



FEDERAL UNIVERSITY OF THE SOUTH FRONTIER
ACADEMIC: RAFAELLE LARISA DE ARRUDA

Java Interpreter
Language Rafis Chuchu

CHAPECÓ – SC
MAY 2019, BRAZIL

Language Presentation

The Rafis Chuchu language provides a flexible, imperative, object-oriented, functional, and self-level syntax. Created by @mazarafa, however, it has a dynamic trip, and its main feature is to allow easy code reading, and require few lines of code when compared to other languages.

INTRODUCTION

It is well known that this work was performed to interpret a dynamic language, written with few lines of code. Rafis is a programming language of the highest level, created by @mazarafa on the idea of "Programming of computers for all", with this development, had in mind the freedom, free code, having flexibility to be interpreted.

RAFIS CHUCHU XPRESSIONS

Arithmetic:

- +: sum
- : subtraction
- *: multiplication
- /: division
- ?: rest

Note that in the expression, we can have at least 2 operands and are sequences. Ex.: $2 + 2 * 3 = 12$. **NOTE: It starts with the first operator, adds and then multiplies.**

In the logical property of the signals:

- ! inequality
- <: smaller
- >: bigger
- &: equality

These logical expressions may contain two operands. The expression does not accept booleans.

Syntax of Language

The syntax refers to the rules that govern the composition of texts in a formal language that constitutes a logical system. Can provide an interpretation. In computer science, syntax tells rules that govern the composition of meaningful texts in a formal language, such as a programming language, making sense to define semantics or meaning, or provide an interpretation.

var - Declaration of Variables

Syntax: var

- Variable is set to 0.
Syntax: var variable_name = 0;
- Variable receives the value 5.0.
Syntax: var variable_name = a + 5.0;
- Variable receives the result of expression.

If - Flow Controller

When we program, we often need a particular block of code to be executed only if a certain condition is true. In such cases, we must make use of a condition structure.

Example:

Syntax: if condition

- If the expression is TRUE execute the instructions until the end of the flow control;
- The end of flow control is represented by end if;

while - Tie

The while repeats an instruction code while the condition set in its header is true. The while structure may be the simplest to understand at this point, however, not infrequently we find students who say they do not understand the functioning of this structure. If you have understood the if structure, consider the while structure as the if statement, but instead of executing your statement code once, it will execute as long as the defined expression equals true.

Example:

Syntax: while condition

- If it is TRUE execute the instructions until the end of the loop or until some unconditional bypass command;
- The 'break' command is an unconditional deviation. It breaks the loop that is running.
- With the end of the loop, we represent with the command end while.

get - Input Command

syntax: get variable_name

- get receives a value that has the origin of the keyboard. A variable with this name will be created and assigned a value. Ex: it's like a "print" or "println".

show - Output command

Show always skips a line at the end of its run.

syntax: show 'string'

- Prints "string" on the screen.
syntax: show 'string'> 'string2'
- You can concatenate strings by using the '>' command to print on the screen. The same holds true for mathematical variables or expressions.

EXAMPLES OF IMPLEMENTATION OF RAFIS CHUCHU.

I) The following code prints the average between two numbers

```
.
    get x
    get y
    show x + y / 2
```

II) The following code prints "is prime" if a number greater than or equal to 2 is prime. If not, it prints "not cousin";

```
    get x
    var y = 2
    var bool
    while y < x
    - - - - - if x%y & 0
    - - - - - show ' não é primo'
    - - - - - var bool = 1
    - - - - - break
    - - - - - end if
    - - - - - var y = y + 1
    end while
    if bool & 0
    - - - - - show 'e primo'
    end if
```

III) The code prints the table from 1 to 10;

```
    var x
    var y = 1
    while y < 11
    - - - - var x=1
    - - - - show 'tabuada do ' > y
    - - - - show ''
    - - - - while x < 11
    - - - - - show x > ' * ' > y > ' = ' > x*y
    - - - - - var x = x + 1
    - - - - end while
    - - - - var y = y + 1
    - - - - show ''
    end while
```

IV) Printing on the screen or whole numbers, as well as floating numbers.

```
show 'Hello world Rafis Chuchu'
show ' Vamos pro Tomorrowland belgium parça !'
show 1 + 1 // 2
```

V) Application with operators:

```
var a = 1 + 1  
var b = 2 * 2  
var c = 2 + 2 * 2
```

Ex = $2 + 2 * 2 = 8$

```
show a  
show b  
show c
```

IV) If you want to comment the code, always use the "#". This method works only for each line you want to comment on.

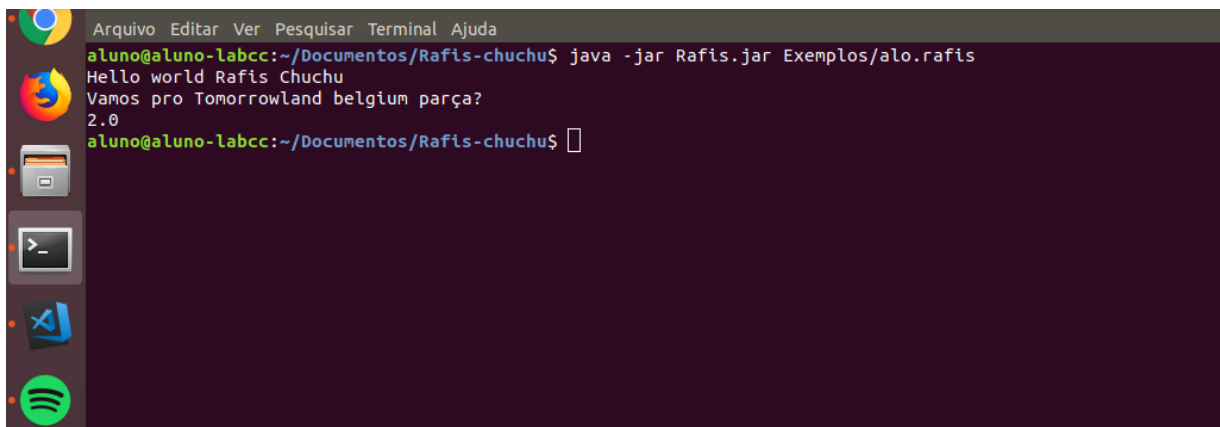
```
# show 'Hellow rafis'
```

Execution of the Rafis Chuchu language

To run the code, type in the command line:

```
java -jar Rafis.jar .;./ directory/of/archive/file_name.rafis
```

Examples of how to run the program.



The screenshot shows a terminal window with a dark background. The title bar at the top reads "Arquivo Editar Ver Pesquisar Terminal Ajuda". The prompt is "aluno@aluno-labcc:~/Documentos/Rafis-chuchu\$". The command entered is "java -jar Rafis.jar Exemplos/alo.rafis". The output of the program is displayed in three lines: "Hello world Rafis Chuchu", "Vamos pro Tomorrowland belgium parça?", and "2.0". The prompt returns to "aluno@aluno-labcc:~/Documentos/Rafis-chuchu\$". On the left side of the terminal, there is a vertical dock with icons for various applications, including a web browser, a file manager, a terminal, and a music player.