**PYTHON BASED COMPILER**

**Main Objective:**

* To design a python based compiler and make it work with multiple languages. But until feature update it only supports limited syntax of python.

**Requirements:**

Here we are using “rply” modules for Lexer and parser. “llvmlite” module for code generator. And we are also using subprocess, random, string and tkinter modules.

* Python 3.7 or above
* LLVM 8.0 ( <http://releases.llvm.org/download.html> )
* Clang package (Visual Studio 15)
* LLC (LLVM static compiler)

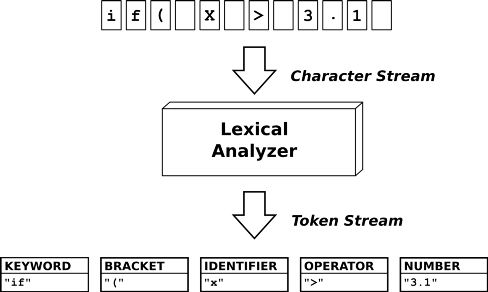
**Design:**

Compiler Design | Introduction of Compiler design**:** <https://www.geeksforgeeks.org/introduction-compiler-design/>

This compiler is divided into three sections namely: Lexer, Parser and Code Generator.

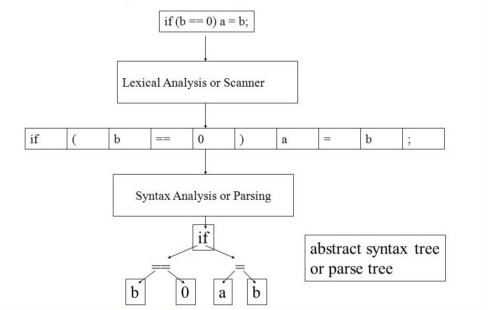
**Lexer:**

* Lexical analysis is the first phase of a compiler.
* Lexical analyzer breaks syntaxes into a series of tokens.



**Parser:**

The second component in our compiler is the **Parser**. Its role is to do a syntax check of the program. It takes the list of tokens as input and creates an [AST](https://en.wikipedia.org/wiki/Abstract_syntax_tree) as output.



**Code Generator:**

The third and last component of out compiler is the **Code Generator**. It’s role is to transform the AST created from the parser into machine language or an IR. In this case, it’s going to transform the AST into LLVM IR.

**Resources:**

* Writing your own programming language and compiler with Python: <https://blog.usejournal.com/writing-your-own-programming-language-and-compiler-with-python-a468970ae6df>
* LLVMlite Documentation
* Rply Documentation
* Clang Documentation
* LLC Documentation