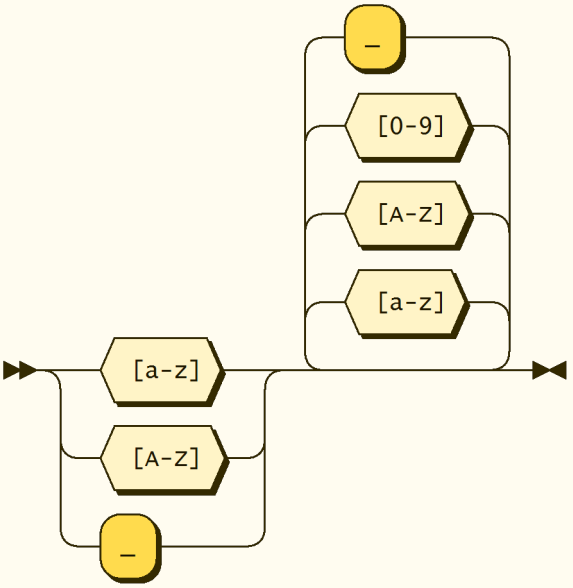
identifier:

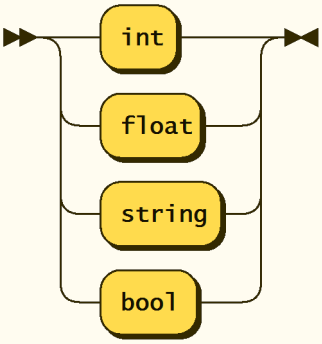



```
identifier
        ::= [a-zA-Z_] [a-zA-Z0-9_]*
```

referenced by:

- for_loop
- function_call
- function_def
- variable
- variable_def

type:

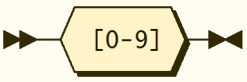



```
type     ::= 'int'
           | 'float'
           | 'string'
           | 'bool'
```

referenced by:

- function_def
- variable_def

digit:




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

referenced by:

- constant

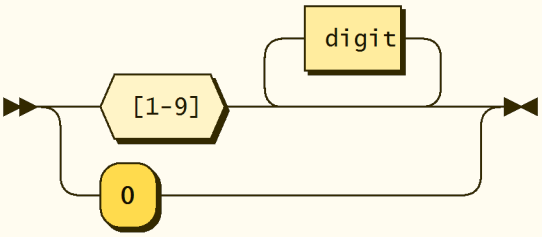
variable:


```
variable ::= identifier
```

referenced by:

- assignment
- for_loop
- function_arg
- logical_formula

constant:

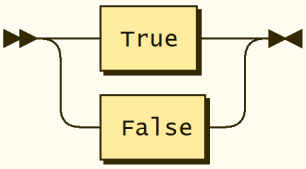



```
constant ::= [1-9] digit*
           | '0'
```

referenced by:

- for_loop
- logical_formula
- value

logical_value:

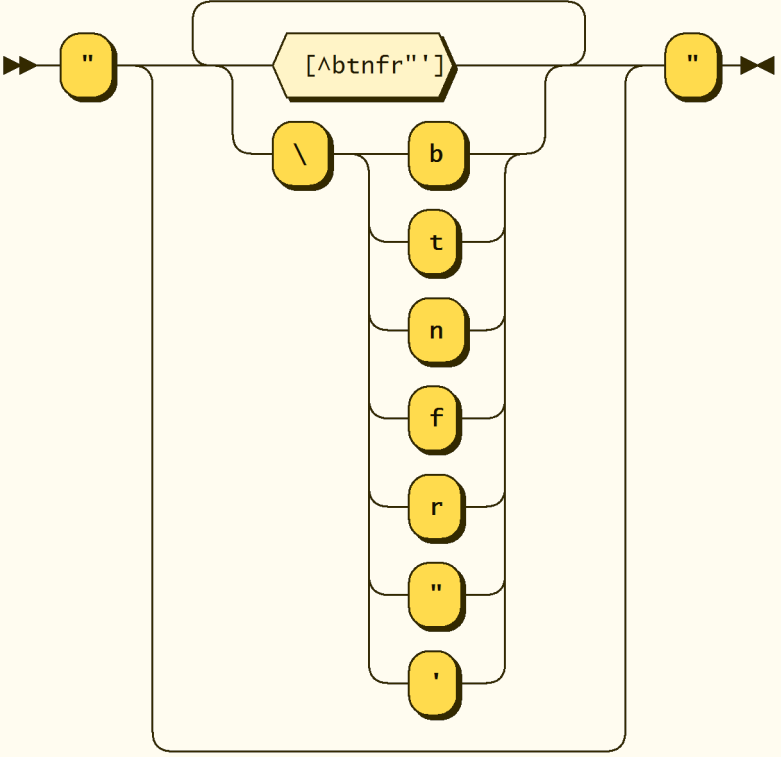



```
logical_value
          ::= True
            | False
```

