Instat

Instagram Analysis Language

Project Manager: Jane Li

Language Guru: Enze Li

System Architect: Zhilei Miao

System Integrator: Songyan Hou

Tester: Qiurui Jin

#helloworld

```
// Hello world! for Instat

print "Hello world!";

show #helloworld;
```

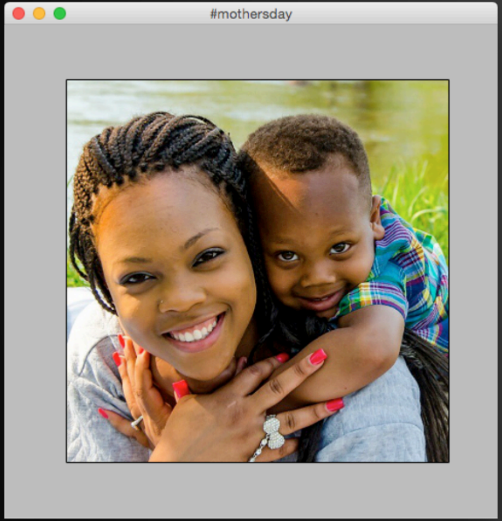
Language Design

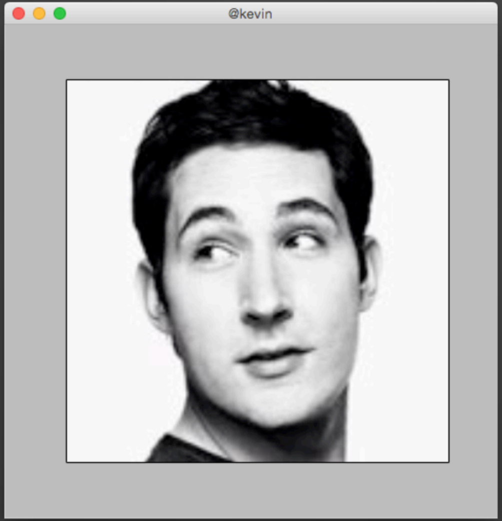
- Simple and natural
- Easy Instagram access
- Easy graphical display

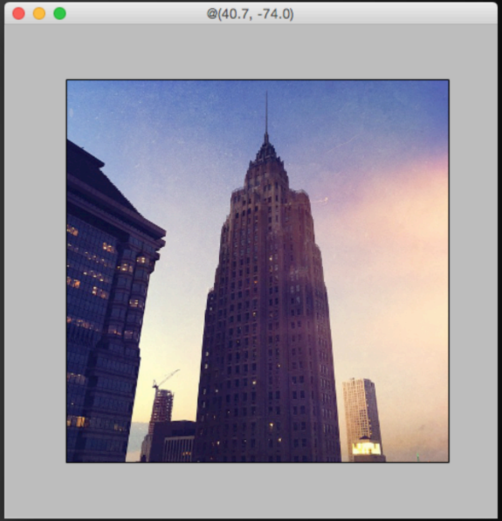
Syntactic Constructs

```
function quicksort(list, low, high) {
    i = low;
    j = high;
    pivot = list[low+(high-low)/2];
    while i <= j {
        // left as homework
    }
    if low < j {
        quicksort(list, low, j);
    }
    if i < high {
        quicksort(list, i, high);
    }
}
lst = [7, 4, 8, 5, 2, 6, 1];
print lst;
quicksort(lst, 0, length lst - 1);
print lst;
```

