

Our Language Grammar

| | | |
|---------------------|--|-----------------------------------|
| Program | ::== Block \$ | |
| Block | ::== { StatementList } | <i>Curly braces denote scope.</i> |
| StatementList | ::== Statement StatementList ::== ϵ | |
| Statement | ::== PrintStatement ::== AssignmentStatement ::== VarDecl ::== WhileStatement ::== IfStatement ::== Block | |
| PrintStatement | ::== print (Expr) | |
| AssignmentStatement | ::== Id = Expr | <i>= is assignment.</i> |
| VarDecl | ::== type Id | |
| WhileStatement | ::== while BooleanExpr Block | |
| IfStatement | ::== if BooleanExpr Block | |
| Expr | ::== IntExpr ::== StringExpr ::== BooleanExpr ::== Id | |
| IntExpr | ::== digit intop Expr ::== digit | |
| StringExpr | ::== " CharList " | |
| BooleanExpr | ::== (Expr boolop Expr) ::== boolval | |
| Id | ::== char | |
| CharList | ::== char CharList ::== space CharList ::== ϵ | |
| type | ::== int string boolean | |
| char | ::== a b c ... z | |
| space | ::== <i>the space character</i> | |
| digit | ::== 0 1 2 3 4 5 6 7 8 9 | |
| boolop | ::== == != | <i>== is test for equality.</i> |
| boolval | ::== false true | |
| intop | ::== + | |

Comments are bounded by `/*` and `*/` and ignored by the lexer.