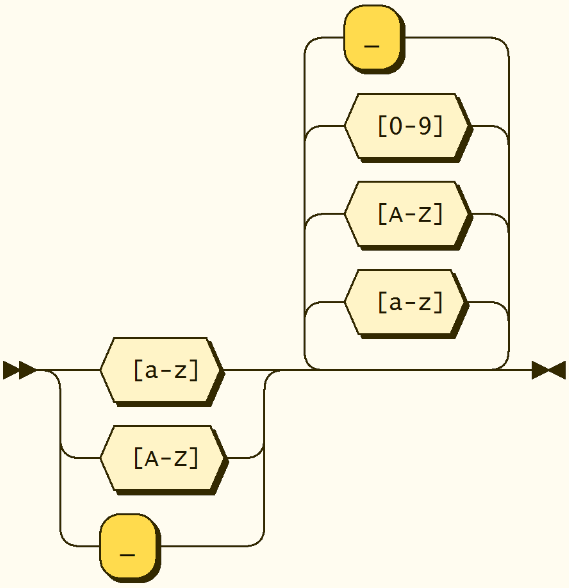
identifier:

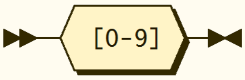


```
identifier
         ::= [a-zA-Z_] [a-zA-Z0-9_]*
```

referenced by:

- for_statetemt
- function_call
- variable

digit:

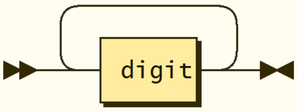


```
digit    ::= [0-9]
```

referenced by:

- constant

constant:

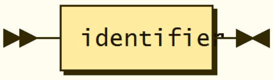


```
constant ::= digit+
```

referenced by:

- value

variable:



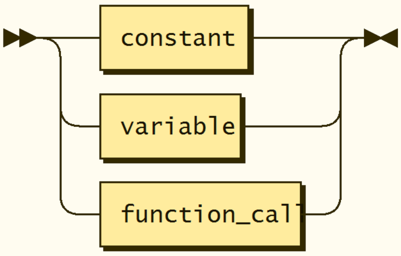
```
variable ::= identifier
```

referenced by:

- for_statetemt
- statement

- value

value:

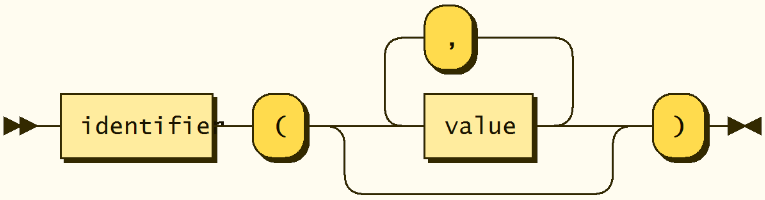


```
value    ::= constant
           | variable
           | function_call
```

referenced by:

- function_call
- logical_statement
- statement

function_call:

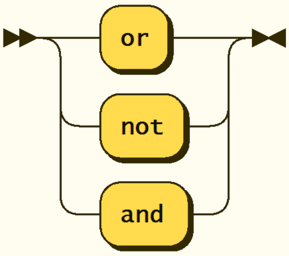


```
function_call
         ::= identifier '(' ( value ( ',' value )* )? ')'
```

referenced by:

- for_statetemt
- statement
- value

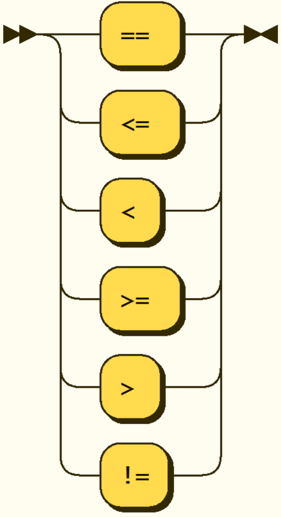
logical_operator:



```
logical_operator
         ::= 'or'
           | 'not'
           | 'and'
```

no references

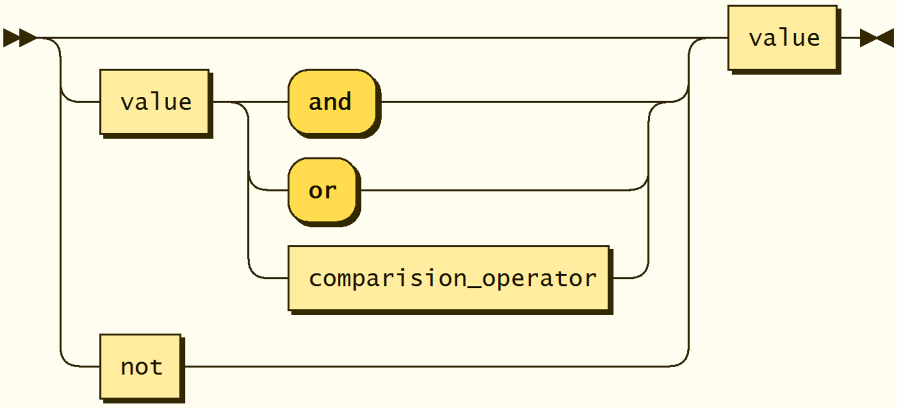
comparison_operator:



```
comparison_operator
         ::= '=='
           | '<='
           | '<'
           | '>='
           | '>'
           | '!='
```

no references

logical_statement:

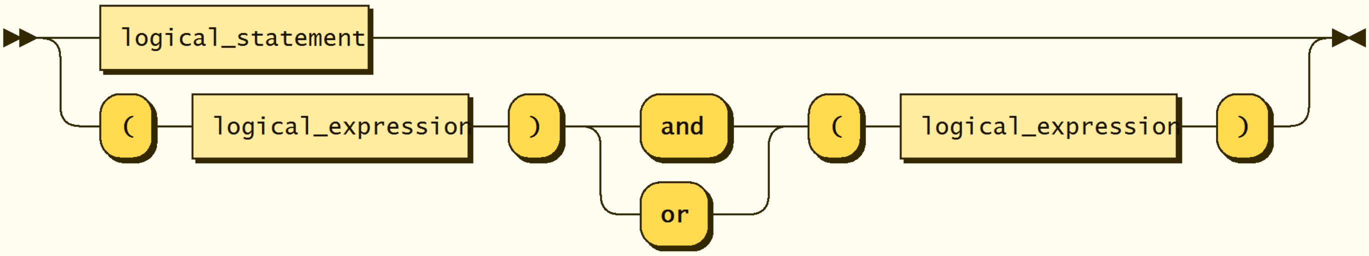


```
logical_statement
         ::= ( value ( 'and' | 'or' | comparision_operator ) | not )? value
```

referenced by:

- logical_expression

logical_expression:



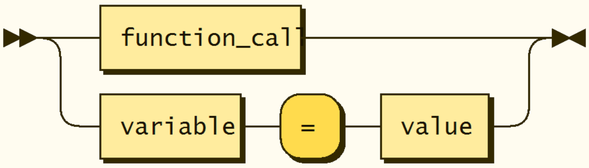
```
logical_expression
         ::= logical_statement
           | '(' logical_expression ')' ( 'and' | 'or' ) '(' logical_expression ')'
```

referenced by:

- if_statement
- logical_expression

- while_statement

statement:

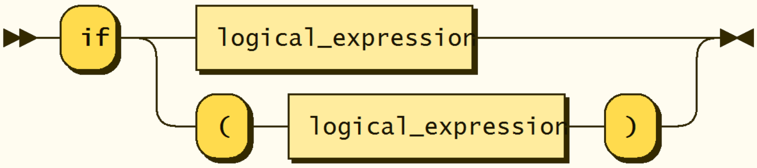


```
statement
         ::= function_call
           | variable '=' value
```

referenced by:

- block_statement

if_statement:

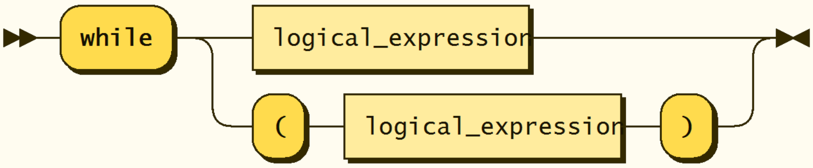


```
if_statement
         ::= 'if' ( logical_expression | '(' logical_expression ')' )
```

referenced by:

- new_block_statement

while_statement:

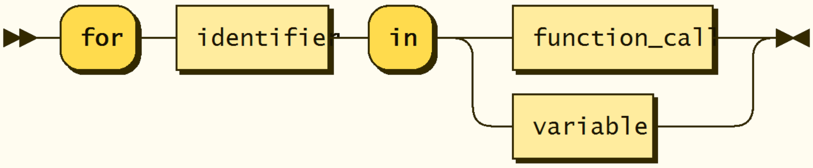


```
while_statement
         ::= 'while' ( logical_expression | '(' logical_expression ')' )
```

referenced by:

- new_block_statement

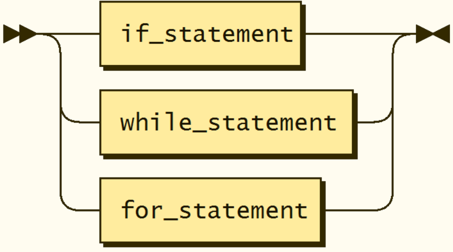
for_statetemt:



```
for_statetemt
         ::= 'for' identifier 'in' ( function_call | variable )
```

no references

new_block_statement:

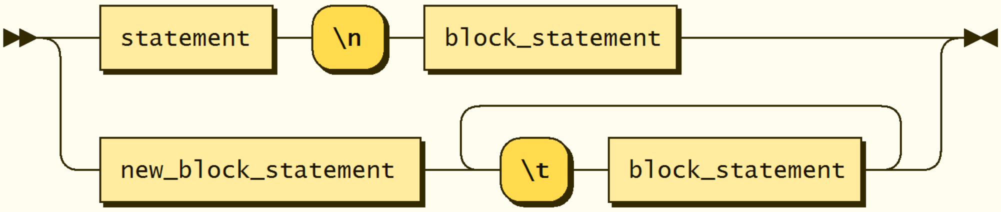


```
new_block_statement
         ::= if_statement
           | while_statement
           | for_statement
```

referenced by:

- block_statement

block_statement:



```
block_statement
         ::= statement '\n' block_statement
           | new_block_statement ( '\t' block_statement )+
```

referenced by:

- block_statement

... generated by RR - Railroad Diagram Generator