referenced by:

- logical_formula
- value

string:

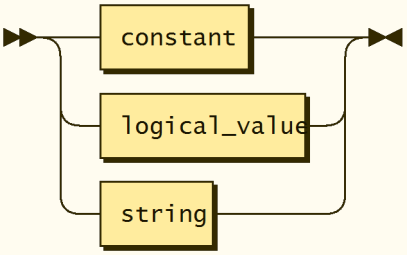



```
string   ::= '"' ( [^btnfr"'] | '\' [btnfr"'] )* '"'
```

referenced by:

- logical_formula
- value

value:

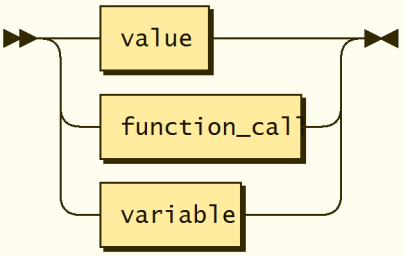



```
value    ::= constant
           | logical_value
           | string
```

referenced by:

- assignment
- function_arg

function_arg:

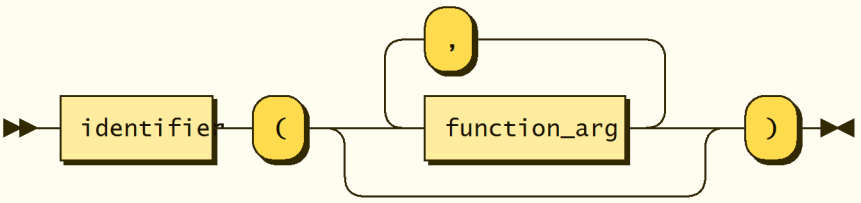



```
function_arg
         ::= value
           | function_call
           | variable
```

referenced by:

- function_call
- variable_def

function_call:

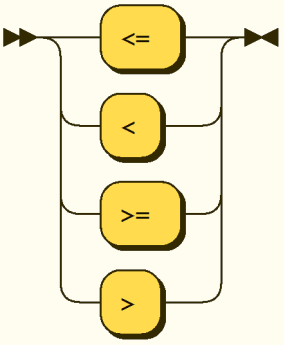



```
function_call
        ::= identifier '(' ( function_arg ( ',' function_arg )* )? ')'
```

referenced by:

- assignment
- for_loop
- function_arg
- logical_formula

comparison_operator:

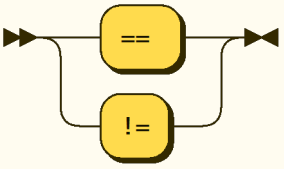



```
comparison_operator
        ::= '<='
          | '<'
          | '>='
          | '>'
```

referenced by:

- logical_formula

equality_operator:

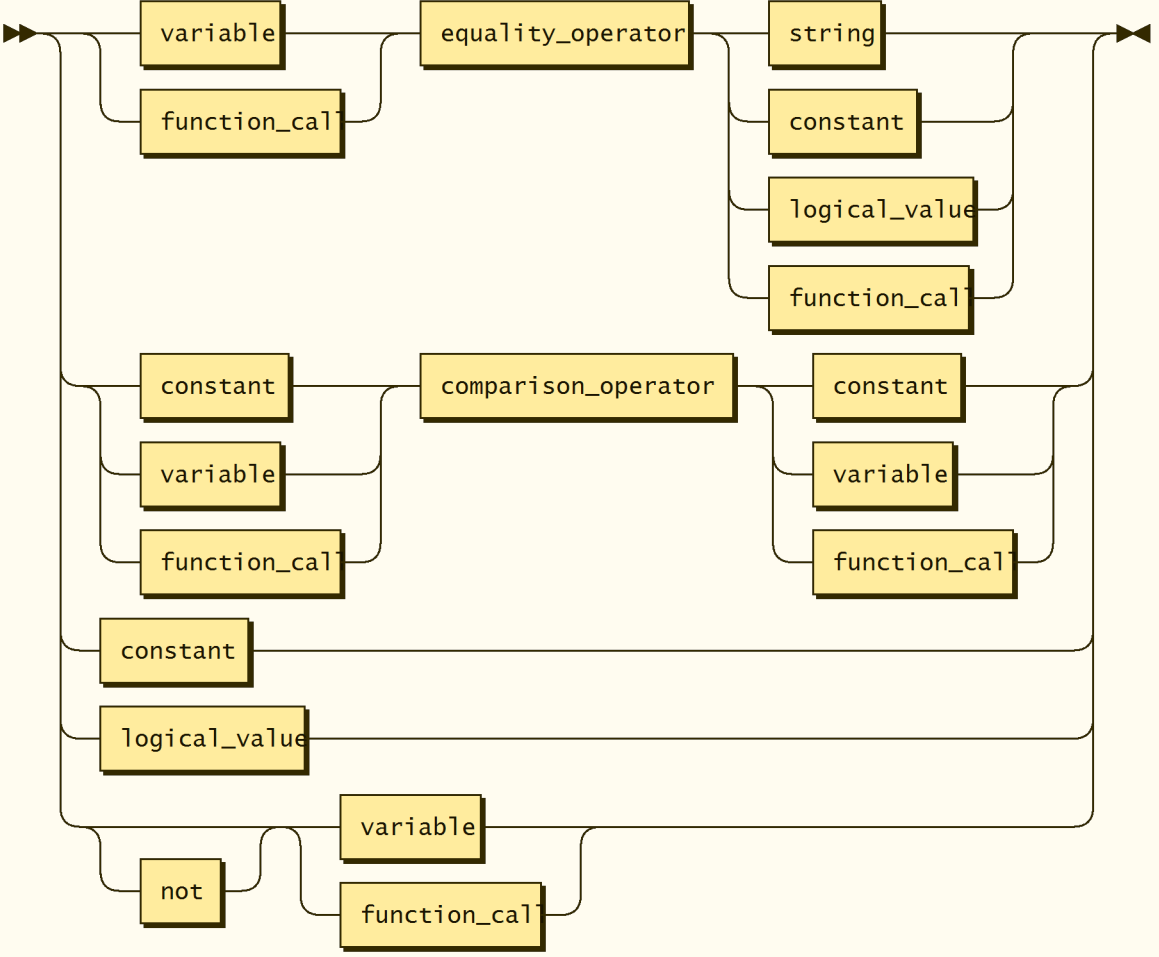



```
equality_operator
        ::= '=='
          | '!='
```

referenced by:

- logical_formula

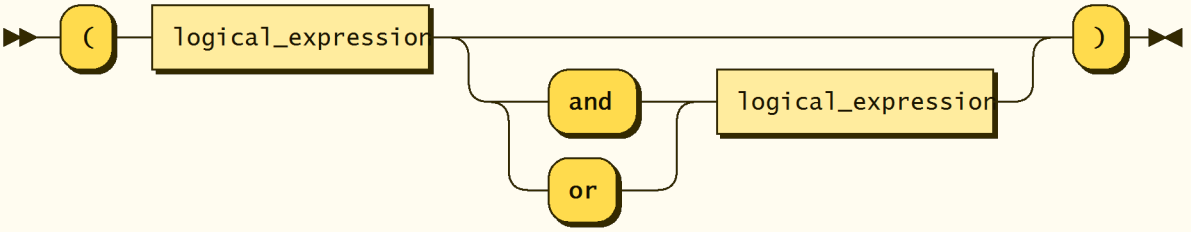
logical_formula:




```
logical_formula
        ::= ( variable | function_call ) equality_operator ( string | constant | logical_value | function_c
          | ( constant | variable | function_call ) comparison_operator ( constant | variable | function_ca
          | constant
          | logical_value
          | not? ( variable | function_call )
```

no references

logical_expression:




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

referenced by:

- assignment
- if_statement
- logical_expression
- while_loop

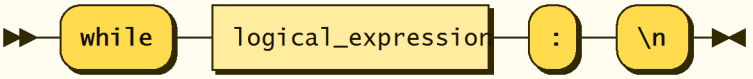
if_statement:


```
if_statement
        ::= 'if' logical_expression ':' '\n'
```

referenced by:

- code_block

while_loop:

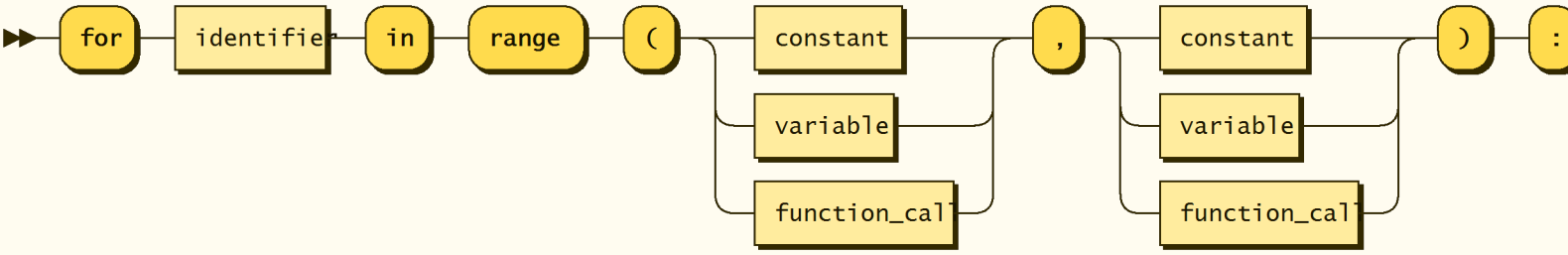



```
while_loop
        ::= 'while' logical_expression ':' '\n'
```

referenced by:

- code_block

for_loop:

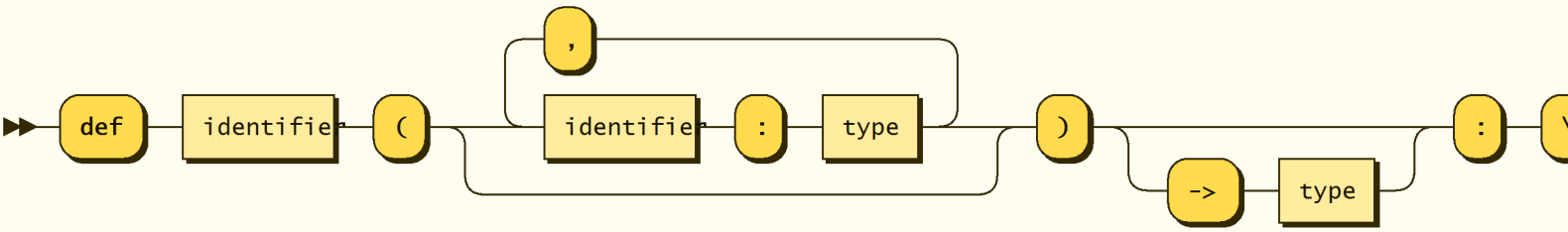



```
for_loop ::= 'for' identifier 'in' 'range' '(' ( constant | variable | function_call ) ',' ( constant | vari
':' '\n'
```

referenced by:

- code_block

function_def:

