**Analizador léxico Documentación**

**Palabras Reservadas.**

|  |  |
| --- | --- |
| Palabras Reservadas | |
| Palabra | Descripción |
| protected | Declara un método o clase protegido. |
| private | Declara un método o clase privado. |
| public | Declara un método o clase público. |
| while | Para iniciar un bucle |
| for | Para iniciar un ciclo de i a n; |
| if | Para hacer una pregunta |
| else | En conjunto con ‘if’ lo que se hace si la pregunta es falsa. |
| char | Declarar una variable como carácter. |
| short | Declarar una variable como un numero entero pequeño. |
| void | Definir que un método no retorna nada. |
| long | Declarar una variable como un numero entero grande. |
| float | Declarar una variable como un numero decimal. |
| double | Declarar una variable como un entero muy grande. |
| bool | Declarar una variable como un booleano |
| int | Declarar una variable como un entero. |
| class | Declarar una clase. |
| return | Definir un punto de retorno de un método. |
| try | Clausula para intentar realizar una operación. |
| cath | Captura un error. |
| static | Declara un método o variable como estática. |

**Identificadores:**

Un identificador es cualquier nombre que define una variable, clase o método.

Las **reglas** para declarar un identificador es que este debe comenzar con una letra o un guion bajo seguido de cero más letras, guiones bajos o números.

Como se define en la expresión: **[a-zA-z\_]+( [a-zA-z\_]+ | [0-9] )\***

**Operadores Aritméticos:**

|  |  |
| --- | --- |
| Operadores Aritméticos | |
| Operador | Descripción |
| = | Asignación |
| + | suma |
| - | Resta |
| \* | Multiplicación |
| / | División |

**Ejemplos:**

**int** a = 4;

**int** b= 2;

**int** c= a + b;

//print(c) 🡺 6

**Operadores Lógicos**

|  |  |
| --- | --- |
| Operadores Lógicos | |
| Operador | Descripción |
| && | and |
| || | or |
| ! | Negación o Not |
| true | Verdadero |
| false | Falso |

**Ejemplo:**

Bool p = true;

Bool q = false;

//Print(p&&q) 🡺 false

//Print(¡p&&q) 🡺 false

**Operadores relacionales**

|  |  |
| --- | --- |
| Operadores Relacionales | |
| Operador | Descripción |
| < | Menor que |
| > | Mayor que |
| <= | Menor o igual |
| >= | Mayor o igual |
| == | Igual |
| ¡= | Diferente |