Graphics







```
show #mothersday;
```

```
show @kevin;
```

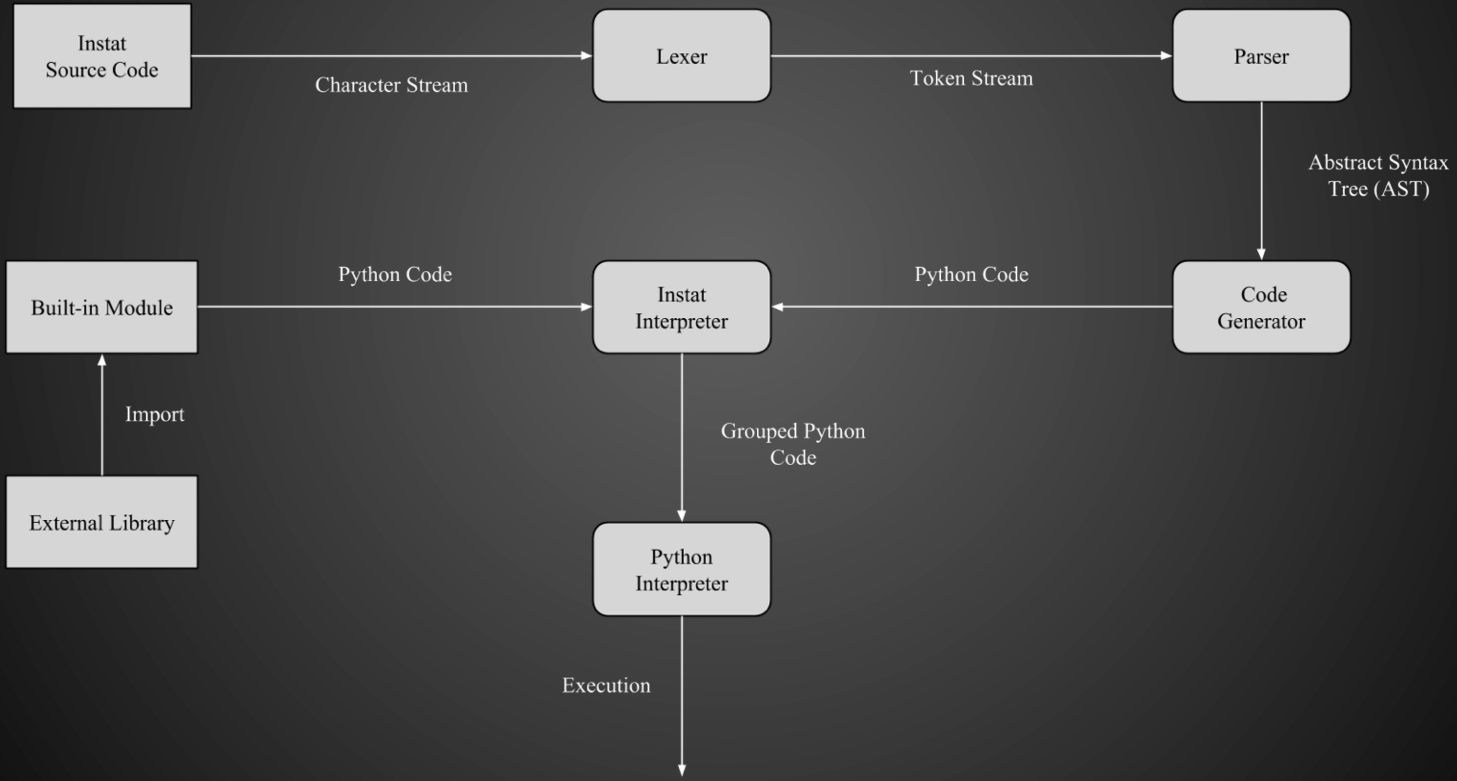
```
show @(40.7, -74.0);
```

Project Management

Weekly Meetings

Coding Days

Translator Architecture



Translator Architecture

example:

```
// Hello world! for Instat
print "Hello world!”;
show #helloworld;
```

Token Stream

```
LexToken(PRINT,'print',5,107)
LexToken(STRING,'"hello world"',5,113)
LexToken(SEMICOLON,';',5,126)
LexToken(SHOW,'show',6,140)
LexToken(HASHTAG,'#helloworld',6,145)
LexToken(SEMICOLON,';',6,156)
```

```
"program",
[
    "print",
    [
        "string",
        "\"hello world\""
    ]
],
[
    "show",
    [
        "tag",
        "#helloworld"
    ]
]
```

