

Zolon Grammar

ZolonScript = Expression+

Expression = '(' Expression ')'
 | Function
 | InfixOperation
 | PrefixOperation
 | Literal
 | Identifier
 | List

Function = Identifier '->' (Expression ':' Expression '|')* Expression

InfixOperation = Expression InfixOperator Expression

InfixOperator = AssignmentOperator
 | PipeOperator
 | ArithmeticOperator
 | ComparisonOperator
 | BooleanOperator

PrefixOperation = PrefixOperator Expression

PrefixOperator = 'not'

AssignmentOperator = '='

PipeOperator = '|>'

ArithmeticOperator = '+' | '-' | '*' | '/'

ComparisonOperator = '<' | '>' | '<=' | '>=' | '!=' | '=='

BooleanOperator = 'and' | 'or'

Literal = NumberLiteral | BooleanLiteral

NumberLiteral = IntegerLiteral ('.' IntegerLiteral)?

IntegerLiteral = (0..9)+

BooleanLiteral = 'true' | 'false'

Identifier = IdentifierCharacter (0..9 | IdentifierCharacter)*

IdentifierCharacter = ' _ ' | A..Z | a..z

List = '[' Expression* ']'

Sample Derivations

incr = x -> x + 1

ZolonScript
= Expression
= InfixOperation
= Expression InfixOperator Expression
= Identifier AssignmentOperator Function
= IdentifierCharacter IdentifierCharacter IdentifierCharacter IdentifierCharacter '=' Identifier
'->' Expression
= 'i' 'n' 'c' 'r' '=' IdentifierCharacter '->' BinaryOperation
= 'i' 'n' 'c' 'r' '=' 'x' '->' Expression InfixOperator Expression
= 'i' 'n' 'c' 'r' '=' 'x' '->' Identifier ArithmeticOperator Literal
= 'i' 'n' 'c' 'r' '=' 'x' '->' IdentifierCharacter '+' NumberLiteral
= 'i' 'n' 'c' 'r' '=' 'x' '->' 'x' '+' IntegerLiteral
= 'i' 'n' 'c' 'r' '=' 'x' '->' 'x' '+' '1'
= 'incr = x -> x + 1'

**a = y ->
y > 5: y - 5 |
y**

ZolonScript
= Expression
= InfixOperation
= Expression InfixOperator Expression
= Identifier AssignmentOperator Function
= IdentifierCharacter '=' Identifier '->' Expression ':' Expression '|' Expression
= 'a' '=' 'y' '->' InfixOperation ':' InfixOperation '|' Identifier
= 'a' '=' 'y' '->' Expression InfixOperator Expression ':' Expression InfixOperator Expression '|' IdentifierCharacter
= 'a' '=' 'y' '->' Identifier ComparisonOperator Literal ':' Identifier ArithmeticOperator Literal '|' 'y'
= 'a' '=' 'y' '->' IdentifierCharacter '>' NumberLiteral ':' IdentifierCharacter '-' NumberLiteral '|' 'y'
= 'a' '=' 'y' '->' 'y' '>' '5' ':' 'y' '-' '5' '|' 'y'
= 'a = y -> y > 5: y - 5 | y'

Lexeme Types

L_PARENTHESIS:

(

R_PARENTHESIS:

)

L_BRACKET:

[

R_BRACKET:

]

ASSIGNMENT_OPERATOR:

=

PIPE_OPERATOR:

|>

ARITHMETIC_OPERATOR:

+

-

*

/

COMPARISON_OPERATOR:

<

>

<=

>=

!=

==

BOOLEAN_OPERATOR:

and

or

PREFIX_OPERATOR:

not

COLON: *

:

BAR: *

|

MAPPING_OPERATOR:

->

NUMBER_LITERAL:

(0..9)+ ('.' (0..9)+)?

BOOLEAN_LITERAL:

true

false

IDENTIFIER:

(_ A-Z a-z)+ (0-9 _ A-Z a-z)*