parse :

block EOF

block :

statement Return expression ';)'

statement :

assignment ';)'
ifStatement
forStatement
whileStatement

assignment :

Identifier '=)' expression

ifStatement :

ifStat elseIfStat elseStat Stop

ifStat :
elseIfStat :

┤├— If — expression — Start — block —┤├

┤├— Else — If — expression — Start — block —┤├

elseStat :

┤├— Else — Start — block —┤├

forStatement :

┤├— For — Identifier — '=)' — expression — To — expression — Start — block — Stop —┤├

whileStatement :

┤├— While — expression — Start — block — Stop —┤├

expression :
Def :

┤├— 'def' —┤├

If :

expression:

'-' expression

'!' expression

expression '^' expression

expression '*' expression

expression '/' expression

expression '%' expression

expression '+' expression

expression '-' expression

expression '>=' expression

expression '<=' expression

expression '>' expression

expression '<' expression

expression '==' expression

expression '!=' expression

expression '&&' expression

expression '||' expression

expression '?' expression ':' expression

expression In expression

Number

Bool

Null

Identifier

String

'(' expression ')'

Input '(' String ')'

Else :

Return :

For :

While :
To :

'while'

'to'

Start :

'start'
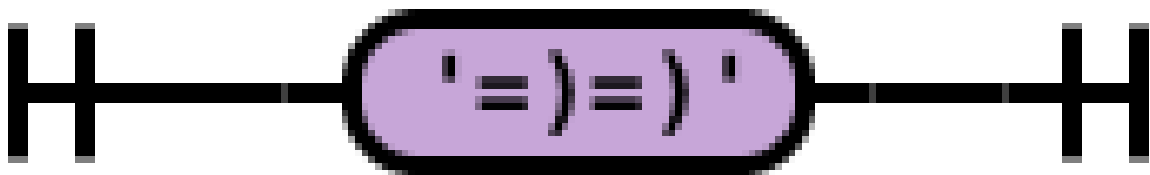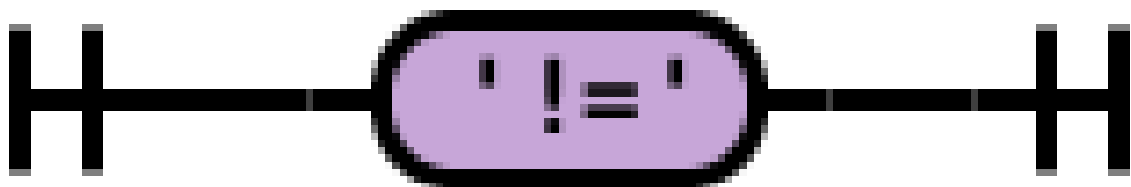
Stop :

'stop'

Input :
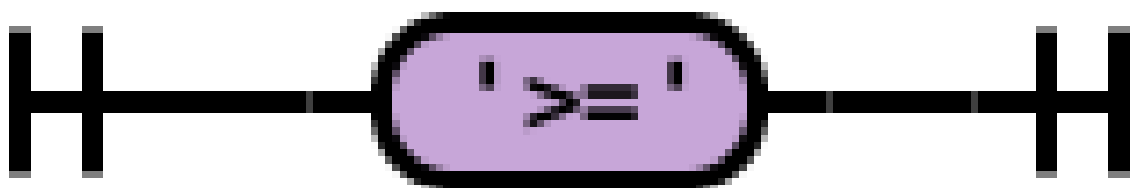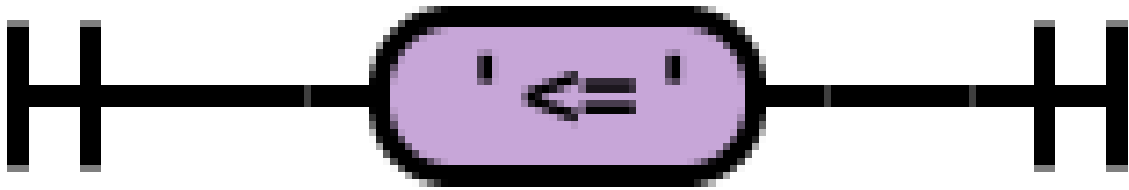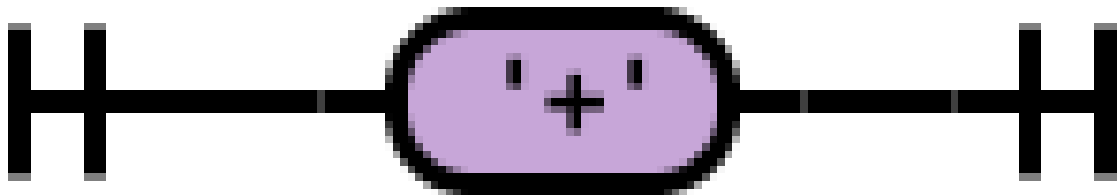In :