AST

```
print "hello world"
show(Tag('#helloworld'))
```

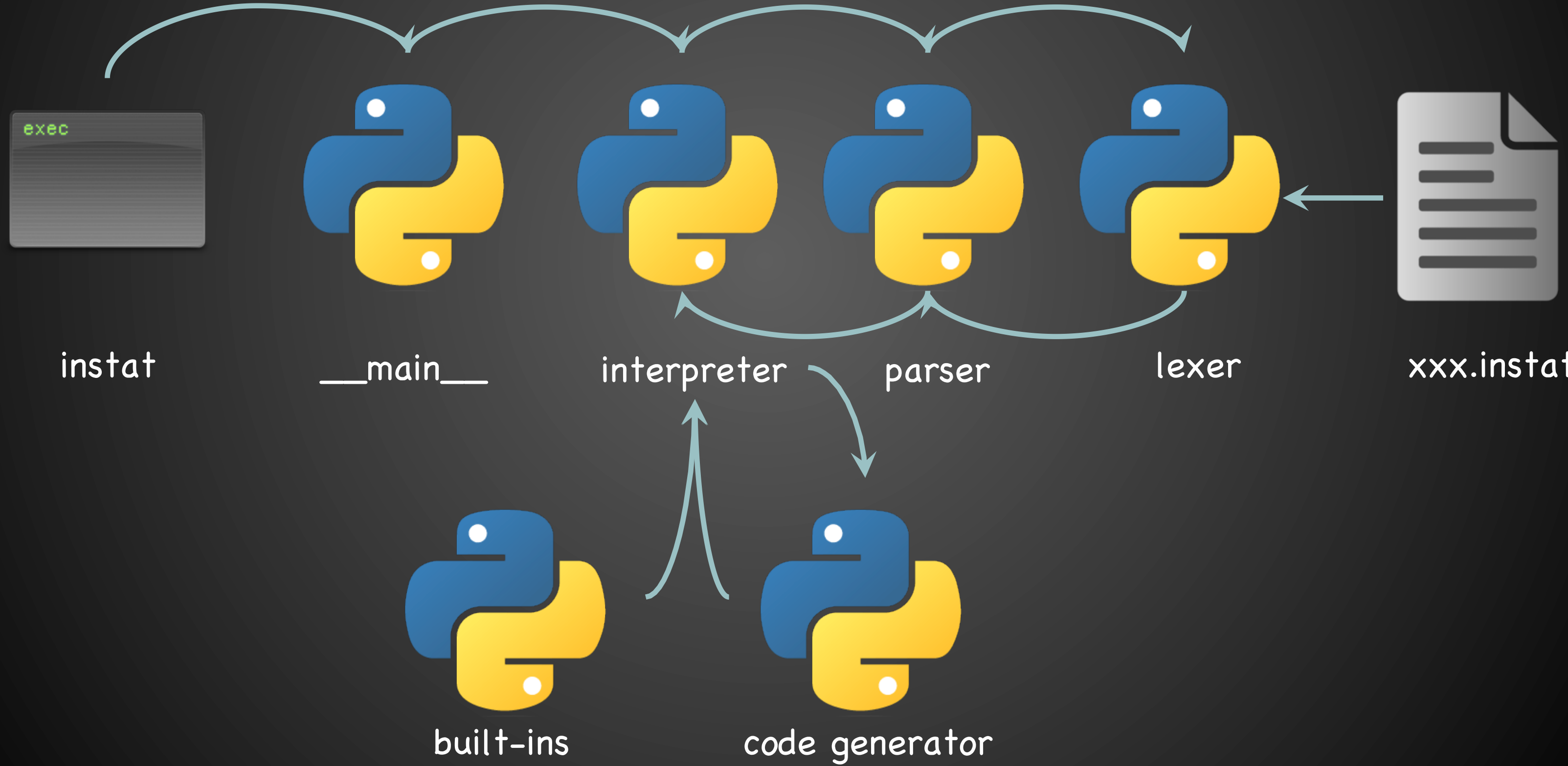
Python Code

Runtime Environment

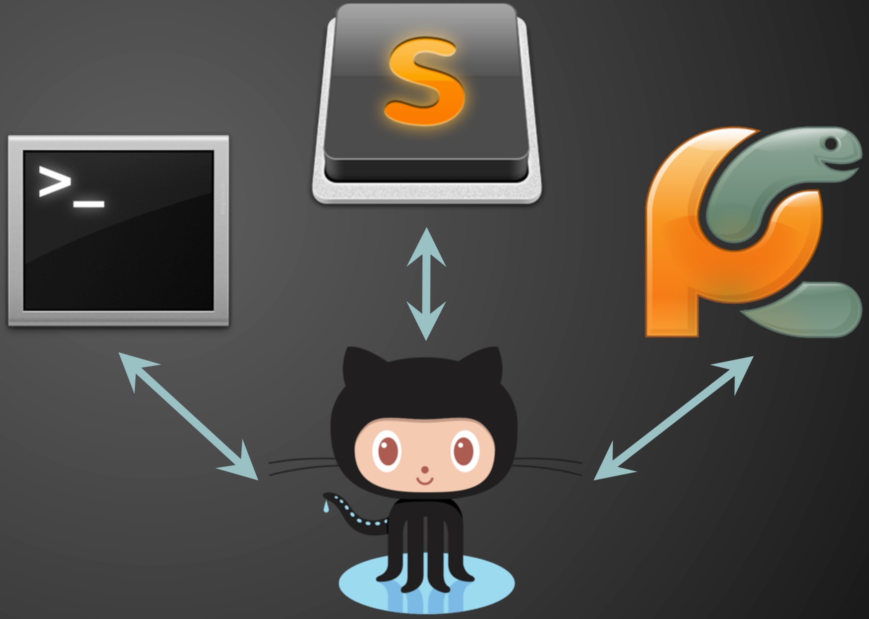
```
sudo port install py27-numpy py27-scipy
py27-matplotlib py27-ipython +notebook
py27-pandas py27-sympy py27-nose
sudo pip install Pillow
sudo pip install ply
sudo pip install python-instagram


cd interpreter
zip -r ../instat.zip *
cd ..
echo '#!/usr/bin/env python' | cat -
instat.zip > instat
chmod +x instat
```

