

Inicio

<init> ::= <instrucciones>*

<instrucciones> ::= <instruccion>+

Instrucciones generales

<instruccion> ::= <print>

| <println>

| <declaracionMultiple>

| <declaracionMultipleSimple>

| <declaracionMultipleSinTipo>

| <asignacionMultiple>

| <structDef>

| <structInst>

| <accesoStruct>

| <asigStruct>

| <fnSinParametro>

| <fnConParametro>

| <llamadaFuncionesSinParametro>

| <llamadaFuncionesConParametro>

| <retorno>

| <asigIncre>

| <asigDecre>

| <asignacion>

| <sif>

| <sfor>

| <sSwitch>

| <break>

| <sliceDef>

| <declaracionSliceVacio>

| <modificacionElementoSlice>

Impresiones

<print> ::= 'print' '(' <listaExpr> ')'

<println> ::= 'println' '(' <listaExpr> ')'

Declaraciones

<declaracionMultiple> ::= 'mut' <listaIDS> <tipos> '=' <listaExpr>

<declaracionMultipleSimple> ::= 'mut' <listaIDS> <tipos>

<declaracionMultipleSinTipo> ::= 'mut' <listaIDS> '=' <listaExpr>

<asignacionMultiple> ::= <listaIDS> '=' <listaExpr>

Struct

<structDef> ::= 'struct' IDENTIFICADOR '{' <atributos>+ '}'

<atributos> ::= <tipos> IDENTIFICADOR ';'

<structInst> ::= IDENTIFICADOR '=' IDENTIFICADOR <structInit>

<structInit> ::= '{' <listaStructs> '}'

<listaStructs> ::= <expresion> (',' <expresion>)*

<accesoStruct> ::= IDENTIFICADOR '.' IDENTIFICADOR

<asigStruct> ::= IDENTIFICADOR '.' IDENTIFICADOR '=' <expresion>

Funciones

<fnSinParametro> ::= 'fn' IDENTIFICADOR '(') <tipoRetorno>? '{' <bloqueFuncion> '}'

<fnConParametro> ::= 'fn' IDENTIFICADOR '(' <listaPar> ')' <tipoRetorno>? '{' <bloqueFuncion> '}'

<bloqueFuncion> ::= (<instruccion> | <expresion>)*

<llamadaFuncionesSinParametro> ::= IDENTIFICADOR '(')

<llamadaFuncionesConParametro> ::= IDENTIFICADOR '(' <listaExpr> ')'

Slice

<sliceDef> ::= IDENTIFICADOR '=' <sliceLiteral>

<sliceLiteral> ::= '[' <tipos> '{' <listaExpr> '}'

| '[]' <tipos> '{' <listaExprList> '}'

<accesoElementoSlice> ::= IDENTIFICADOR '[' <expresion> ']'

| IDENTIFICADOR '[' <expresion> ']' '[' <expresion> ']'

<modificacionElementoSlice> ::= IDENTIFICADOR '[' <expresion> ']' '=' <expresion>
| IDENTIFICADOR '[' <expresion> ']' '[' <expresion> ']' '=' <expresion>

<declaracionSliceVacio> ::= 'mut' IDENTIFICADOR '[' <tipos>
| 'mut' IDENTIFICADOR '[' <tipos>

Llamadas nativas

<fnAtoi> ::= 'atoi' '(' <listaExpr> ')'
<fnParseToFloat> ::= 'parseFloat' '(' <listaExpr> ')'
<fnTypeOf> ::= 'typeof' '(' <listaExpr> ')'
<fnIndexOf> ::= 'indexOf' '(' <listaExpr> ')'
<fnJoin> ::= 'join' '(' <listaExpr> ')'
<fnLen> ::= 'len' '(' <listaExpr> ')'
<fnAppend> ::= 'append' '(' <listaExpr> ')'

Asignaciones

<asigIncre> ::= IDENTIFICADOR '+=' <expresion>
<asigDecre> ::= IDENTIFICADOR '-=' <expresion>
<asignacion> ::= IDENTIFICADOR '=' <expresion>
| IDENTIFICADOR '++'
| IDENTIFICADOR '--'

Listas

<listaIDS> ::= IDENTIFICADOR '(' IDENTIFICADOR)*
<listaExpr> ::= <expresion> '(' <expresion>)*
<listaExprList> ::= '{' <listaExpr> '}' ('{' <listaExpr> '}')* ','?
<listaPar> ::= <parametro> '(' <parametro>)*
<parametro> ::= IDENTIFICADOR <tipos>

Expresiones

<expresion> ::= '-' <expresion>

| '!' <expresion>

| <expresion> '%' <expresion>

| <expresion> '/' <expresion>

| <expresion> '*' <expresion>

| <expresion> '-' <expresion>

| <expresion> '+' <expresion>

| <expresion> '!=' <expresion>

| <expresion> '==' <expresion>

| <expresion> '<=' <expresion>

| <expresion> '>=' <expresion>

| <expresion> '<' <expresion>

| <expresion> '>' <expresion>

| <expresion> '||' <expresion>

| <expresion> '&&' <expresion>

| ENTERO

| DECIMAL

| CADENA

| RUNE

| 'true'

| 'false'

| <accesoElementoSlice>

| '[' <listaExpr>? ']'

| IDENTIFICADOR

| '(' <expresion> ')'

| <fnAtoi>

| <fnParseToFloat>

| <fnTypeOf>

- | <accesoStruct>
- | <llamadaFuncionesSinParametro>
- | <llamadaFuncionesConParametro>
- | <fnAppend>
- | <fnIndexOf>
- | <fnJoin>
- | <fnLen>

Condicionales

<sif> ::= 'if' '(' ('<expresion>' | <expresion>) <bloque> <elseifPart>* ('else' <bloque>)?

<elseifPart> ::= 'else if' '(' ('<expresion>' | <expresion>) <bloque>

<bloque> ::= '{' <instrucciones>* '}'

Ciclos

<sfor> ::= 'for' <expresion> <bloque>

| 'for' <asignacion> ';' <expresion> ';' <asignacion> <bloque>

| 'for' IDENTIFICADOR ',' IDENTIFICADOR '=' 'range' <expresion> <bloque>

Switch

<sSwitch> ::= 'switch' <expresion> '{' <caseBlock>* <defaultBlock>? '}'

<caseBlock> ::= 'case' <expresion> ':' <instrucciones>

<defaultBlock> ::= 'default' ':' <instrucciones>

Otros

<retorno> ::= 'return' <expresion> ';'?

<break> ::= 'break'

<continue> ::= 'continue'

<tipos> ::= 'int' | 'float64' | 'string' | 'bool' | IDENTIFICADOR

<tipoRetorno> ::= <tipos> | 'rune'