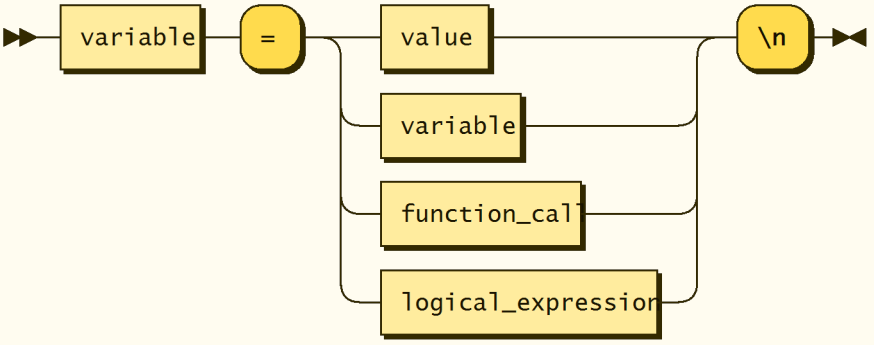



```
function_def
        ::= 'def' identifier '(' ( identifier ':' type ( ',' identifier ':' type )* )? ')' ( '->' type )? '
```

referenced by:

- code_block

assignment:

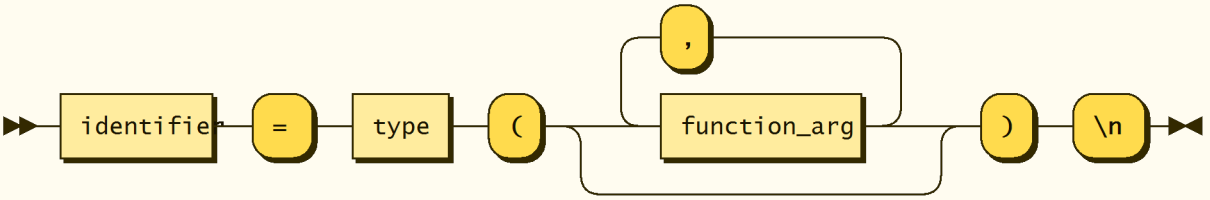


```
assignment
        ::= variable '=' ( value | variable | function_call | logical_expression ) '\n'
```

referenced by:

- code_block

variable_def:

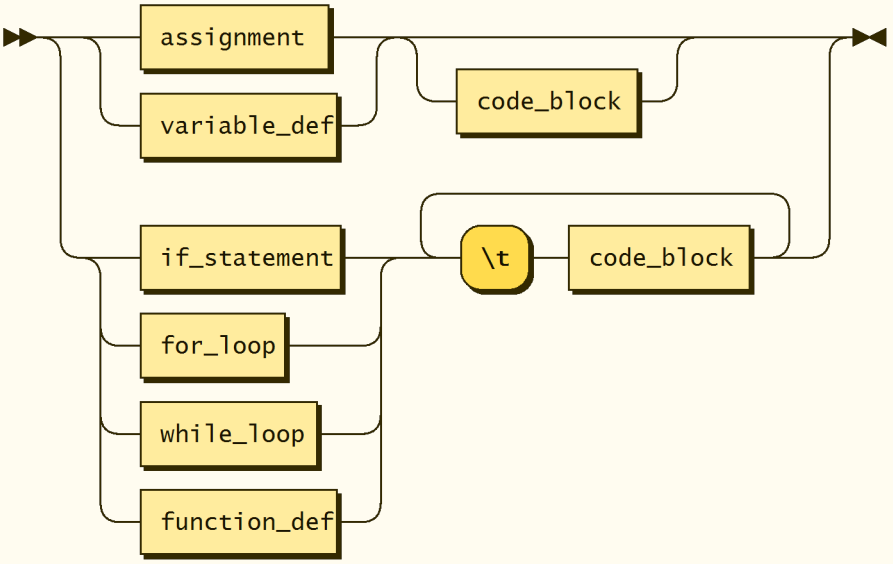



```
variable_def
        ::= identifier '=' type '(' ( function_arg ( ',' function_arg )* )? ')' '\n'
```

referenced by:

- code_block

code_block:




```
code_block
        ::= ( assignment | variable_def ) code_block?
          | ( if_statement | for_loop | while_loop | function_def ) ( '\t' code_block )+
```

referenced by:

- code_block