Install_packages.sh

Runtime Environment



Development Environment



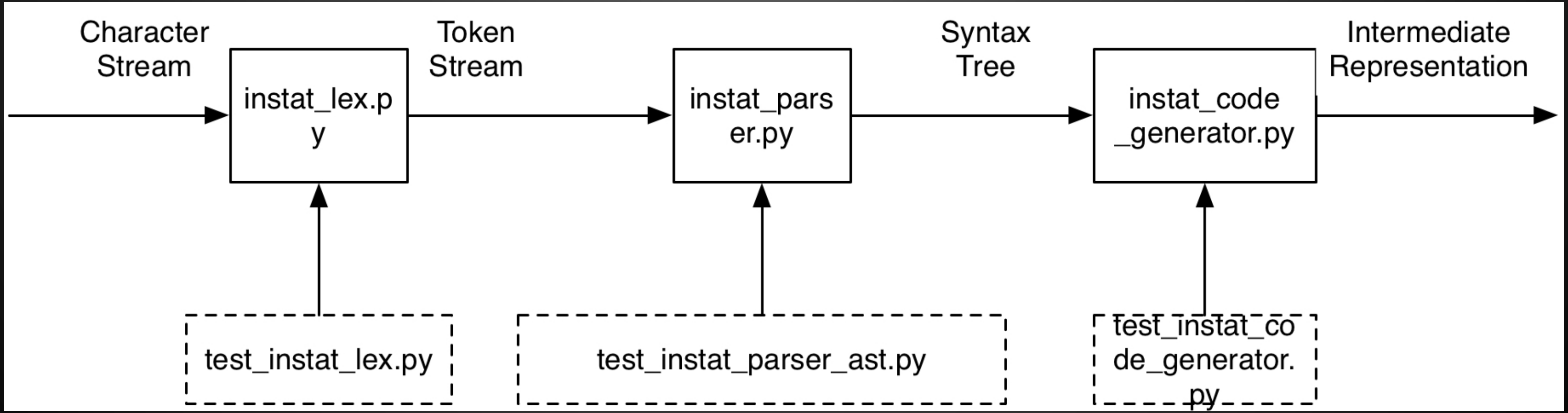
Generation Tools

- Frontend: Python-Lex-Yacc (PLY)
  - instat_lexer.py
  - instat_parser.py
- Backend: python-instagram, PyLab, Pillow
  - instat_built_in.py

