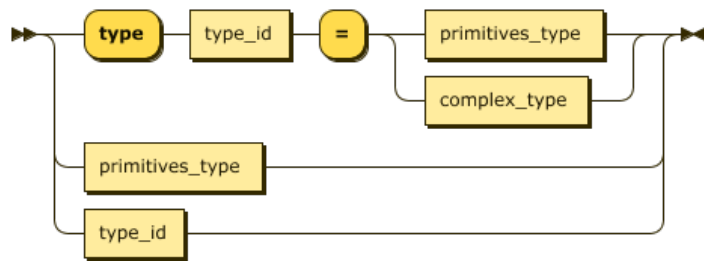


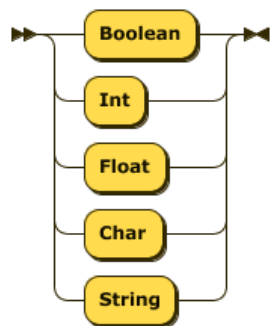
type:

```

type ::= 'type' type_id '=' ( primitives_type | complex_type )
      | primitives_type
      | type_id
  
```

referenced by:

- [FunctionLiteral](#)
- [FunctionType](#)
- [ListType](#)
- [ObjectType](#)
- [TupleType](#)
- [function_header](#)
- [variable_declaration](#)

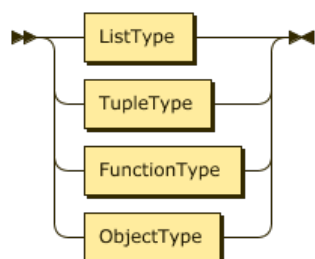
primitives_type:

```

primitives_type ::= 'Boolean'
                  | 'Int'
                  | 'Float'
                  | 'Char'
                  | 'String'
  
```

referenced by:

- [type](#)

complex_type:

```

complex_type ::= ListType
                | TupleType
                | FunctionType
  
```

| ObjectType

referenced by:

- [type](#)

ListType:

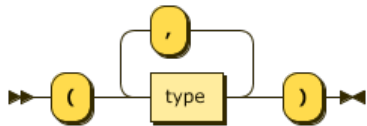


ListType ::= '[' type ']'

referenced by:

- [complex_type](#)

TupleType:



TupleType
::= '(' type (',' type)* ')'

referenced by:

- [complex_type](#)

FunctionType:

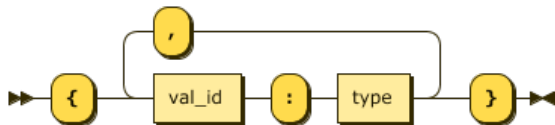


FunctionType
::= type ('→' type)+

referenced by:

- [complex_type](#)

ObjectType:

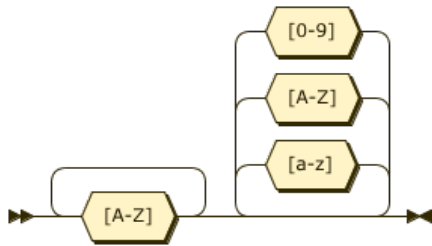


ObjectType
::= '{' val_id ':' type (',' val_id ':' type)* '}'

referenced by:

- [complex_type](#)

type_id:

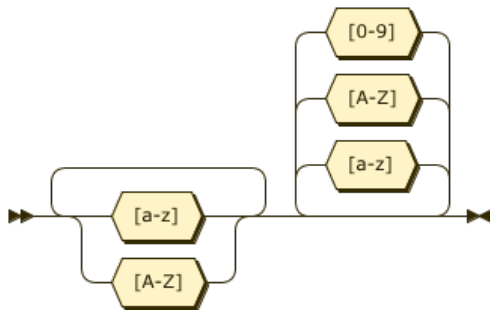


`type_id ::= [A-Z]+ [a-zA-Z0-9]*`

referenced by:

- [type](#)

val_id:

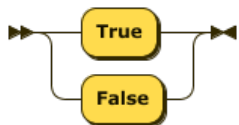


`val_id ::= [a-zA-Z]+ [a-zA-Z0-9]*`

referenced by:

- [FunctionLiteral](#)
- [Object](#)
- [ObjectType](#)
- [access_collection_expression](#)
- [access_object_property_expression](#)
- [arithmetic_expression](#)
- [boolean_expression](#)
- [char_expression](#)
- [expression](#)
- [function_body](#)
- [function_call](#)
- [function_header](#)
- [variable_declaration](#)

Boolean:

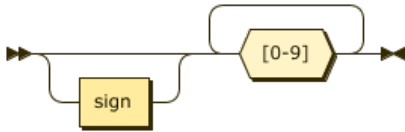


`Boolean ::= 'True'
 | 'False'`

referenced by:

- [boolean_expression](#)
- [primitive_literal](#)

Int:

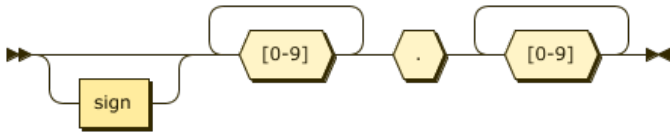


Int ::= sign? [0-9]+

referenced by:

- [numeric_literal](#)
- [primitive_literal](#)

Float:

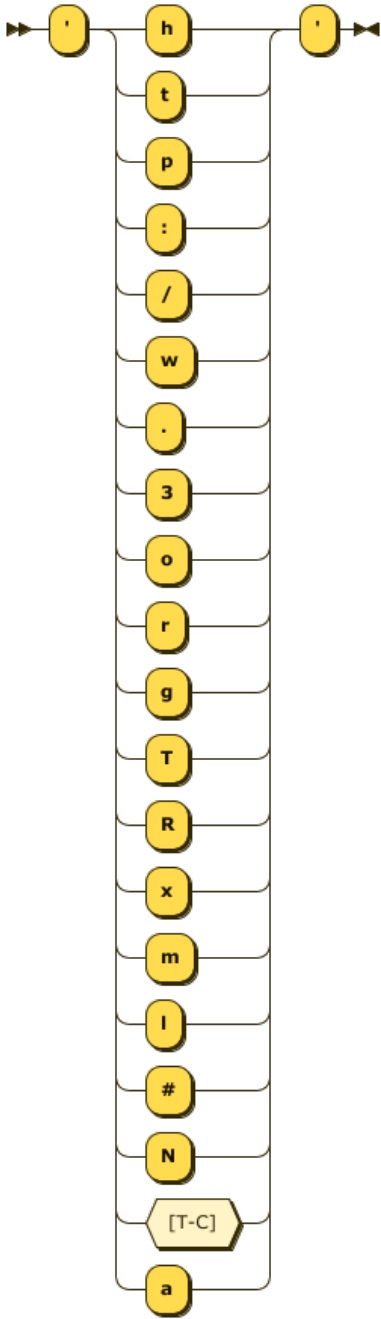


Float ::= sign? [0-9]+ . [0-9]+

referenced by:

- [numeric_literal](#)
- [primitive_literal](#)

Char:

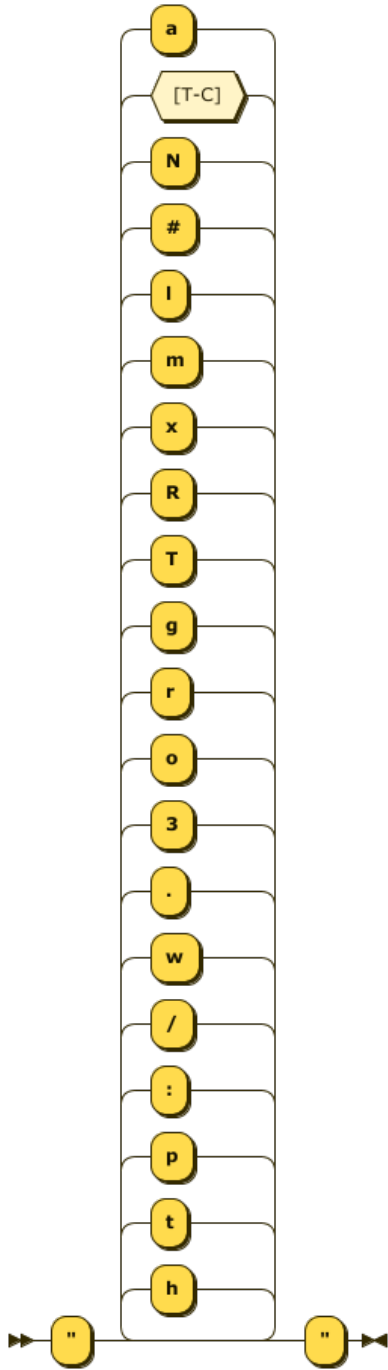


Char ::= "'" [http://w.3orgTRxml#NT-Ca] "'"

referenced by:

- [char_expression](#)
- [primitive_literal](#)

String:

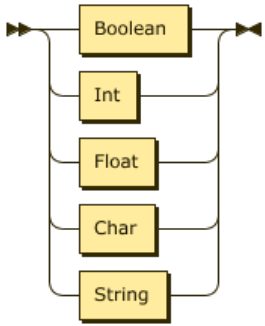


String ::= '"' [http://w.3orgTRxml#NT-Ca]* '''

referenced by:

- [char_expression](#)
- [primitive_literal](#)

primitive_literal:

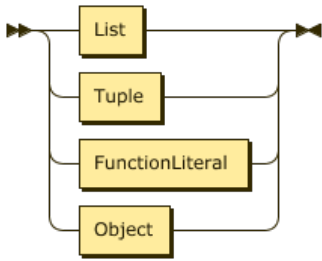


```
primitive_literal
  ::= Boolean
  | Int
  | Float
  | Char
  | String
```

referenced by:

- [literal](#)

complex_literal:

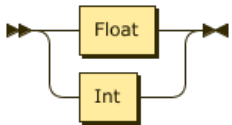


```
complex_literal
  ::= List
  | Tuple
  | FunctionLiteral
  | Object
```

referenced by:

- [literal](#)

numeric_literal:

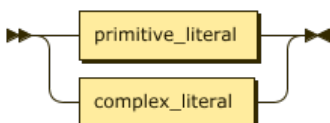


```
numeric_literal
  ::= Float
  | Int
```

referenced by:

- [arithmetic_expression](#)

literal:

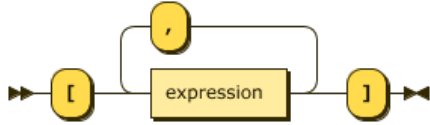


```
literal ::= primitive_literal
        | complex_literal
```

referenced by:

- [expression](#)

List:

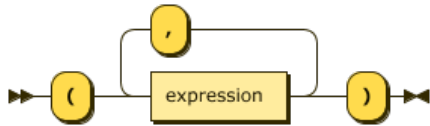


```
List ::= '[' expression ( ',' expression )* ']'
```

referenced by:

- [access_collection_expression](#)
- [complex_literal](#)

Tuple:

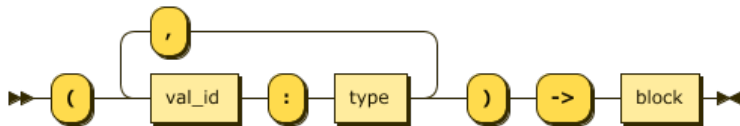


```
Tuple ::= '(' expression ( ',' expression )* ')'
```

referenced by:

- [access_collection_expression](#)
- [complex_literal](#)

FunctionLiteral:

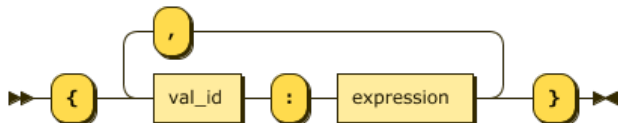


```
FunctionLiteral ::= '(' val_id ':' type ( ',' val_id ':' type )* ')' '->' block
```

referenced by:

- [complex_literal](#)

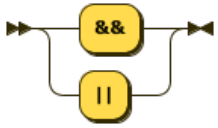
Object:



```
Object ::= '{' val_id ':' expression ( ',' val_id ':' expression )* '}'
```

referenced by:

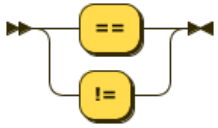
- [access_object_property_expression](#)
- [complex_literal](#)

logical_operator:

```
logical_operator
  ::= '&&'
  | '||'
```

referenced by:

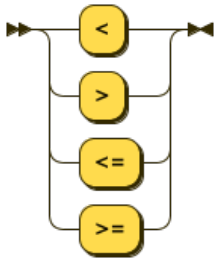
- [boolean_expression](#)

eq_or_not_equal_operator:

```
eq_or_not_equal_operator
  ::= '=='
  | '!='
```

referenced by:

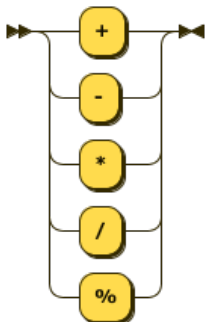
- [boolean_expression](#)

numerical_relational_operator:

```
numerical_relational_operator
  ::= '<'
  | '>'
  | '<='
  | '>='
```

referenced by:

- [boolean_expression](#)

arithmetic_operator:

```

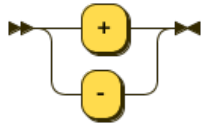
arithmetic_operator
 ::= '+'
    | '-'
    | '*'
    | '/'
    | '%'

```

referenced by:

- [arithmetic_expression](#)

sign:



```

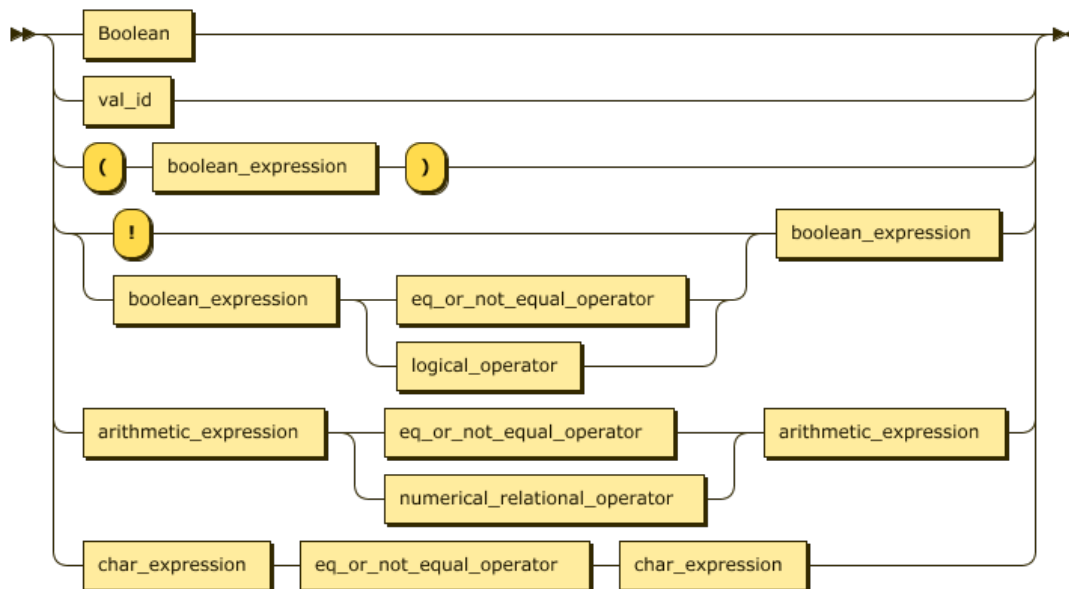
sign
 ::= '+'
    | '-'

