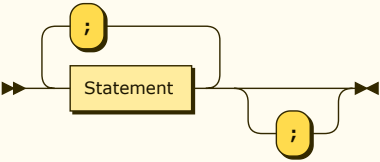


Statements:

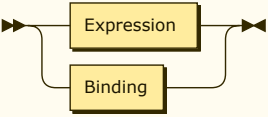


Statements ::= Statement (';' Statement) * ';' ?

referenced by:

- [Binding](#)
- [IfExpression](#)
- [LambdaExpression](#)
- [LetExpression](#)
- [LoopExpression](#)
- [SwitchExpression](#)

Statement:

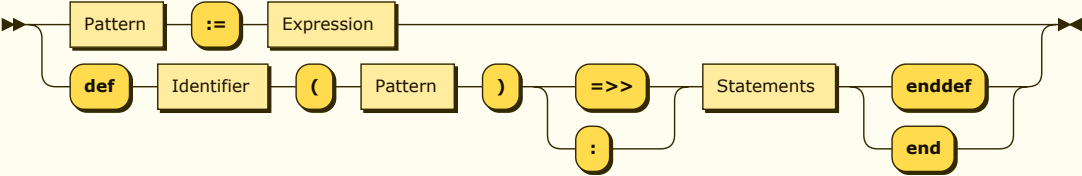


Statement ::= Expression | Binding

referenced by:

- [Statements](#)

Binding:

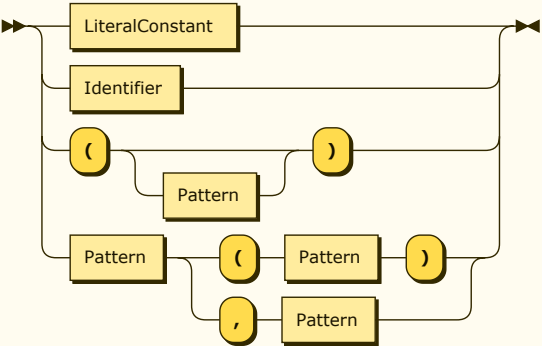


Binding ::= Pattern ':' '=' Expression | 'def' Identifier '(' Pattern ')' ('=>' | ':') Statements ('enddef' | 'end')

referenced by:

- [Query](#)
- [Statement](#)

Pattern:

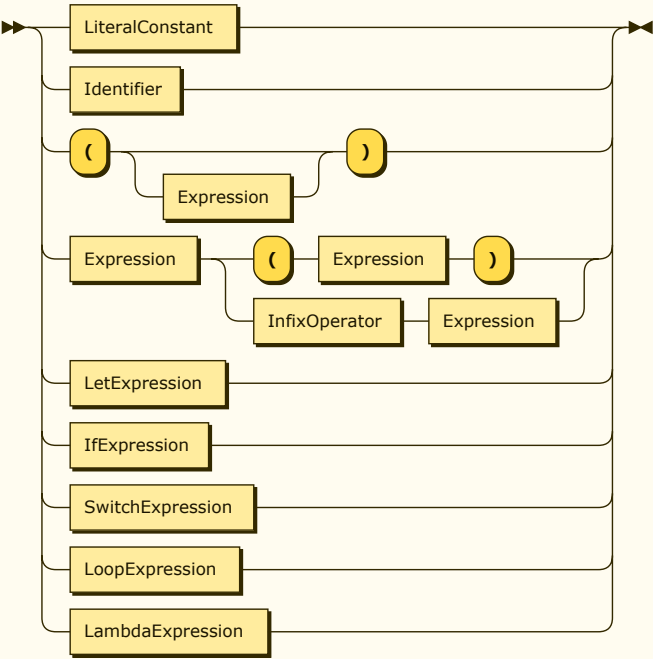


Pattern ::= LiteralConstant | Identifier | '(' Pattern? ')' | Pattern ('(' Pattern ')' | ',' Pattern)

referenced by:

- [Binding](#)
- [LambdaExpression](#)
- [Pattern](#)
- [SwitchExpression](#)

Expression:

