Matlab language elements are in bold green where they may conflict with the EBNF language. TAB, SPACE and NEWLINE are constants, as you would imagine.

```
program     ::=  { statements }
statements  ::=  statement { stmtsep statements }
statement   ::=  expression | assignment | if_stmt | for_stmt | zerosfunc
if_stmt     ::=  if conditional stmtsep statements
                 { stmtsep elseif conditional stmtsep statements }
                 [ stmtsep else statements ]
                 stmtsep end
for_stmt    ::=  for assignment stmtsep statements stmtsep end
zerosfunc   ::=  zeros | zeros() | zeros(digits) | zeros(matrixidx)

assignment  ::=  variable = expression
expression  ::=  number | string | operation | conditional | range |
                 vector [ ( vectorindex ) ] | matrix [ ( matrixindex ) ]
variable    ::=  word | vector [ ( vectorindex ) ] | matrix [ ( matrixindex ) ]

operation   ::=  [ uniop ] number | [ uniop ] vector | [ uniop ] matrix |
                 number binop number | vector binop number | matrix binop number |
                 vector binop vector | matrix binop vector | matrix binop matrix

conditional ::=  condition | ( condition )
condition   ::=  [ unicond ] number | [ unicond ] vector | [ unicond ] matrix |
                 number bincond number | vector bincond number |
                 matrix bincond number | vector bincond vector |
                 matrix bincond vector | matrix bincond matrix

unicond     ::=  not
uniop       ::=  plus | minus
binop       ::=  plus | minus | times | rdiv | ldiv | etimes | erdiv | eldiv
bincond     ::=  eq | neq | gt | lt | ge | le

plus        ::=  +
minus       ::=  -
times       ::=  *
rdiv        ::=  \
ldiv        ::=  /
etimes      ::=  .*
erdiv       ::=  .\
eldiv       ::=  ./
not         ::=  ~
eq          ::=  ==
neq         ::=  ~=
gt          ::=  >
lt          ::=  <
ge          ::=  >=
le          ::=  <=

matrixindex ::=  rngsep | matrixidx { , matrixidx } | [ [ ] intrange [ ] ]
matrixidx   ::=  {digits} posdigit { rngsep {digits} posdigit }
matrix      ::=  [] | [ rowvec rowsep rowvec {rowsep rowvec} ]

vectorindex ::=  vectoridx | {digits} posdigit [ rngsep END ]
vectoridx   ::=  {digits} posdigit [ rngsep {digits} posdigit ]
vector      ::=  rowvec | colvec | chararray

chararray   ::=  ' { letter | digits | space |  punct } '
colvec      ::=  [ integer { rowsep integer} ] | [ float { rowsep float} ]
```

```
rowvec      ::=  integer {colsep integer} | float {colsep float} | range |
                 [integer {colsep integer}] | [float {colsep float}] | [range]

range       ::=  intrange | floatrange
intrange    ::=  integer [rngsep number] rngsep integer
floatrange  ::=  float [rngsep number] rngsep float

stmtsep     ::=  ; | nl | space
colsep      ::=  space | ,
rowsep      ::=  ; | nl
rngsep      ::=  :

string      ::=  " {space} [ word | punct ] { space [ word | punct ] } {space} "
word        ::=  letter { letter | unsignedint | _ }
letter      ::=  lcletter | ucletter
lcletter    ::=  a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
ucletter    ::=  A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
punct       ::=  ,|.|/|<|>|?|;|:|[|]|\|{|}||`|~|!|@|#|$|%|^|&|*|(|)|-|_|=|+|'|"

number      ::=  integer | float
integer     ::=  [sign] unsignedint
float       ::=  [sign] digits {digits} . [ digits {digits} ]
unsignedint ::=  digits {digits}
digits      ::=  zero | posdigit
posdigit    ::=  one | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

zero        ::=  0
one         ::=  1
sign        ::=  +|-

space       ::=  [ SPACE | TAB ] { [ SPACE | TAB ] }
nl          ::=  NEWLINE
```