**Ejemplo:**

If(a!=b){

Print(a>=b) => true

}

**Literales**

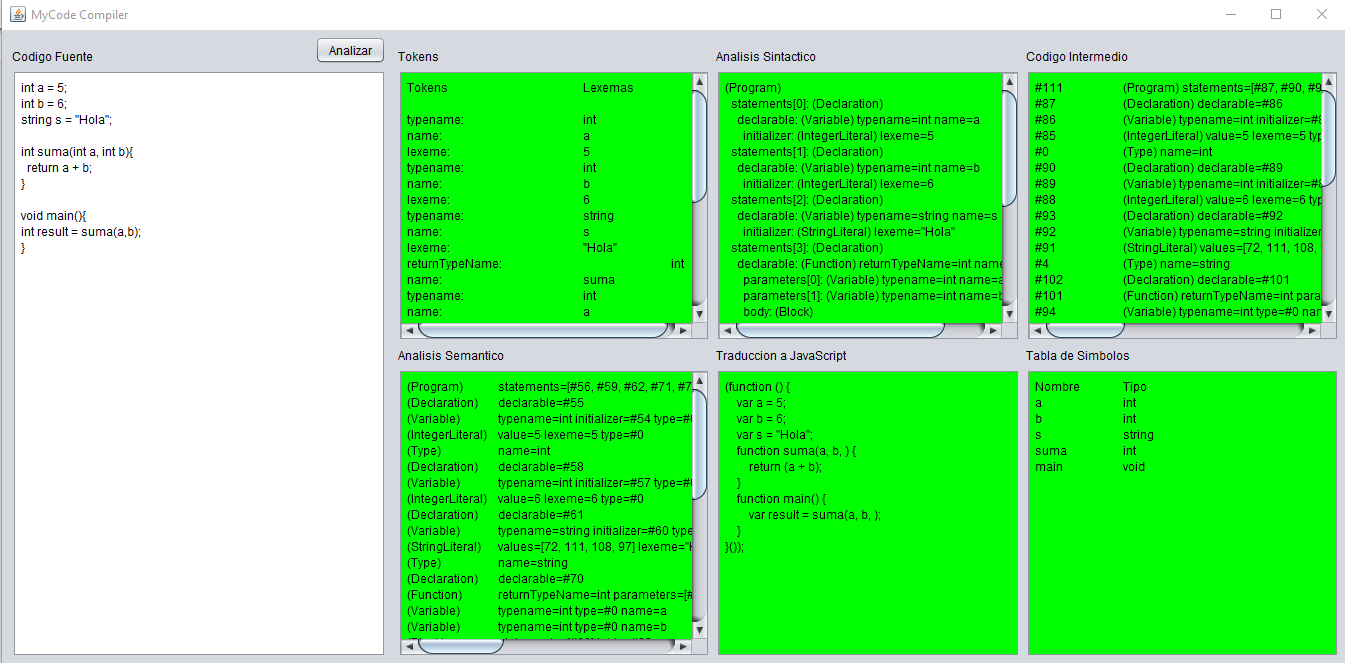
Una literal es una cadena de texto plano.

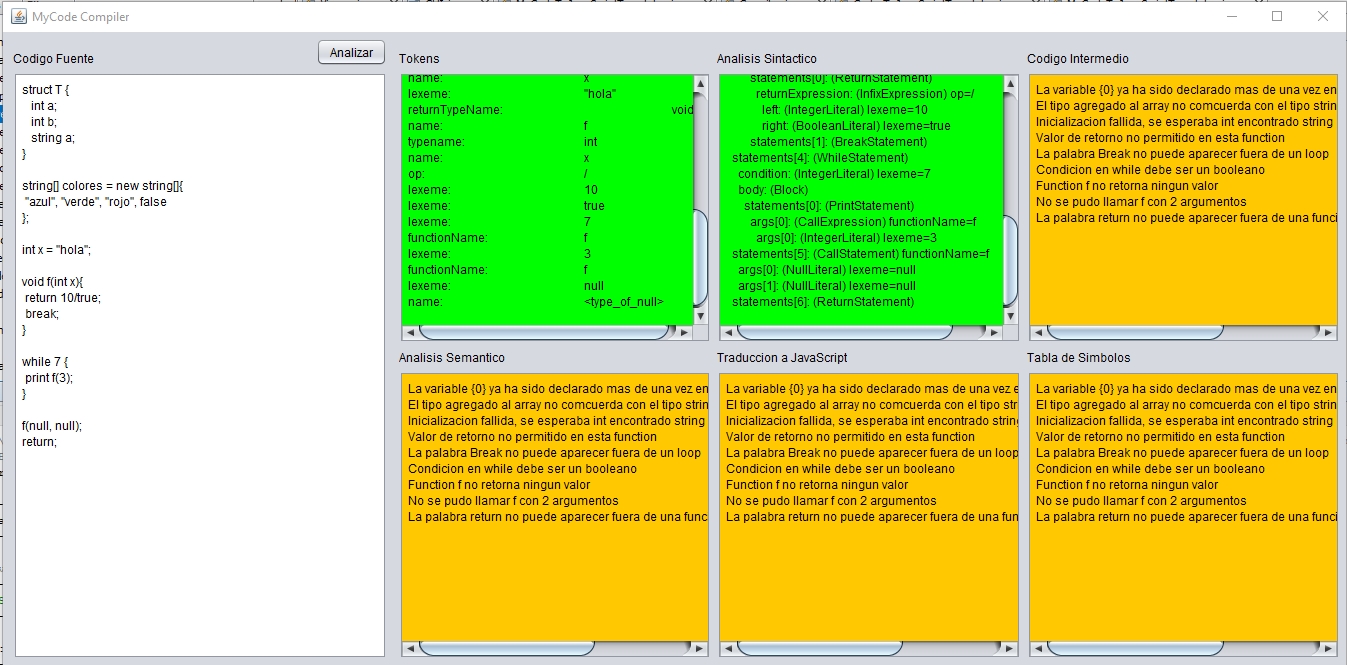
Para definir una literal se deber encerrar el texto entre comillas **(“ | ‘):**