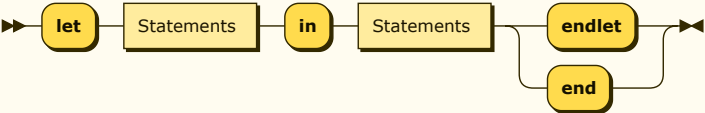


```
Expression
 ::= LiteralConstant
    | Identifier
    | '(' Expression? ')'
    | Expression ( '(' Expression ')' | InfixOperator Expression )
    | LetExpression
    | IfExpression
    | SwitchExpression
    | LoopExpression
    | LambdaExpression
```

referenced by:

- [Binding](#)
- [Expression](#)
- [IfExpression](#)
- [Query](#)
- [Statement](#)
- [SwitchExpression](#)

LetExpression:

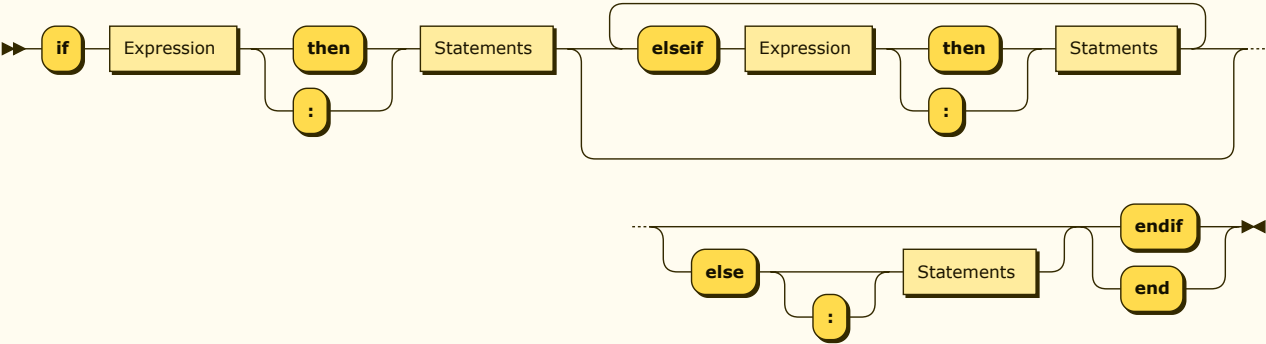


```
LetExpression
 ::= 'let' Statements 'in' Statements ( 'endlet' | 'end' )
```

referenced by:

- [Expression](#)

IfExpression:

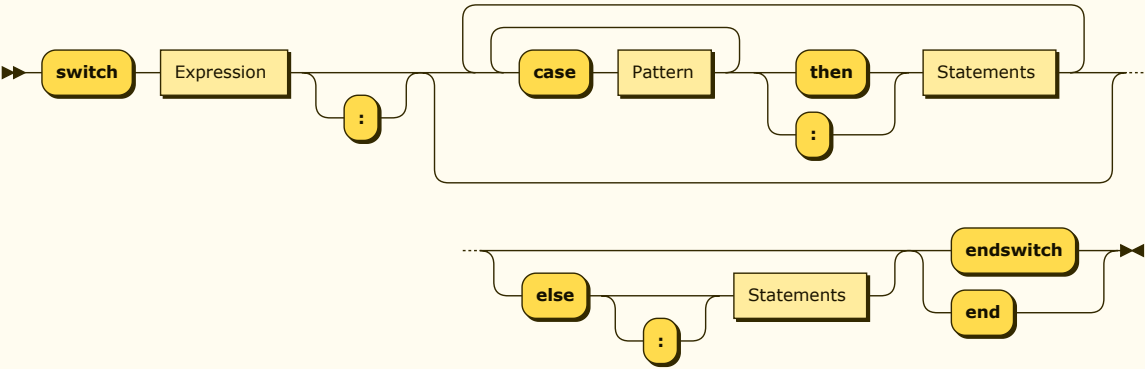


```
IfExpression
 ::= 'if' Expression ( 'then' | ':' ) Statements ( 'elseif' Expression ( 'then' | ':' ) Statements ) * ( 'else' ':'? Statements )? ( 'endif' | 'end' )
```

referenced by:

- [Expression](#)

SwitchExpression:

