

INICIO : INSTRUCCIONES EOF

INSTRUCCIONES : INSTRUCCIONES INSTRUCCION

INSTRUCCION :

VECTOR

|DECLARACION

|ASIGNACION

|SENTENCIA_CASTEO

|SENTENCIA_IF

|SENTENCIA_SWITCH

|SENTENCIA_WHILE

|SENTENCIA_FOR

|SENTENCIA_DO_WHILE

|SENTENCIA_DO_UNTIL

|SENTENCIA_RETURN

|FUNCION

|METODO

|SENTENCIA_LLAMADA

|SENTENCIA_IMPRIMIR

|FUNCION_TOLOWER

|FUNCION_Toupper

|FUNCION_ROUND

|FUNCION_LENGTH

|FUNCION_TYPEOF

|FUNCION_TOSTRING

|FUNCION_TOCHARARRAY

|SENTENCIA_INCREMENTO

|SENTENCIA_TRANSFERENCIA

- | SENTENCIA_PUSH
- | SENTENCIA_POP
- | SENTENCIA_RUN
- | error_punto_coma
- | error
- | error_EOF

DECLARACION :

- TIPO_VARIABLE LISTA_ID punto_coma
- | TIPO_VARIABLE LISTA_ID igual OPERACION punto_coma
- | TIPO_VARIABLE LISTA_ID igual SENTENCIA_TERNARIA punto_coma

ASIGNACION :

- LISTA_ID igual OPERACION punto_coma
- | id cor_izq OPERACION cor_der igual OPERACION punto_coma
- | LISTA_ID igual SENTENCIA_TERNARIA punto_coma

SENTENCIA_CASTEO :

- TIPO_VARIABLE LISTA_ID igual par_izq TIPO_VARIABLE par_der OPERACION punto_coma

TIPO_VARIABLE :

- string
- | int
- | double
- | boolean
- | char

;

OPERACION :

//OPERACIONES ARITMETICAS

'menos' OPERACION %prec UMENOS

|OPERACION mas OPERACION

|OPERACION menos OPERACION

|OPERACION por OPERACION

|OPERACION div OPERACION

|OPERACION mod OPERACION

|OPERACION potencia OPERACION

|par_izq OPERACION par_der

//OPERACIONES DE INCREMENTO

|id mas_mas

|id menos_menos

//valores primitivos

|id

|numero

|cadena

|caracter

|true

|false

|LLAMADA_ARITMETICA

//OPERADORES RELACIONALES

|OPERACION mayor OPERACION

|OPERACION menor OPERACION

|OPERACION mayor_igual OPERACION

|OPERACION menor_igual OPERACION

|OPERACION igual_que OPERACION

| OPERACION dif_que OPERACION

//OPERADORES LOGICOS

| OPERACION and OPERACION

| OPERACION or OPERACION

| not OPERACION

//ACCESO A VECTORES

| id cor_izq OPERACION cor_der

//| id cor_izq OPERACION_ARITMETICA cor_der cor_izq OPERACION_ARITMETICA cor_der

| id punto length par_izq par_der

| id punto pop par_izq par_der {

LISTA_ID :

LISTA_ID coma id

| id

SENTENCIA_INCREMENTO:

id mas_mas punto_coma

//| mas_mas id punto_coma

| id menos_menos punto_coma

//| menos_menos id punto_coma

VECTOR :

TIPO_VARIABLE cor_izq cor_der id igual new TIPO_VARIABLE cor_izq PARAMETROS_LLAMADA
cor_der punto_coma

```

|TIPO_VARIABLE cor_izq cor_der id igual new TIPO_VARIABLE cor_izq cor_der punto_coma

| TIPO_VARIABLE cor_izq cor_der cor_izq cor_der id igual llave_izq llave_izq
PARAMETROS_LLAMADA llave_der coma llave_izq PARAMETROS_LLAMADA llave_der llave_der
punto_coma

| TIPO_VARIABLE cor_izq cor_der cor_izq cor_der id igual llave_izq
llave_izq llave_der coma llave_izq llave_der llave_der punto_coma
;

```

//CICLO IF

SENTENCIA_IF :

```

if par_izq OPERACION par_der llave_izq BLOQUE llave_der LISTA_ELIF

|if par_izq OPERACION par_der llave_izq BLOQUE llave_der

```

LISTA_ELIF :

```

else llave_izq BLOQUE llave_der

|elif par_izq OPERACION par_der llave_izq BLOQUE llave_der LISTA_ELIF

|elif par_izq OPERACION par_der llave_izq BLOQUE llave_der

```

// ciclo switch

SENTENCIA_SWITCH :

```

switch par_izq OPERACION par_der llave_izq CASOS llave_der

```

;

CASOS :

```

CASOS case OPERACION dos_puntos INSTRUCCIONES

|case OPERACION dos_puntos INSTRUCCIONES

|CASOS default dos_puntos INSTRUCCIONES

```

//CICLO WHILE

SENTENCIA_WHILE :

```
while par_izq OPERACION par_der llave_izq BLOQUE llave_der  
;
```

SENTENCIA_DO_WHILE :

```
do llave_izq BLOQUE llave_der while par_izq OPERACION par_der punto_coma
```

SENTENCIA_DO_UNTIL :

```
do llave_izq BLOQUE llave_der until par_izq OPERACION par_der punto_coma
```

//ciclo for

SENTENCIA_FOR :

```
for par_izq SENTENCIA_FOR1 OPERACION punto_coma SENTENCIA_FOR2 par_der llave_izq  
BLOQUE llave_der
```

SENTENCIA_FOR1 :

DECLARACION

|ASIGNACION

;

SENTENCIA_FOR2 :

id mas_mas

|id menos_menos

|id igual OPERACION

;

// OPERADOR TERNARIO

SENTENCIA_TERNARIA :

```
OPERACION interrogacion2 OPERACION dos_puntos OPERACION
```

SENTENCIA_TERNARIA2 :

OPERACION interrogacion2 INSTRUCCION dos_puntos INSTRUCCION

;

SENTENCIA_RETURN :

return OPERACION punto_coma

|return punto_coma

FUNCION :

id par_izq PARAMETROS_FUNCION par_der dos_puntos TIPO_VARIABLE llave_izq BLOQUE
llave_der

|id par_izq par_der dos_puntos TIPO_VARIABLE llave_izq BLOQUE llave_der

PARAMETROS_FUNCION :

PARAMETROS_FUNCION coma TIPO_VARIABLE id

| TIPO_VARIABLE id

METODO :

id par_izq par_der dos_puntos void llave_izq BLOQUE llave_der

|id par_izq PARAMETROS_FUNCION par_der dos_puntos void llave_izq BLOQUE llave_der

|id par_izq par_der llave_izq BLOQUE llave_der

SENTENCIA_LLAMADA :

id par_izq par_der punto_coma

|id par_izq PARAMETROS_LLAMADA par_der punto_coma

LLAMADA_ARITMETICA :

id par_izq par_der

|id par_izq PARAMETROS_LLAMADA par_der

;

PARAMETROS_LLAMADA :

PARAMETROS_LLAMADA coma OPERACION

|OPERACION

SENTENCIA_IMPRIMIR :

print par_izq OPERACION par_der punto_coma

|println par_izq OPERACION par_der punto_coma

SENTENCIA_TRANSFERENCEIA:

break punto_coma

|continue punto_coma

FUNCION_TOLOWER :

TIPO_VARIABLE LISTA_ID igual toLower par_izq OPERACION par_der punto_coma

FUNCION_TOUPPER :

TIPO_VARIABLE LISTA_ID igual toUpper par_izq OPERACION par_der punto_coma

FUNCION_ROUND :

TIPO_VARIABLE LISTA_ID igual round par_izq OPERACION par_der punto_coma

FUNCION_LENGTH :

TIPO_VARIABLE LISTA_ID igual length par_izq OPERACION par_der punto_coma

FUNCION_TYPEOF :

TIPO_VARIABLE LISTA_ID igual typeof par_izq OPERACION par_der punto_coma

FUNCION_TOSTRING :

TIPO_VARIABLE LISTA_ID igual toStringA par_izq OPERACION par_der punto_coma

FUNCION_TOCHARARRAY :

TIPO_VARIABLE cor_izq cor_der id igual toCharArray par_izq cadena par_der punto_coma

SENTENCIA_PUSH :

id punto push par_izq OPERACION par_der punto_coma

SENTENCIA_POP :

id punto pop par_izq par_der punto_coma

SENTENCIA_RUN :

run SENTENCIA_LLAMADA

BLOQUE

: INSTRUCCIONES