

EXPRESIONES REGULARES

"toCharArray" { 'CHARARRAY' }

"new" { 'NEW' }

"(int)" { 'CASTEOINT' }

"(double)" { 'CASTEODOUBLE' }

"toString" { 'CASTEOSTRING' }

"(char)" { 'CASTEOCHAR' }

"typeof" { 'CASTEOTIPO' }

"toLowerCase" { 'CASTEOTOLOWER' }

"toUpperCase" { 'CASTEOTOUPPER' }

"length" { 'LENGTH' }

"round" { 'ROUND' }

truncate { 'TRUNCATE' }

"--" { 'DECRE' }

"++" { 'INCRE' }

"==" { 'IGUALIGUAL' }

"^" { 'POT' }

"%" { 'MOD' }

"(" { 'PARA' }

")" { 'PARC' }

"[" { 'CORA' }

"]" { 'CORC' }

"," { 'PYC' }

"=" { 'IGUAL' }

"," { 'COMA' }

"{" { 'LLAVA' }

"}" { 'LLAVC' }

"?" { 'INTERROGACION' }

"." { 'DSPNTS' }

/* OPERADORES ARITMETICOS */

"+" { 'MAS' }

"*" { 'MULTI' }

"/" { 'DIV' }

"-" { 'MENOS' }

"%" { 'MOD' }

"^" { 'POT' }

/* OPERADORES RELACIONALES */

">=" { 'MAYORIGUAL' }

">" { 'MAYORQUE' }

"<=" { 'MENORIGUAL' }

"<" { 'MENORQUE' }

"!=" { 'DIFERENTE' }

/* OPERADORES LOGICOS */

"&&" { 'AND' }

"||" { 'OR' }

"!" { 'NOT' }

"list" { 'LIST' }

/*Palabras reservadas*/

"evaluar" { 'EVALUAR' }

"true" { 'TRUE' }

"false" { 'FALSE' }

"int" { 'INT' }
"double" { 'DOUBLE' }
"string" { 'STRING' }
"char" { 'CHAR' }
"boolean" { 'BOOLEAN' }

"writeline" { 'WRITELINE' }
"println" { 'PRINT' }
"print" { 'PRINTLN' }

"if" { 'IF' }
"else if" { 'ELSEIF' }
"while" { 'WHILE' }
"do" { 'DO' }
"until" { 'UNTIL' }
"else" { 'ELSE' }
"break" { 'BREAK' }

"for" { 'FOR' }
"switch" { 'SWITCH' }
"case" { 'CASE' }
"tostring" { 'TOSTRING' }
"toupper" { 'TOUPPER' }
"default" { 'DEFAULT' }

"continue" { 'CONTINUE' }

"main" { 'MAIN' }
"void" { 'VOID' }

"return" { 'RETURN' }

//SIMBOLOS ER

[0-9]+(("[0-9]+)\b { 'DECIMAL' }

{num} { 'ENTERO' }

{id} { 'ID' }

{cadena} { 'CADENA' }

{caracter} { 'CHARACTER' }

Terminales

Evaluar, aritmetica ,primitivo ,writeline , declaracion , ast , tipo, identificador, relacional , lógica, asignación, Ifs, While, DoWhile, ternario, detener, Switch, caso, For, continuar, función, llamada, startwith, símbolo, retorno, errores

No terminales

'INTERROGACION'

'OR'

'AND'

'NOT' 'CASTEODOUBLE' 'CASTEOINT' 'CASTEOSTRING' 'CASTEOCHAR' 'CASTEOTIPO'
'CASTEOTOLOWER' 'CASTEOTOUPPER' 'LENGTH' 'ROUND' 'TRUNCATE'

'IGUALIGUAL' 'DIFERENTE' 'MENORQUE' 'MENORIGUAL' 'MAYORQUE' 'MAYORIGUAL'

'MAS' 'MENOS'

'DIV' 'MULTI'

'POT'

'MOD'