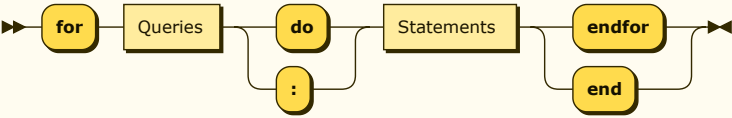


```
SwitchExpression ::= 'switch' Expression ':'? ( ( 'case' Pattern )+ ( 'then' | ':' ) Statements )* ( 'else' ':'? Statements )? ( 'endswitch' | 'end' )
```

referenced by:

- [Expression](#)

LoopExpression:

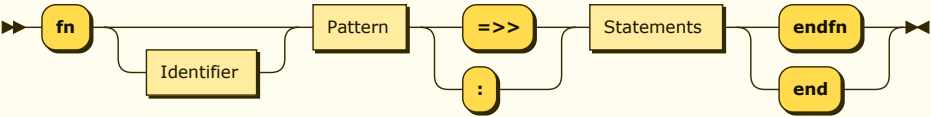


```
LoopExpression ::= 'for' Queries ( 'do' | ':' ) Statements ( 'endfor' | 'end' )
```

referenced by:

- [Expression](#)

LambdaExpression:

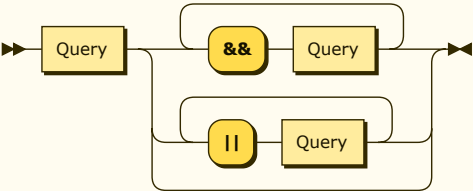


```
LambdaExpression ::= 'fn' Identifier? Pattern ( '=>' | ':' ) Statements ( 'endfn' | 'end' )
```

referenced by:

- [Expression](#)

Queries:

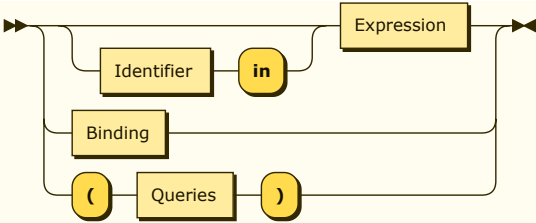


```
Queries ::= Query ( ( '&&' Query )* | ( '||' Query )+ )
```

referenced by:

- [LoopExpression](#)
- [Query](#)

Query:

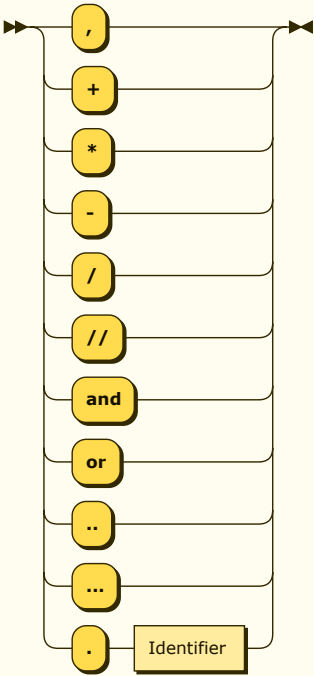


Query ::= (Identifier 'in')? Expression
 | Binding
 | '(' Queries ')'

referenced by:

- [Queries](#)

InfixOperator:

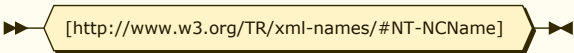


InfixOperator ::= ' , '
 | ' + '
 | ' * '
 | ' - '
 | ' / '
 | ' // '
 | ' and '
 | ' or '
 | ' .. '
 | ' ... '
 | ' . ' Identifier

referenced by:

- [Expression](#)

Identifier:

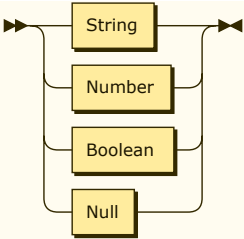


Identifier ::= [http://www.w3.org/TR/xml-names/#NT-NCName]

referenced by:

- [Binding](#)
- [Expression](#)
- [InfixOperator](#)
- [LambdaExpression](#)
- [Pattern](#)
- [Query](#)

LiteralConstant:



```
LiteralConstant
  ::= String
  | Number
  | Boolean
  | Null
```

referenced by:

- [Expression](#)
- [Pattern](#)