Grammar for the Itty-Bitty Teaching Language

T -> **[***S***]**

S -> **[ ]** | **[***S***]** | *SS* | *expr*

expr -> *oper* **|** *stmts*

oper -> **[:=** *oper oper***]** | **[***binops oper oper***]** |
[*unops oper***]** | *constants* | *name*

binops -> **+ | - | * | / | % | ^ | or | and**

unops -> **- | not | sin | cos | tan**

constants -> *strings* | *ints* | *floats*

strings -> **reg_ex for str literal in C** (“any alpha-numeric”) **| true | false**

name -> **reg_ex for ids in C** (any lower and upper char or underscore followed by any combination of lower, upper, digits, or underscores)

ints -> **reg ex for positive/negative ints in C**

floats -> **reg ex for positive/negative doubles in C**

stmts -> *ifstmts* | *whilestmts* | *letstmts* |*printsmts*

printstmts -> **[stdout** *oper***]**

ifstmts -> **[if** *expr expr expr***]** | **[**if *expr expr***]**

whilestmts -> **[while** *expr exprlist***]**

exprlist -> *expr* | *expr exprlist*

letstmts -> **[let [***varlist***]]**

varlist -> **[***name type***]** | **[***name type***]** *varlist*

type -> **bool | int | float | string**