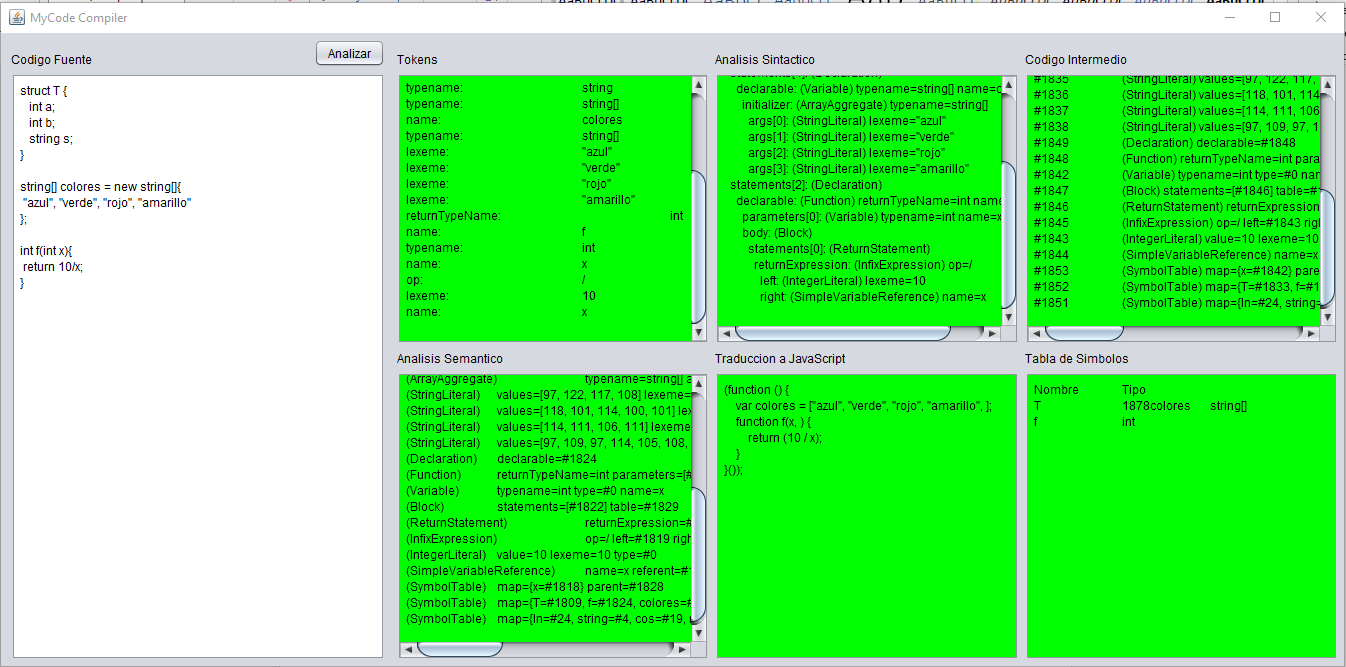
**Ejemplos:**

**String** str = “Hola Mundo”;

**Ejemplos del compilador:**

**Ejemplo #1:**

**Ejemplo #2:**

**Ejemplo #3:**