Null :

Pow :
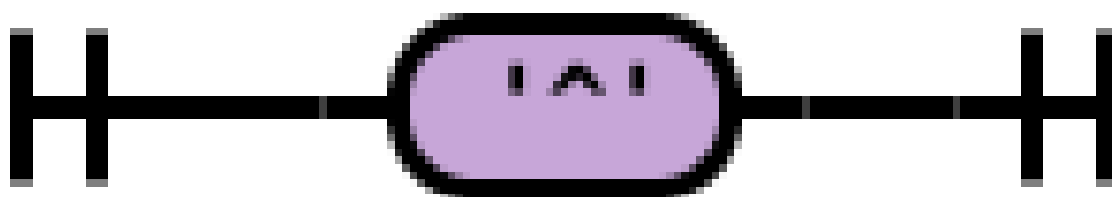
Excl :
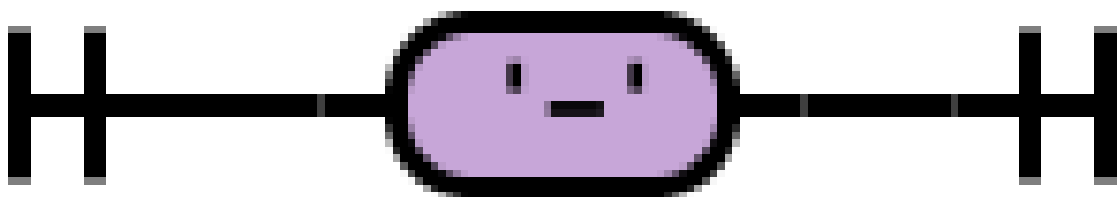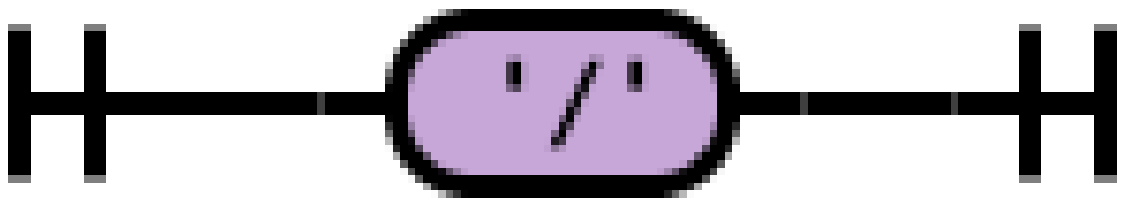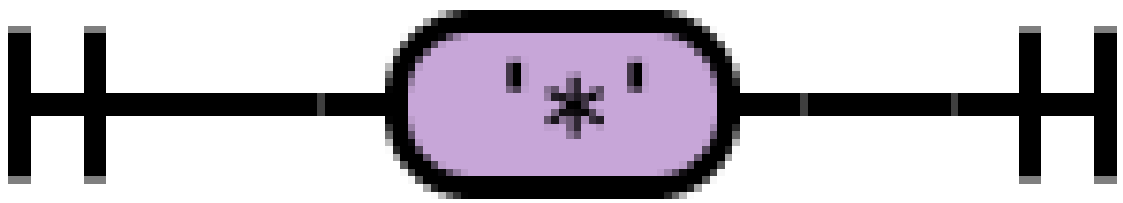GT :
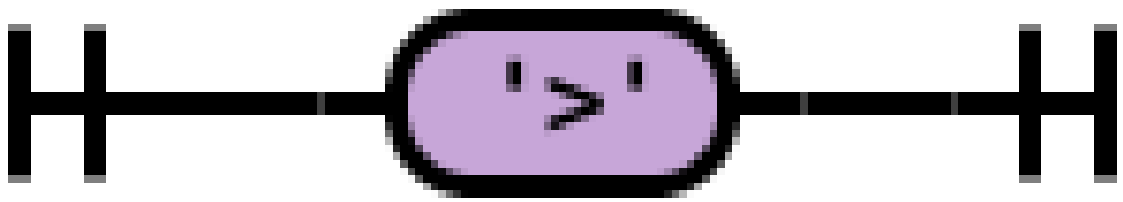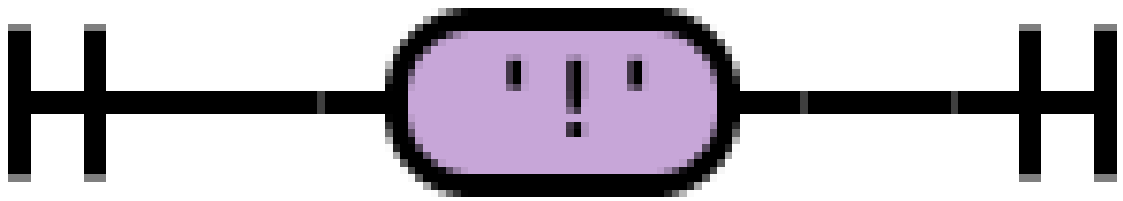
