

# Our Project Grammar

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

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