UMINUS

Gramática

%start inicio

inicio : instrucciones EOF

;

instrucciones : instrucciones instruccion

| instruccion

;

instruccion : declaracion

| startwith

| writeline

| asignacion

| sent_if

| sent_while

| sent_Dowhile

| BREAK PYC

| sent_switch

| sent_for

| ID DECREASE PYC

| ID INCREASE PYC

| CONTINUE PYC

| funciones

| llamada PYC

| RETURN PYC

| RETURN e PYC

| error

;

declaracion : tipo lista_ids IGUAL e PYC

| tipo lista_ids PYC

| tipo CORA CORC lista_ids IGUAL NEW tipo CORA e CORC PYC

| tipo CORA CORC lista_ids IGUAL LLAVA listasimple LLAVC PYC

| tipo CORA CORC lista_ids IGUAL e PYC

| LIST MENORQUE tipo MAYORQUE lista_ids IGUAL NEW LIST MENORQUE tipo MAYORQUE PYC

;

doublearray: doublearray COMA LLAVA listasimple LLAVC

| LLAVA listasimple LLAVC

;

listasimple: listasimple COMA e

| e

;

tipo : INT

| DOUBLE

| STRING

| CHAR

| BOOLEAN

;

lista_ids : lista_ids COMA ID

| ID

;

writeline : PRINTLN PARA e PARC PYC

| PRINT PARA e PARC PYC

;

asignacion : ID IGUAL e PYC

| ID CORA e CORC IGUAL e PYC

| ID CORA e CORC CORA e CORC IGUAL e PYC

;

sent_if : IF PARA e PARC LLAVA instrucciones LLAVC

| IF PARA e PARC LLAVA instrucciones LLAVC ELSE LLAVA instrucciones LLAVC

| ELIF PARA e PARC LLAVA instrucciones LLAVC sent_else

;

sent_else :

ELSE LLAVA instrucciones LLAVC

|

;

sent_while : WHILE PARA e PARC LLAVA instrucciones LLAVC

;

sent_Dowhile : DO LLAVA instrucciones LLAVC WHILE PARA e PARC PYC

| DO LLAVA instrucciones LLAVC UNTIL PARA e PARC PYC

;

sent_for : FOR PARA dec_asig_for PYC e PYC actualizacion_for PARC LLAVA instrucciones LLAVC

;

dec_asig_for : tipo ID IGUAL e

| ID IGUAL e

;

actualizacion_for : ID DECRE

| ID INCRE

| ID IGUAL e

;

sent_switch : SWITCH PARA e PARC LLAVA caselist LLAVC

| SWITCH PARA e PARC LLAVA caselist default LLAVC

| SWITCH PARA e PARC LLAVA default LLAVC

;

caselist : caselist caso

| caso

;

caso : CASE e DSPNTS instrucciones

;

default : DEFAULT DSPNTS instrucciones

;

funciones

: ID PARA lista_params PARC DSPNTS tipo LLAVA instrucciones LLAVC

| ID PARA PARC DSPNTS tipo LLAVA instrucciones LLAVC

| ID PARA lista_params PARC DSPNTS VOID LLAVA instrucciones LLAVC

| ID PARA PARC DSPNTS VOID LLAVA instrucciones LLAVC

;

lista_params : lista_params COMA tipo ID

| tipo ID

| tipo CORA CORC ID

| tipo CORA CORC CORA CORC ID

;

llamada : ID PARA lista_vals PARC

| ID PARA PARC

;

lista_vals : lista_vals COMA e

| e

;

startwith : RUN llamadaeee PYC

;

llamadaeee : ID PARA lista_vals PARC

| ID PARA PARC

;

e : e MAS e

| e MENOS e

| e MULTI e

| e DIV e

| e POT e

| e MOD e

| e MAYORIGUAL e

| e MAYORQUE e

| e MENORIGUAL e

| e MENORQUE e

| e IGUALIGUAL e

| e DIFERENTE e

| e AND e

| e OR e

| NOT e
| CASTEODOUBLE e
| CASTEOINT e
| CASTEOSTRING PARA e PARC
| CASTEOCHAR e
| CASTEOTIPO PARA e PARC
| CASTEOTOLOWER PARA e PARC
| CASTEOTOUPPER PARA e PARC
| LENGTH PARA e PARC
| CHARARRAY PARA e PARC
| ROUND PARA e PARC
| MENOS e %prec UMINUS
| PARA e PARC
| DECIMAL
| ENTERO
| ID
| ID CORA e CORC
| ID CORA e CORC CORA e CORC
| CADENA
| CHARACTER
| TRUE
| FALSE
| e INTERROGACION e DSPNTS e
| ID INCRE
| ID DECRE
| llamada
;