Compiler: NBScript Compiler

///

Advanced Feature:

~~Coroutines~~

Features:

Function calls with arguments and return values

Local variables (globals optional)

Number, string and vector datatypes with dynamic typing, maybe reference counting

if/while/return statements

Assignment (for variables / vectors)

Most standard operators with precedence (+ - / \* == != < > && || ! unary -), bracketing (), indexing [], literal values

Syntax: (c & lua like)

Expression = name = Expression | vector = Expression | orExp

vector = name[number]

Statement = Expression | var Expression | if Expression then StatementList [else StatementList] end | while Expression do StatementList end| break | return Expression | func funcName funcBody

ExpressionList = {Expression , } Expression

StatementList = {Statement ; } Statement [;]

orExp = andexp [(”||”) orExp]

andexp = compexp [(“&&”) andexp]

compexp = eqExp [(“>”|”<”|“<=”|”>=”) compexp]

eqExp = plusexp [(“==”|”!=”) eqExp]

plusexp = mulexp [ ( “+” | “-” ) plusexp ]  
mulexp = unaryExp [ ( “\*” | “/” ) mulexp ]

unaryExp = factor [(“!”|”not”) unaryExp]  
factor = “(“ orExp “)” | number | true | false | vector | string| functionCall

functionCall = funcName (ExpressionList)

funcName = name

funcBody = StatementList end

Details:

number supports int & float

string use “ ”

vector(or container) supports number,string and vector

there is no {}

[number] only use in the vector to index the value

name should start with a letter not a number, name doesn’t support .