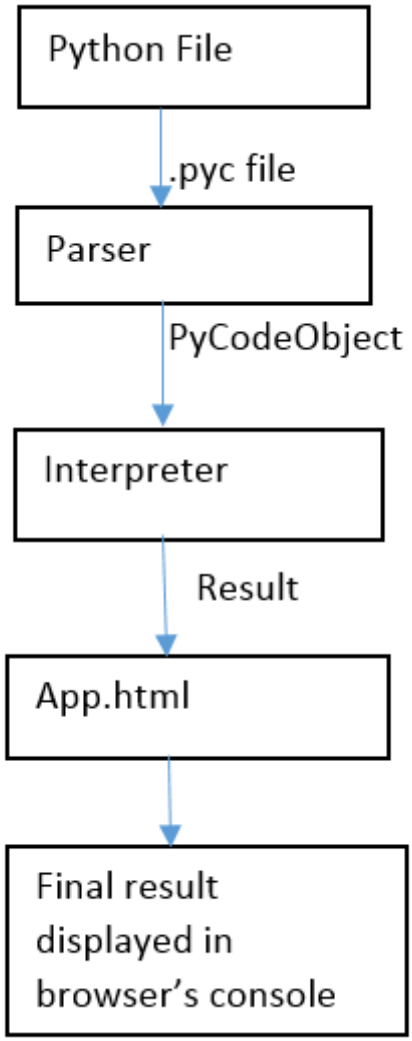
TyPy: A Python Bytecode Interpreter for the Browser

Puja Mishra, Kate Silverstein

27 October 2014

1 Understanding the Requirements

The goal of this project is to implement a Python bytecode interpreter which is engine dependent and also can be hosted on a variety of browsers. So the high level picture of how we have implemented is shown below.



2 Design Approach

The input to the system is the python program written by the user and compiled using "compileall" module. This *.pyc file is uploaded when the user opens application, this file is passed to the parser where parser parses the bytecodes and generates a PyCodeObject to the the interpreter (which here is a stack based interpreter) , which interprets it and gives the result on the browser's console.

2.1 Parser

2.1.1 Structure of a Python Bytecode File

The bytecode is an attribute of the code object and is found in cocode attribute of the code object and contains instructions for the interpreter. Bytecode is nothing but a series of bytes.The interpreter will loop through each byte, look up what it should do for each one, and then do that thing. Using the dis module we can disassemble the bytecode.

2.2 Interpreter

The interpreter gets the python code object to interpret

2.2.1 Stack-Based Interpreters

2.3 Other Design Considerations

2.3.1 Client-Side vs. Server-Side Javascript

3 Results

3.1 Supported Functionality

4 Future Work

5 Conclusion

6 References