simpL

Rules

program Top

Text notation:

```
program : command+ ;
```

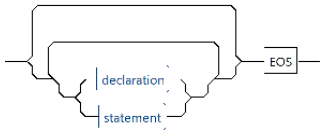
Visual notation:

command Top

Text notation:

```
command : (declaration | statement)* EOS ;
```

Visual notation:

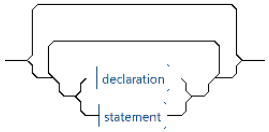


body Top

Text notation:

```
body : (declaration | statement)* ;
```

Visual notation:

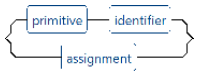


declaration Top

Text notation:

```
declaration : primitive identifier | assignment ;
```

Visual notation:

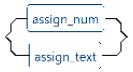


assignment Top

Text notation:

```
assignment : assign_num | assign_text ;
```

Visual notation:

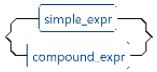


statement Top

Text notation:

```
statement : simple_expr | compound_expr ;
```

Visual notation:



simple_expr Top

Text notation:

```
simple_expr : bool_expr | arith_expr ;
```

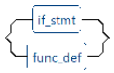
Visual notation:

compound_expr Top

Text notation:

```
compound_expr : if_stmt | func_def ;
```

Visual notation:

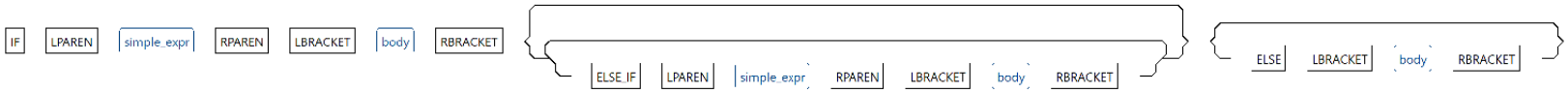


if_stmt Top

Text notation:

```
if_stmt : IF LPAREN simple_expr RPAREN LBRACKET body RBRACKET (ELSE_IF LPAREN simple_expr RPAREN LBRACKET body RBRACKET)* (ELSE LBRACKET body RBRACKET)? ;
```

Visual notation:



func_def Top

Text notation:

```
func_def : DEF ID LPAREN param_list RPAREN LBRACKET body RBRACKET ;
```

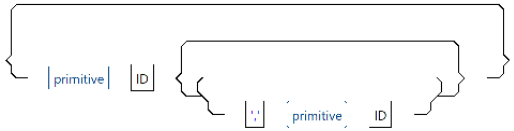
Visual notation:

param_list Top

Text notation:

```
param_list : (primitive ID (',' primitive ID)*)? ;
```

Visual notation:

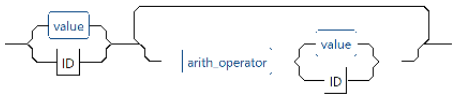


arith_expr Top

Text notation:

```
arith_expr : (value | ID) (arith_operator (value | ID))+ ;
```

Visual notation:

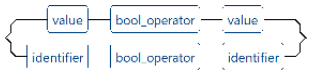


bool_expr Top

Text notation:

```
bool_expr : value bool_operator value | identifier bool_operator identifier ;
```

Visual notation:

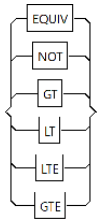


bool_operator Top

Text notation:

```
bool_operator : EQUIV | NOT | GT | LT | LTE | GTE ;
```

Visual notation:

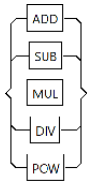


arith_operator Top

Text notation:

```
arith_operator : ADD | SUB | MUL | DIV | POW ;
```

Visual notation:



basic_type Top

Text notation:

```
basic_type : text | value ;
```

Visual notation:

identifier Top

Text notation:

```
identifier : ID ;
```

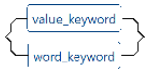
Visual notation:

primitive Top

Text notation:

```
primitive : value_keyword | word_keyword ;
```

Visual notation:



value_keyword Top

Text notation:

```
value_keyword : NUMBER ;
```

Visual notation:

word_keyword Top

Text notation:

```
word_keyword : TEXT ;
```

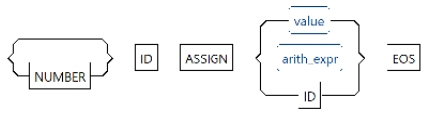
Visual notation:

assign_num Top

Text notation:

```
assign_num : NUMBER? ID ASSIGN (value | arith_expr | ID) EOS ;
```

Visual notation:



assign_text Top

Text notation:

```
assign_text : TEXT? ID ASSIGN (text | ID) EOS ;
```

Visual notation:

value Top

Text notation:

```
value : NUMERIC ;
```

Visual notation:

text Top

Text notation:

```
text : '\'' ID '\'' ;
```

Visual notation:

NUMBER Top

Text notation:

```
NUMBER : 'number' ;
```

Visual notation:

TEXT Top

Text notation:

```
TEXT : 'text' ;
```

Visual notation:

EOS Top

Text notation:

```
EOS : ';' ;
```

Visual notation:

ASSIGN Top

Text notation:

```
ASSIGN : '=' ;
```

Visual notation:

CONV Top

Text notation:

```
CONV : '\'' ;
```

Visual notation:

LPAREN Top

Text notation:

```
LPAREN : '(' ;
```

Visual notation:

RPAREN Top

Text notation:

```
RPAREN : ')' ;
```

Visual notation:

LBRACKET Top

Text notation:

```
LBRACKET : '{' ;
```

Visual notation:

RBRACKET Top

Text notation:

```
RBRACKET : '}' ;
```

Visual notation:

IF Top

Text notation:

```
IF : 'if' ;
```

Visual notation:

ELSE Top

Text notation:

```
ELSE : 'else' ;
```

Visual notation:

ELSE_IF Top

Text notation:

```
ELSE_IF : 'else if' ;
```

Visual notation:

DEF Top

Text notation:

```
DEF : 'def' ;
```

Visual notation:

RETURN Top

Text notation:

```
RETURN : 'return' ;
```

Visual notation:

EQUIV Top

Text notation:

```
EQUIV : 'is' ;
```

Visual notation:

NOT Top

Text notation:

```
NOT : 'not' ;
```

Visual notation:

GT Top

Text notation:

```
GT : '>' ;
```

Visual notation:

LT Top

Text notation:

```
LT : '<' ;
```

Visual notation:

LTE Top

Text notation:

```
LTE : '<=' ;
```

Visual notation:

GTE Top

Text notation:

```
GTE : '>=' ;
```

Visual notation:

ADD Top

Text notation:

```
ADD : '+' ;
```

Visual notation:

SUB Top

Text notation:

```
SUB : '-' ;
```

Visual notation:

MUL Top

Text notation:

```
MUL : '*' ;
```

Visual notation:

DIV Top

Text notation:

```
DIV : '/' ;
```

Visual notation:

POW Top

Text notation:

```
POW : '^' ;
```

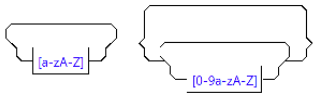
Visual notation:

ID Top

Text notation:

```
ID : [a-zA-Z]+[0-9a-zA-Z]* ;
```

Visual notation:

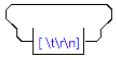


WS Top

Text notation:

```
WS : [ \t\r\n]+ -> skip ;
```

Visual notation:



NUMERIC Top

Text notation:

```
NUMERIC : ([0-9]+ | [0-9]+.[0-9]+) ;
```

Visual notation: