

fobi

Mathematical Calculation & Graphical Language

Patrick Afrifah (pa2487)

1. Introduction

fobi is a high-level, general purpose, and dynamic programming language which has a subset of Python grammar. It utilizes LLVM as the backend to develop bytecode.

2. Language Description

fobi allows users to use fewer lines of code to implement and manipulate arrays and matrices. This programming language can also be used to implement one variable linear regression models.

2.1. Comments

fobi uses C /C++ and Java style comments for single and multiple line comments. Example:

```
// This is a single line comment
```

```
/*  
    This is a multiple line comment  
*/
```

2.2. Language Keywords

Keyword	Description	Example
fnc	Define a function	fnc myfunction() :
while	while loop	while X: pass
as	Part of the <i>with-as</i> statement	with X as Y
for	loop over a collection of things	for X in Y: pass
elif	Else if condition	if: elif: Y; else: W
else	Else condition	if: elif: Y; else: W
in	Part of for-loops. Also a test of X in Y	
try	Try this block, and if exception, go to except	try: pass
except	If an exception happens, do this	except Error, e: print e
global	Declare that you want a global variable	global X
pass	This block is empty	fnc empty(): pass
return	Exit the function with a return value	fnc X(): return Y
with	With an expression as a variable do	with X as Y: pass
continue	Don't process more of the loop , do it again	while True: continue
break	Stop this loop right now	while True: break
plot	Creates a line plot	plot(x, y)
plot3	Creates a 3D plot	plot(x, y, z)
scatter	Creates a scatter plot	scatter(x, y)
len	Length of an array, string, matrix, or list	len('hello') == 5
not	Logical not	not True == False
or	Logical or	True or False == True
include	Include/ import libraries	
from	Importing specific parts of a library	
print	Print this string	print "this string"

and	Logical and	True and False == False
open	Open and read file	open('myfile.txt')

2.3. Data Types

Type	Description	Example
True	True boolean value	True or False == True
False	False Boolean value	False and True == False
strings	Stores textual information	x = 'fobi'
numbers	Stores integers	I = 9
floats	Stores decimals	1 = 4.545
map	Stores a key = value mapping of things	m = {'w':7, 'v': 15}

2.4. Operators

Operator	Description	Example
+	Addition / concatenation	2 + 3 == 5 / "hello" + " world" == hello world
-	Subtraction	6 - 3 == 3
*	Multiplication	3 * 4 == 12
<	Less than	2 < 8 == True
<=	Less or equal to	2 <= 8 == True
>	Greater than	3 > 6 == False
>=	Greater than or equal to	3 >= 6 == False
==	Equal	6 == 7 == False
!=	Not equal	6 != 7 == True
-=	Subtract and assign	x = 1; x -= 2
+=	Add and assign	x = 1; x += 2
%	Modulus	5 % 2 == 1
%=	Modulus and assign	x = 1; x %= 2
^	Power	3 ^ 2 == 9
^=	Power assign	x = 1; x ^= 2

2.5. String Escape

Escape	Description
\\	Backslash
\'	Single-quote
\"	Double-quote
\a	Bell
\b	Backspace
\f	Formfeed
\n	Newline
\r	Carriage
\t	Tab
\v	Vertical tab

2.6. Function Definition

Functions are defined with the keyword *func*. Example of a user defined function;

```
fnc myFunction(msg){
    // This is function returns a string
    return ("The return value of myFunction() is " + msg)
}
```

3. Source Code

```
// Flow control
```

```
// Define an array
```

```
r = array([1,2,3]) // ans: r = [1 2 3]
```

```
s = 2*r // ans: r = [2 4 6]
```

```
// Define a matrix
```

```
M = matrix([1 2 3; 4 5 8; 9 9 9]) // ans: M =  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 8 \\ 9 & 9 & 9 \end{bmatrix}$ 
```

```
b = x*M // ans: b = 27 33 42
```

```
N = matrix([r; s]) // ans: N =  $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$ 
```

```
// Plot example
```

```
x = array([-10:0.1:10])
```

```
y = sin(x)
```

```
plot(x,y)
```

```
// Plot x by y
```