LT :
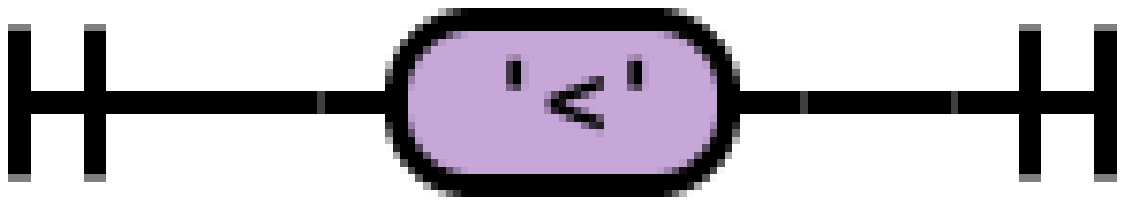
Or :
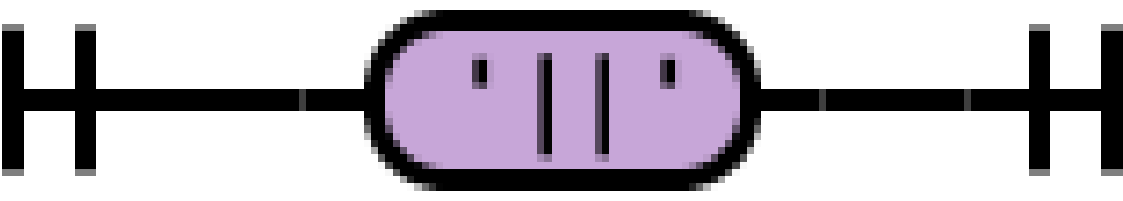
And :
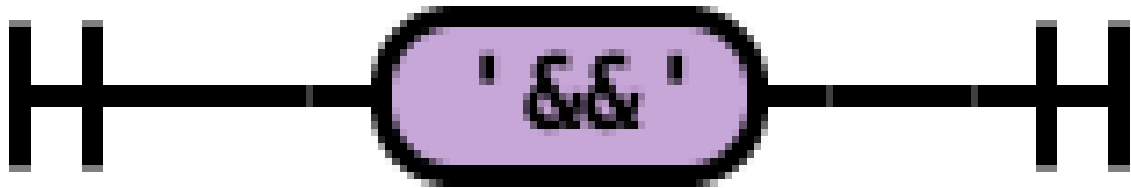
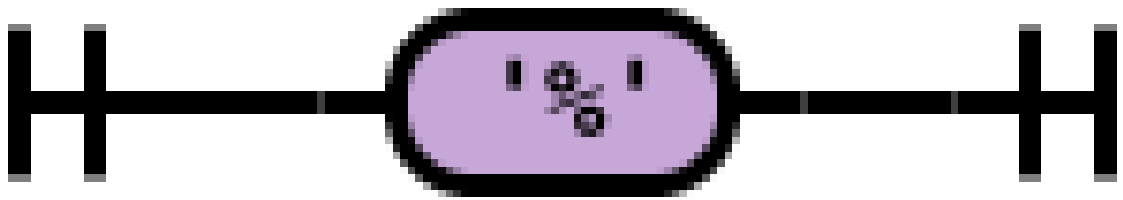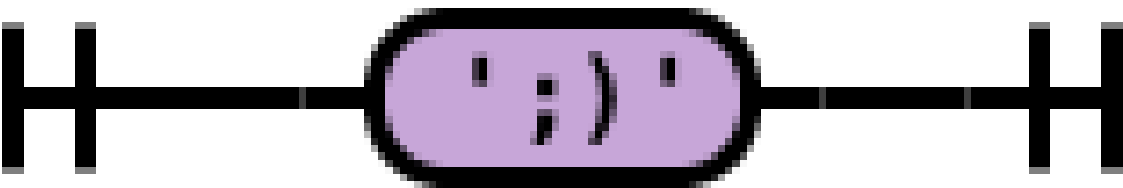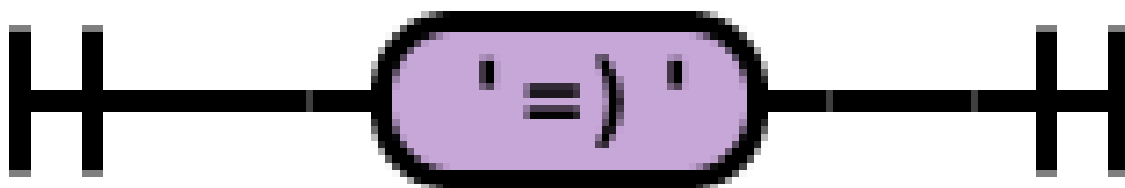Equals :

