Tabla de Tokens de Ruby

|  |  |  |
| --- | --- | --- |
| TOKEN | PATRÓN | LEXEMA |
| require | require | require |
| nil | nil | nil |
| when | when | when |
| def | def | def |
| false | false | false |
| not | not | not |
| while | while | while |
| for | for | for |
| then | then | then |
| do | do | do |
| if | if | if |
| true | true | true |
| begin | begin | begin |
| else | else | else |
| break | break | break |
| elsif | elsif | elsif |
| retry | retry | retry |
| unless | unless | unless |
| case | case | case |
| end | end | end |
| return | return | return |
| until | until | until |
| comma | , | , |
| semicolon | ; | ; |
| escaped\_quote | \\” | \\” |
| crlf | \n | \n |
| plus | + | + |
| minus | - | - |
| div | / | / |
| mul | \* | \* |
| mod | % | % |
| exp | \*\* | \*\* |
| equal | == | == |
| not\_equal | != | != |
| greater | > | > |
| less | < | < |
| less\_equal | <= | <= |
| greater\_equal | >= | >= |
| assign | = | = |
| pluss\_assign | += | += |
| minus\_assign | -= | -= |
| mul\_assign | \*= | \*= |
| div\_assign | /= | /= |
| mod\_assign | %= | %= |
| exp\_assign | \*\*= | \*\*= |
| bit\_and | & | & |
| bit\_or | | | | |
| bit\_xor | ^ | ^ |
| bit\_not | ~ | ~ |
| bit\_shl | << | << |
| bit\_shr | >> | >> |
| and | and | && | and |
| or | or | | | | or |
| not | not | ! | not |
| left\_rbracket | ( | ( |
| right\_rbracket | ) | ) |
| rigth\_sbracket | [ | [ |
| left\_sbracket | ] | ] |
| INT | [0,9]+ | 1 |
| FLOAT | [0-9]+'.'[0-9]+; | 1.1 |
| DOLLAR | $ | $ |
| IDFUNCTION | [a-zA-Z\_][a-zA-Z0-9\_]\*[! | ?] | empty! |
| ID | [a-zA-Z\_][a-zA-Z0-9\_]\* | hola |

**\***El analizador léxico tendrá que tener en cuenta además los comentarios, espacios y tabulaciones.

* **SL\_COMMENT** cuyo patrón es **('#' ~('\r' | '\n)\*' '\n') -> skip;**
* **ML-COMMENT** cuyo patrón es **('=begin' .\*? '=end\n') -> skip;**
* **WS cuyo** patrón es **(‘ ‘|’\t’)+ -> skip;**