```

referenced by:

- [Float](#)
- [Int](#)

boolean_expression:



```

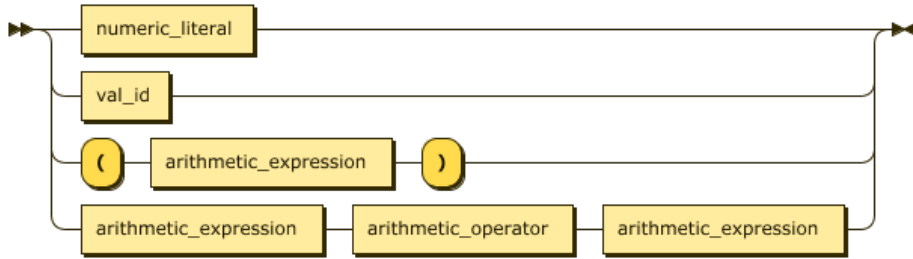
boolean_expression
 ::= Boolean
    | val_id
    | '(' boolean_expression ')'
    | '!' boolean_expression
    | boolean_expression eq_or_not_equal_operator logical_operator boolean_expression
    | arithmetic_expression eq_or_not_equal_operator numerical_relational_operator arithmetic_expression
    | char_expression eq_or_not_equal_operator char_expression

```

referenced by:

- [boolean_expression](#)
- [conditional_expression](#)
- [expression](#)

arithmetic_expression:



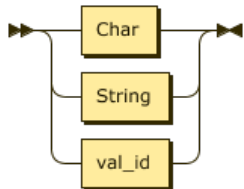
```

arithmetic_expression
  ::= numeric_literal
  | val_id
  | '(' arithmetic_expression ')'
  | arithmetic_expression arithmetic_operator arithmetic_expression
  
```

referenced by:

- [arithmetic_expression](#)
- [boolean_expression](#)
- [expression](#)

char_expression:



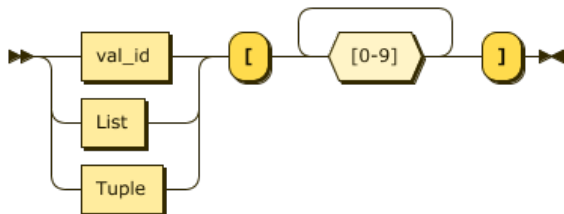
```

char_expression
  ::= Char
  | String
  | val_id
  
```

referenced by:

- [boolean_expression](#)

access_collection_expression:



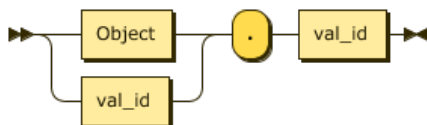
```

access_collection_expression
  ::= ( val_id | List | Tuple ) '[' [0-9]+ ']'
  
```

referenced by:

- [expression](#)

access_object_property_expression:



```

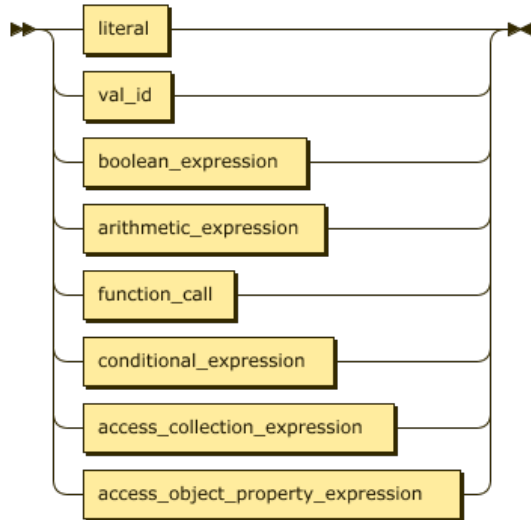
access_object_property_expression
  
