



# Snailz

Melissa Patton & Simon Burleson

# Lexer Yacc.py

```
tokens = ('PLUS', 'MINUS', 'TIMES', 'DIVIDE', 'LPAREN', 'RPAREN',  
         'NAME', 'NUMBER', 'AND', 'OR', 'GR8R', 'LBRA', 'RBRA', 'COM',  
         'COMPEQU', 'EQUALS', 'LES', 'MOD', 'SORT',  
         'NOT', 'EXP', 'IF', 'WHILE', 'FOR', 'ELSE', 'SNAIL')
```

# Parser Yacc.py

- statement\_assign: Assigns a value to a variable.
- statement\_expr: Evaluates an expression.
- expression\_binop: Binary operations (+, -, \*, /).
- expression\_uminus: Unary minus operation.
- expression\_group: Grouped expression.
- expression\_number: Numeric literal.
- expression\_name: Variable name.
- expression\_exp: Exponentiation operation.
- expression\_mod: Modulus operation.
- expression\_gr8r: Greater than comparison.
- expression\_les: Less than comparison.
- expression\_compequ: Equality comparison.
- expression\_lbra: Left bracket.
- expression\_rbra: Right bracket.
- expression\_list: List literal.
- list\_elements: Elements of a list.
- expression\_bogo\_sort: Bogo sort operation.
- expression\_and: Logical AND operation.
- expression\_or: Logical OR operation.
- expression\_not: Logical NOT operation.
- statement\_if: Conditional statement (IF).
- statement\_while: While loop statement.
- statement\_for: For loop statement.
- statement\_print(self, p):
- error: Error handling for syntax error

('nonassoc', 'GR8R', 'LES', 'COMPEQU'),  
('left', 'AND', 'OR'),  
('right', 'NOT'),  
('left', 'PLUS', 'MINUS'),  
('left', 'TIMES', 'DIVIDE'),  
('right', 'EXP'),  
('left', 'MOD'),  
('right', 'UMINUS'),  
('right', 'IF', 'ELSE', 'WHILE', 'FOR'),  
('nonassoc', 'LBRA', 'RBRA'),

## Precedence

# Classes

- class ASTNode: (makes AST nodes)
- class Parser: (main)
- class Snailz(Parser): (grammar definitions)

# Evaluate function (massive switch)

- Number
- WHILE
- FOR
- SNAIL
- IF
- Assignment
- Variable
- Arithmetic and Comparison Operators: (addition, subtraction, multiplication, division, and greater than, less than, equal to)
- Logical Operators: (NOT, AND, OR)
- List
- Print

# Key SLOW Features

`snail`- prints out SNAILZ 1,000 times

`>> (list)`- bogo sorts a list

`There never is a slower pace than snails competing in a race(x)`- print statement

# Basic Syntax

`x=3`

`x=[1,2,3]`

`3>x`

`3>= 5`

`3*(3+3)`

`12+5*7`

`if x==3 condition`

`if x==3 condition else condition`

`for (x<3 x=x+1) condition`

`while (x<8) condition`