Interpreter Works
Source Code
Preprocessing
Compiler
Preprocessing
Output

Compiler	Interpreter
• Compiler is a language processor which converts high level language to low level language and vise-versa.	• Interpreter is a language processor which converts high level language to low level language and vise-versa.
• It takes the entire program at a time.	• It takes a single line of code or instruction at a time.
• It generates intermediate object code.	• It does not produce any intermediate object code.
• The compilation is done before execution.	• Compilation and execution takes place simultaneously.
• Speed is Comparatively faster.	• Speed is slower.
• Memory requirement is more due to the creation of object code.	• It requires less memory as it does not create any intermediate object.
• Displays all errors after compilation, all at same time.	• Displays error of each line one by one.
• Error detection is difficult.	• Error detection is easier.
• It consumes less time.	• It consumes more time.
• Compiler is more efficient.	• Interpreter is less efficient.
Programming languages like; C, C++, C#, Scala,typescript uses a Compiler.	Programming languages like; PHP, Perl, Python, Ruby uses an Interpreter.