```

```
::= ( Object | val_id ) '.' val_id
```

referenced by:

- [expression](#)

expression:



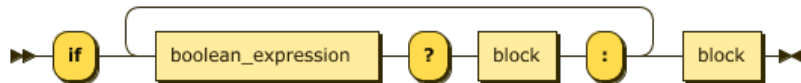
```

expression
  ::= literal
  | val_id
  | boolean_expression
  | arithmetic_expression
  | function_call
  | conditional_expression
  | access_collection_expression
  | access_object_property_expression
  
```

referenced by:

- [List](#)
- [Object](#)
- [Tuple](#)
- [block](#)
- [function_call](#)
- [io](#)
- [main](#)
- [variable_declaration](#)

conditional_expression:



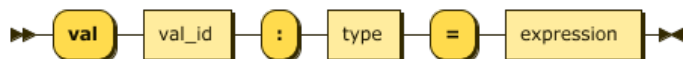
```

conditional_expression
  ::= 'if' ( boolean_expression '?' block ':' )+ block
  
```

referenced by:

- [expression](#)

variable_declaration:



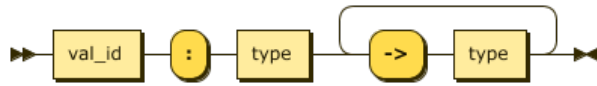
```

variable_declaration
  ::= 'val' val_id ':' type '=' expression
  
```

referenced by:

- [block](#)
- [main](#)

function_header:

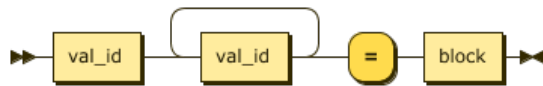


```
function_header
  ::= val_id ':' type ( '->' type )+
```

referenced by:

- [main](#)

function_body:

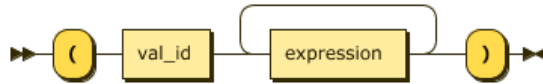


```
function_body
  ::= val_id val_id+ '=' block
```

referenced by:

- [main](#)

function_call:

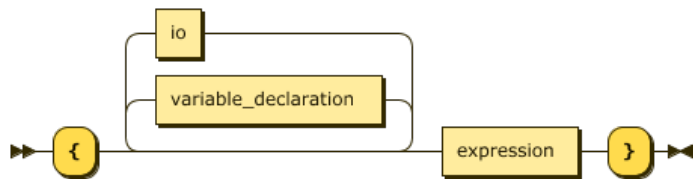


```
function_call
  ::= '(' val_id expression+ ')'
```

referenced by:

- [expression](#)

block:

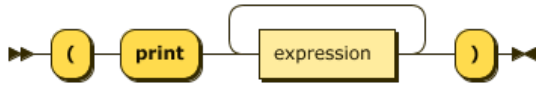


```
block ::= '{' ( variable_declaration | io )* expression '}'
```

referenced by:

- [FunctionLiteral](#)
- [conditional_expression](#)
- [function_body](#)

io:

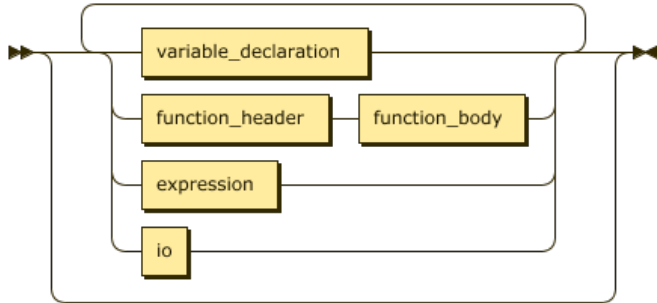


io ::= '(' 'print' expression+ '')

referenced by:

- [block](#)
- [main](#)

main:



main ::= (variable_declaration | function_header function_body | expression | io)*

no references

... generated by [Railroad Diagram Generator](#)