Black Cold Brew
Manual de Referencia

Versión 0.1

Miguel Zegarra

April 10, 2022

Email: armando.zegarra@ucsp.edu.pe

Abstract

Black Cold Brew es un lenguaje de programación con funciones básicas y de tipado estricto. En este manual se encuentra una referencia de sus funciones y formas en las que se puede usar. También una instrucción básica de su semántica, léxica y compilación

CONTENIDO

1 Introducción **1**
1.1 Notación 1

2 Analisis Lexico **3**
2.1 Estructura de Bloque 3
2.2 Tokens 5
2.3 Identificadores y Palabras Claves 6
2.4 Literales 6
2.5 Operadores 9
2.6 Delimitadores 10

Capítulo

UNO

Introducción

Este manual de referencia describe el lenguaje de programación Black Cold Brew. No pretende ser un tutorial.

Si bien trato de ser lo más preciso posible, elegí usar el inglés en lugar de las especificaciones formales para todo excepto la sintaxis y el análisis léxico. Esto debería hacer que el documento sea más comprensible para el lector medio, pero dejará lugar a ambigüedades. En consecuencia, si viene de una isla perdida en Júpiter donde no tienen café y trata de volver a implementar Black Cold Brew solo desde este documento, es posible que tenga que adivinar cosas y, de hecho, probablemente termine implementando un lenguaje bastante diferente. Por otro lado, si está utilizando Black Cold Brew y se pregunta cuáles son las reglas precisas sobre un área particular del lenguaje, definitivamente debería poder encontrarlas aquí. Si desea ver una definición más formal del lenguaje, tal vez podría ofrecer su tiempo como voluntario, o inventar una máquina de clonación :-).

Es peligroso agregar demasiados detalles de implementación a un documento de referencia de idioma: la implementación puede cambiar y otras implementaciones del mismo idioma pueden funcionar de manera diferente. Por otro lado, actualmente solo hay una implementación de Black Cold Brew en uso generalizado, y a veces vale la pena mencionar sus peculiaridades particulares, especialmente cuando la implementación impone limitaciones adicionales. Por lo tanto, encontrará breves "notas de implementación" esparcidas por todo el texto.

1.1 Notación

Las descripciones del análisis léxico y la sintaxis utilizan una notación gramatical estándar del lenguaje. Esto utiliza el siguiente estilo de definición:

```
DEBUG SCAN - [ INT_TYPE ] found at (1:1)
```

En este output, podemos observar el DEBUG SCAN, como el primer identificador de la línea del scanner. Después seguirá con el identificador del token e indicará donde fue encontrado, la línea y la columna.

```
DEBUG SCAN - ID [ a ] found at (1:4)
```

En el caso de los IDs, estos serán identificados desde su token y después, dentro de los corchetes, se indicará el identificador.

Análisis Léxico

Un programa de Black Cold Brew es leído por un analizador. La entrada al analizador es un flujo de tokens, generado por el analizador léxico. Este capítulo describe cómo el analizador léxico divide un archivo en tokens.

Black Cold Brew utiliza el conjunto de caracteres definidos por expresiones regulares para sus literales y coincidencias de caracteres específicos para comandos y palabras reservadas.

2.1 Estructura de Bloque

Un programa de Black Cold Brew se divide en varios bloques lógicos.

2.1.1 Bloques Logicos

El final de un bloque lógico puede estar representado por el inicio de otro bloque lógico o también por un delimitador.

2.1.2 Comentarios

Un comentario comienza con un string (/*) que no forma parte de un literal de cadena y termina con otro string(*/).

2.1.3 Espacios en Blanco y Saltos de Línea

Todas los caracteres en blanco y saltos de línea se ignoran.

2.2 Otros Tokens

Además de SALTOS DE LINEA, ESPACIOS EN BLANCO y COMENTARIOS, existen las siguientes categorías de tokens: identificadores, palabras clave, literales, operadores y delimitadores. Los caracteres de espacio en blanco (aparte de los terminadores de línea, discutidos anteriormente) no son tokens, pero sirven para delimitar tokens. Cuando existe ambigüedad, un token comprende la cadena más larga posible que forma un token legal, cuando se lee de izquierda a derecha.

2.3 Identificadores y palabras clave

Los identificadores (también denominados nombres) se describen mediante las siguientes definiciones léxicas:

```
identificador:  (letra|"_")(letra|dígito|"_")* letra: minuscula | mayuscula
minuscula:      "a"..."z"
mayuscula:      "A"..."Z"
dígito:         "0"..."9"
```

Los identificadores tienen una longitud ilimitada. Si el carácter es minuscula o mayuscula es significativo.

2.3.1 Palabras Reservadas

Los siguientes identificadores se utilizan como palabras reservadas o palabras clave del idioma y no se pueden utilizar como identificadores ordinarios. Deben ser escritos exactamente como están escritos aquí:

```
print
if
read
type
while

int
string
boolean
```

2.4 Literales

Los literales son notaciones para valores constantes de algunos tipos integrados.

2.4.1 Strings

Los literales de cadena se describen mediante las siguientes definiciones léxicas:

```
stringliteral:   shortstring | longstring
shortstring:     "’" shortstringitem* "’" | ’"’ shortstringitem* ’"’
longstring:      "’’’" longstringitem* "’’’" | ’"""’ longstringitem*
’"""’ shortstringitem: shortstringchar | escapeseq
longstringitem: longstringchar | escapeseq
shortstringchar: <any ASCII character except "\" or newline or the
quote> longstringchar: <any ASCII character except "\">
escapeseq:       "\" <any ASCII character>
```

2.4.2 Numeric literals

Solo hay un tipo de números en Black Cold Brew: Enteros.

2.4.3 Integer

Los literales enteros y enteros largos se describen mediante las siguientes definiciones léxicas:

```
integer:        (digito)*
dígito:         "0"..."9"
```

2.5 Operators

The following tokens are operators:

```
+       -       *       =       /
                &       |
                                ==          !=
```

2.6 Delimiters

The following tokens serve as delimiters in the grammar:

```
(       )       /*      */      {       }
$
```