



Modelos de Linguagem de Programação

Unidade 8 – Programação Funcional com a Linguagem LISP



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com






Bibliografia

<http://www.newlisp.org/index.cgi?Documentation>

newlisp.org/index.cgi?Documentation

Drive GDrive R TP T g Ms Prime aws Oracle ibm Foto libgen FStack Y F ML

puts the fun  back in Lisp

Home · About · Downloads · Docs · Modules · Links · Tips&Tricks ·

Documentation

Main documentation and man pages

- newLISP Manual and Reference in [HTML](#) format, no frames.
- newLISP Manual and Reference in [HTML](#) format, with index frame.





Downloading e Instalação

- ✓ O download do **newLISP** pode ser feito em <http://www.newlisp.org/>
- ✓ Disponível em diversas plataformas;
- ✓ O interpretador poderá ser chamado diretamente pela linha de comandos, em uma console;
- ✓ O código fonte pode ser gerado a partir de um editor de programas, como por exemplo o **Notepad++**.





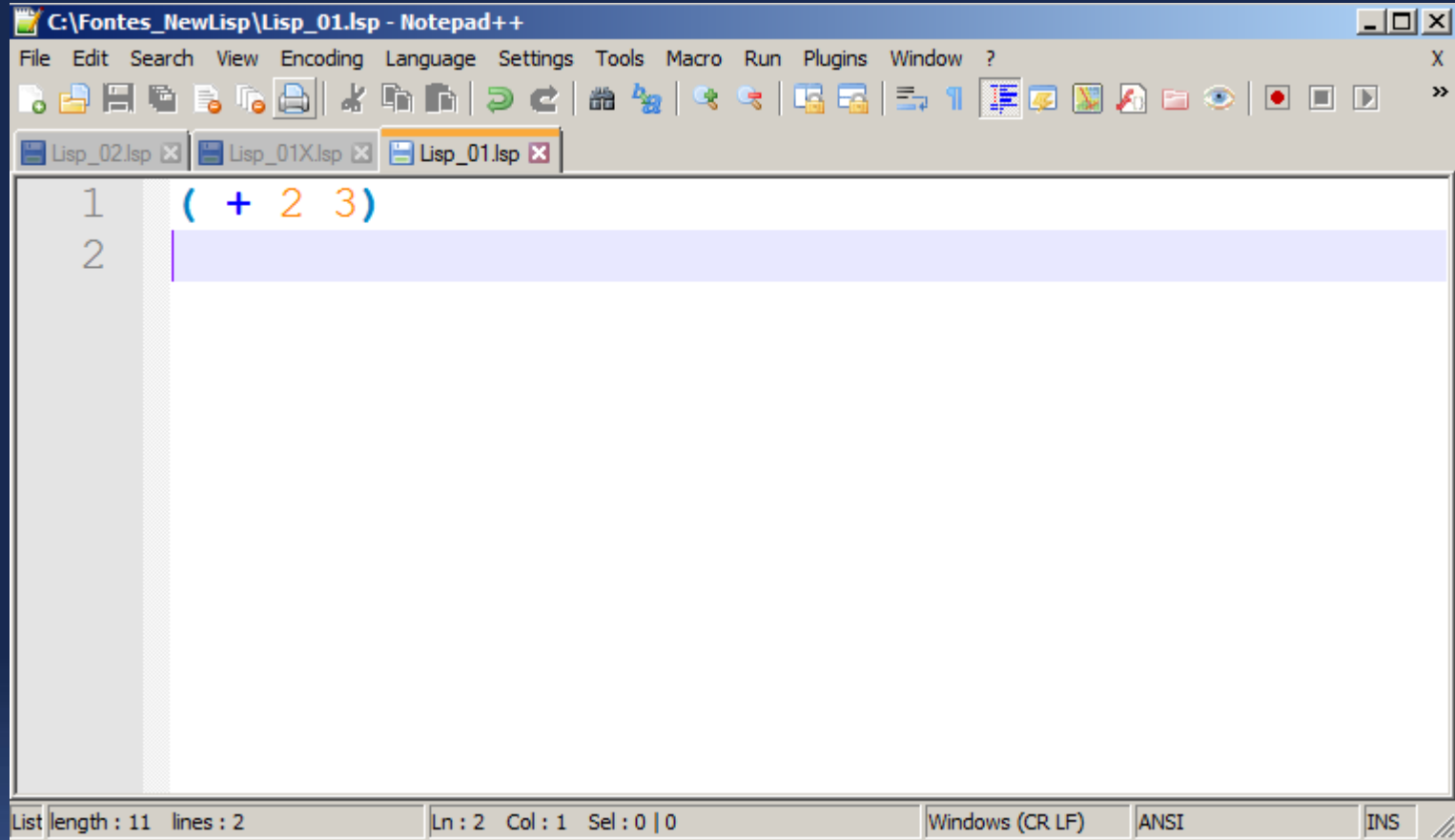
Introdução

```
Command Prompt - newlisp
C:\>newlisp
newLISP v.10.7.5 64-bit on windows IPv4/6 UTF-8 libffi, options: newlisp -h
>
```





Exemplo – Lisp_O1.lsp



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_O1.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations, editing, and development. The tab bar shows three open files: "Lisp_O2.lsp", "Lisp_O1X.lsp", and "Lisp_O1.lsp". The main text area contains two lines of code:

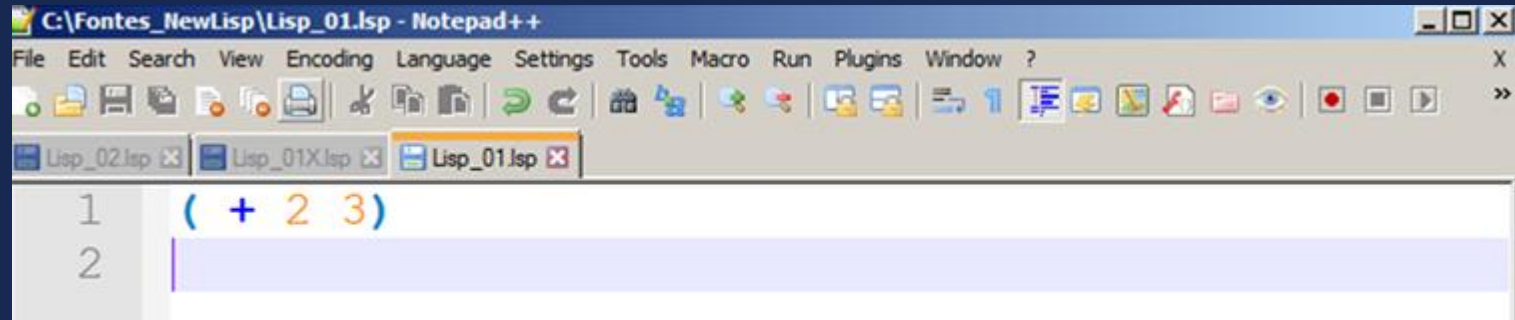
```
1  ( + 2 3 )  
2
```

The status bar at the bottom displays "Ln : 2 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "ANSI", and "INS".

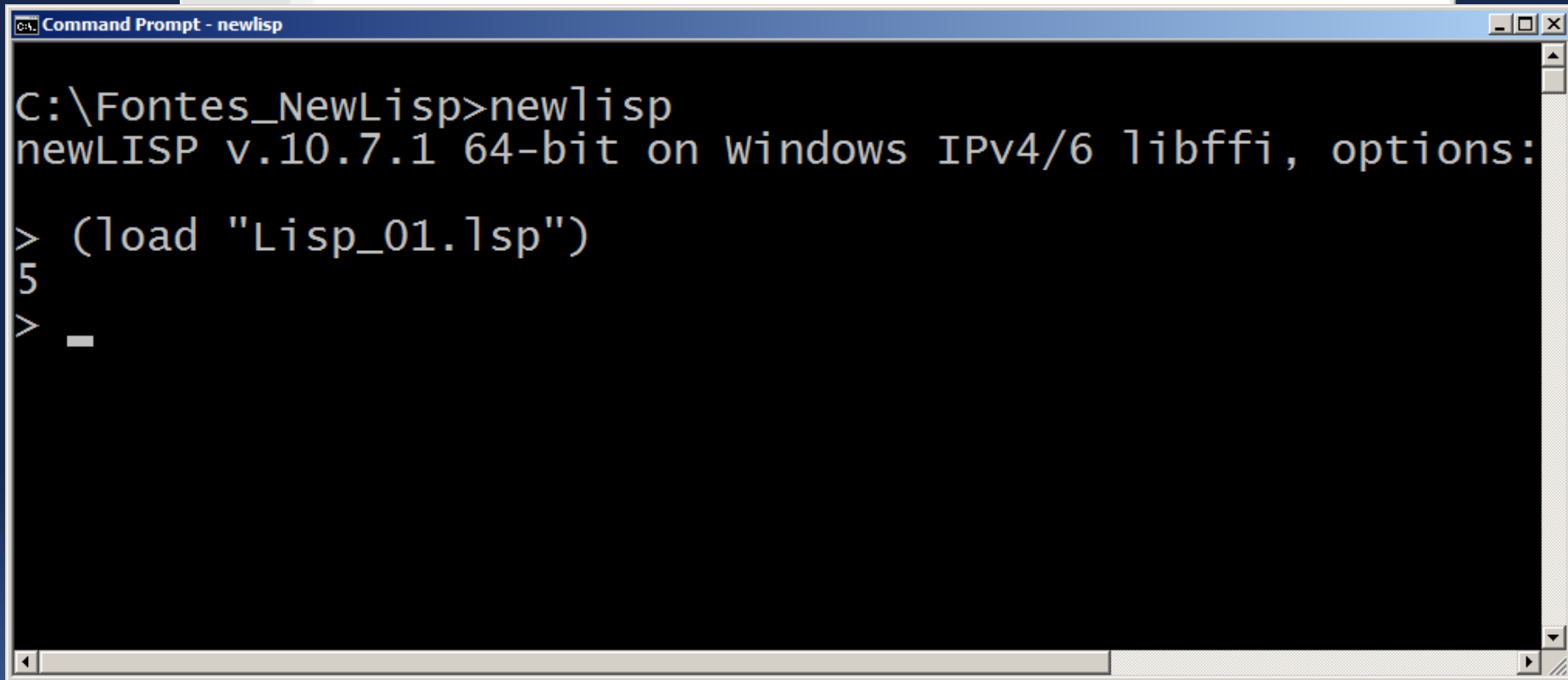




Exemplo – