

## 1 8 - The Complete Syntax of Lua

Here is the complete syntax of Lua in extended BNF. (It does not describe operator precedences.)

```

chunk ::= {stat [`; ]} [laststat [`; ]]

      block ::= chunk

      stat ::= varlist `= explist |
              functioncall |
              do block end |
              while exp do block end |
              repeat block until exp |
              if exp then block {elseif exp then block} [else block] end
|
              for Name `= exp `, exp [, exp] do block end |
              for namelist in explist do block end |
              function funcname funcbody |
              local function Name funcbody |
              local namelist [`= explist]

      laststat ::= return [explist] | break

      funcname ::= Name {`.` Name} [`: Name]

      varlist ::= var {`, var}

      var ::= Name | prefixexp `[ exp `] | prefixexp `.` Name

      namelist ::= Name {`, Name}

      explist ::= {exp `, } exp

      exp ::= nil | false | true | Number | String | `... | function
|
              prefixexp | tableconstructor | exp binop exp | unop exp

      prefixexp ::= var | functioncall | `( exp `)

      functioncall ::= prefixexp args | prefixexp `:` Name args

```

```

args ::= `( [explist] `) | tableconstructor | String
function ::= function funcbody
funcbody ::= `( [parlist] `) block end
parlist ::= namelist `[, `... ] | `...
tableconstructor ::= `{ [fieldlist] `}
fieldlist ::= field {fieldsep field} [fieldsep]
field ::= `[ exp `] `= exp | Name `= exp | exp
fieldsep ::= `, | `;

binop ::= `+ | `- | `* | `/ | `^ | `% | `.. |
        `< | `<= | `> | `>= | `== | `~= |
        and | or

unop ::= `- | not | `#

```