NEquals :
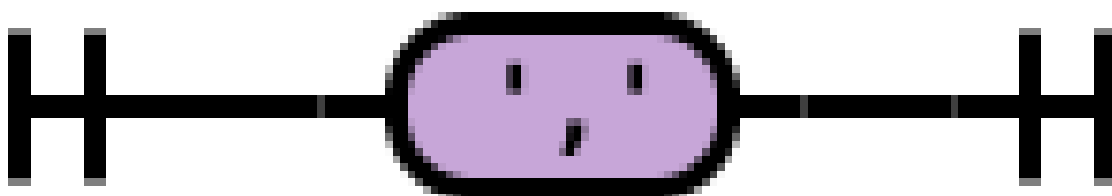

GTEquals :

LTEquals :

Add :

Subtract :

Multiply :
Divide :

Modulus :

SColon :

Assign :

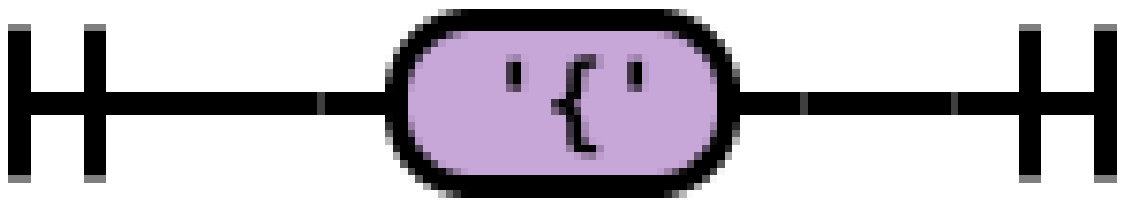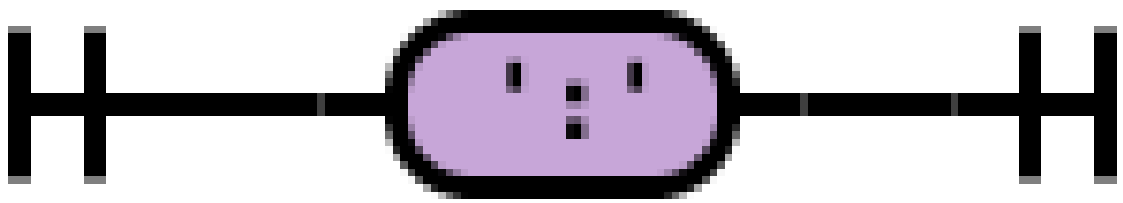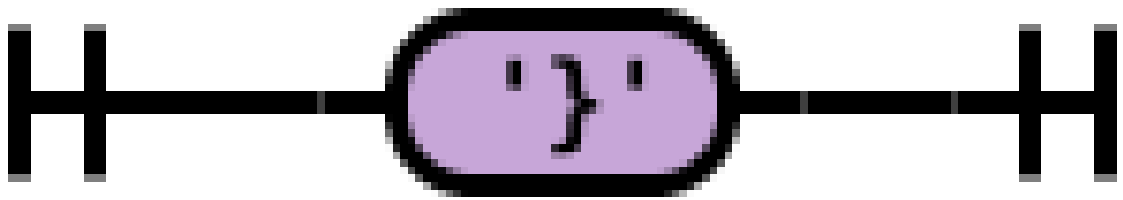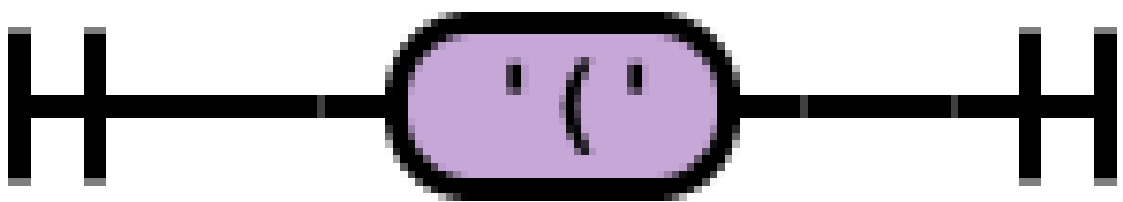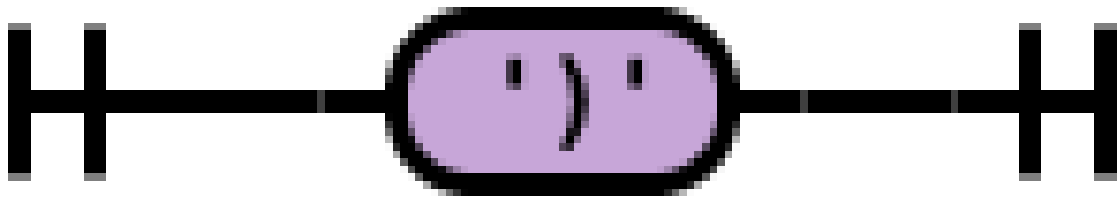
Comma :