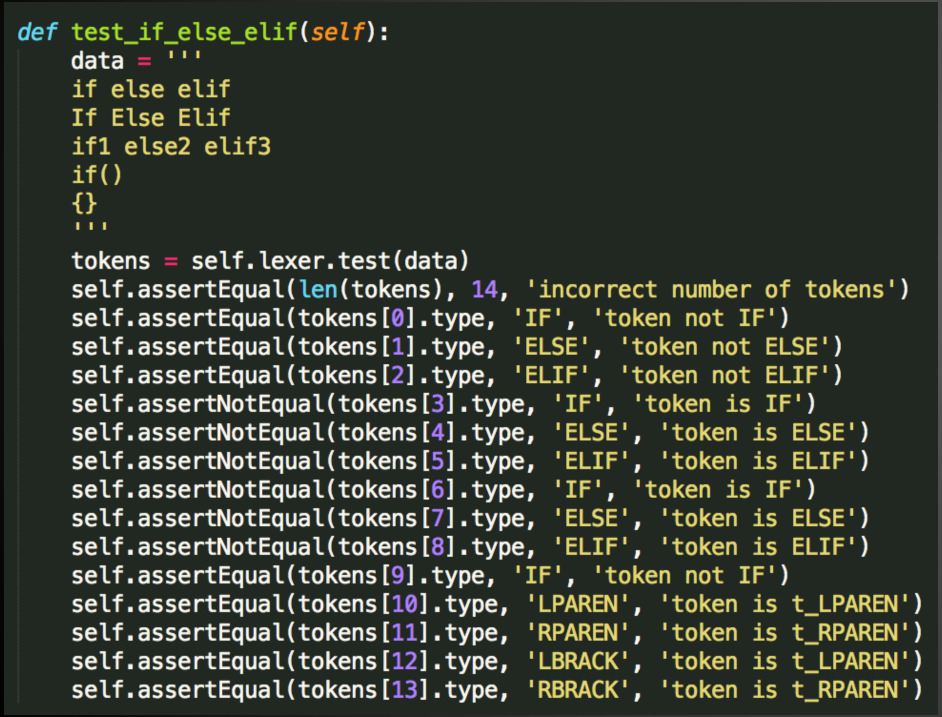
Test Plan

Python unittest

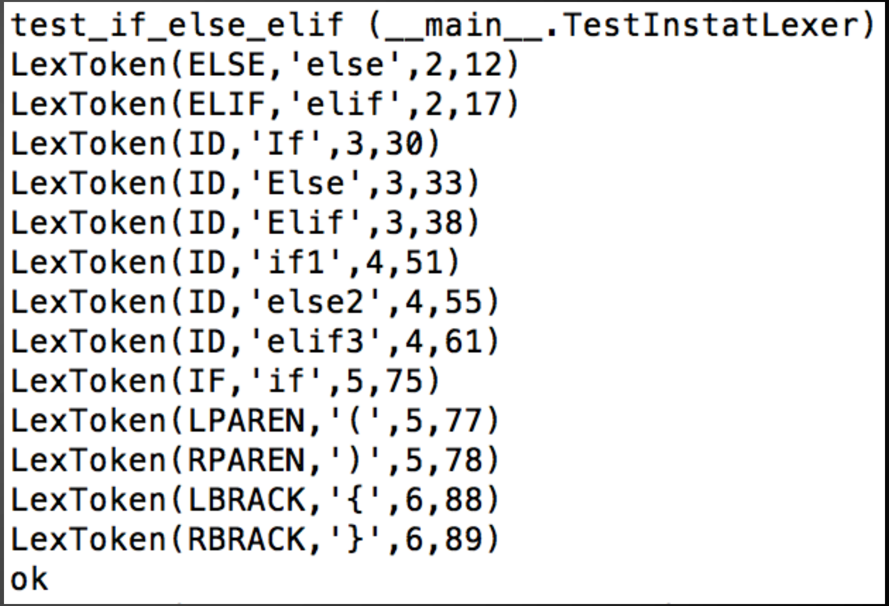
File Structure:



Sample LEX Input   Sample Output



```python
def test_if_else_elif(self):
    data = '''
    if else elif
    If Else Elif
    if1 else2 elif3
    if()
    {}
    '''
    tokens = self.lexer.test(data)
    self.assertEqual(len(tokens), 14, 'incorrect number of tokens')
    self.assertEqual(tokens[0].type, 'IF', 'token not IF')
    self.assertEqual(tokens[1].type, 'ELSE', 'token not ELSE')
    self.assertEqual(tokens[2].type, 'ELIF', 'token not ELIF')
    self.assertNotEqual(tokens[3].type, 'IF', 'token is IF')
    self.assertNotEqual(tokens[4].type, 'ELSE', 'token is ELSE')
    self.assertNotEqual(tokens[5].type, 'ELIF', 'token is ELIF')
    self.assertNotEqual(tokens[6].type, 'IF', 'token is IF')
    self.assertNotEqual(tokens[7].type, 'ELSE', 'token is ELSE')
    self.assertNotEqual(tokens[8].type, 'ELIF', 'token is ELIF')
    self.assertEqual(tokens[9].type, 'IF', 'token not IF')
    self.assertEqual(tokens[10].type, 'LPAREN', 'token is t_LPAREN')
    self.assertEqual(tokens[11].type, 'RPAREN', 'token is t_RPAREN')
    self.assertEqual(tokens[12].type, 'LBRACK', 'token is t_LPAREN')
    self.assertEqual(tokens[13].type, 'RBRACK', 'token is t_RPAREN')
```



```
test_if_else_elif (__main__.TestInstatLexer)
LexToken(ELSE,'else',2,12)
LexToken(ELIF,'elif',2,17)
LexToken(ID,'If',3,30)
LexToken(ID,'Else',3,33)
LexToken(ID,'Elif',3,38)
LexToken(ID,'if1',4,51)
LexToken(ID,'else2',4,55)
LexToken(ID,'elif3',4,61)
LexToken(IF,'if',5,75)
LexToken(LPAREN,'(',5,77)
LexToken(RPAREN,')',5,78)
LexToken(LBRACK,'{',6,88)
LexToken(RBRACK,'}',6,89)
ok
```

Sample AST Input

```python
def test_helloworld(self):
    tree = self.print_result(instat_tests.helloworld_test)
    test_tree = ast.PrintNode(ast.StringNode("\"hello world\""))
    ast_string = ast.test_display_tree(test_tree)
    self.assertEqual(tree, ast_string, 'fail to generate \"hello world\" abstract syntax tree')
```

Test case

```
helloworld_test = """
// Hello world program for Instat
/* Hello world comment style */
print "hello world";
"""
```

Conclusion

- Lessons Learned
  - Start early and read ahead
  - Build the architecture first
  - Code together

Demo Time!!!

Q&A