QMark :
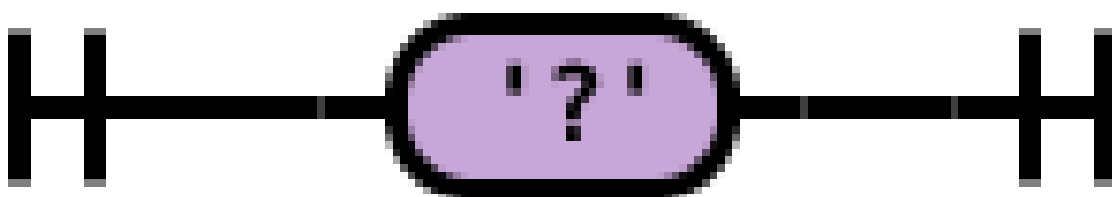
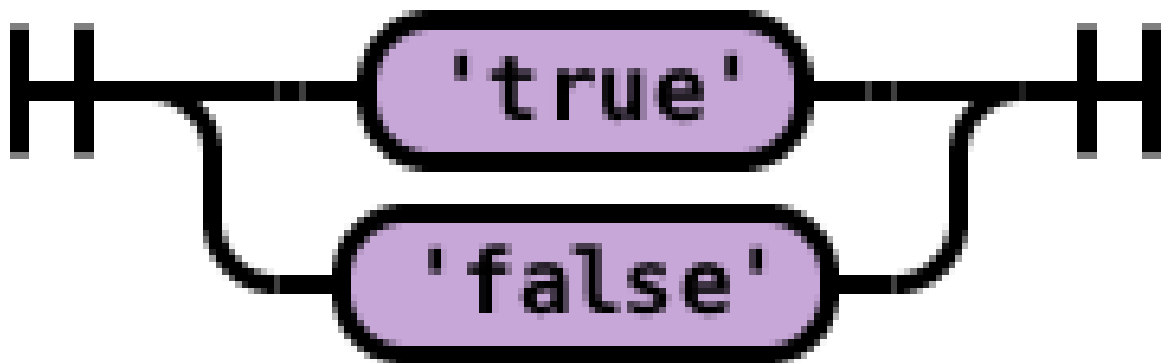
Colon :
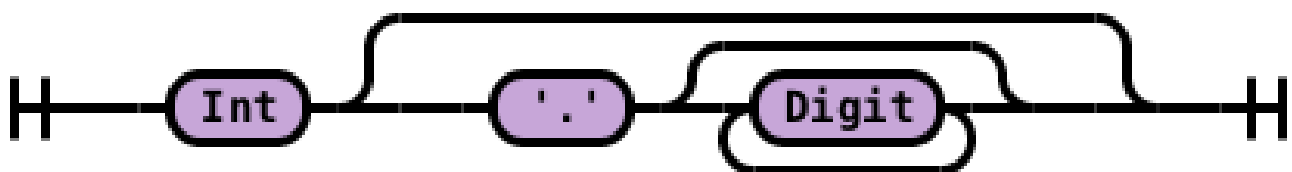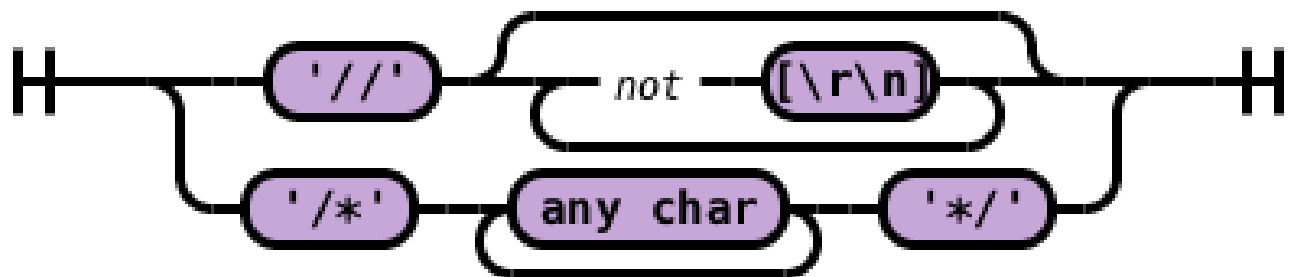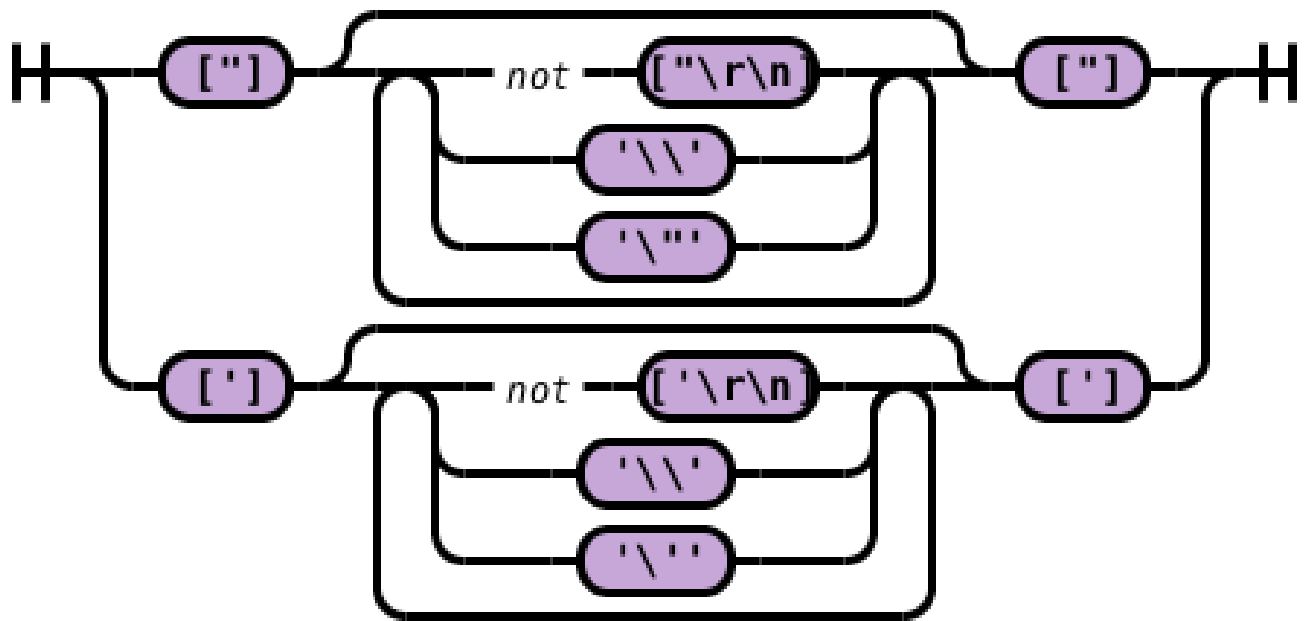OBrace :

CBrace :

OParen :

CParen :

Bool :

Number :

Identifier :

String :
Comment :

Space :

Int :
Digit :

Number:

```
├┤──┬─ [1-9] ─┬─ Digit ─┬──├┤
    │         └─────────┘  │
    └───── '0' ────────────┘
```

Digit:

```
├┤────── [0-9] ──────├┤
```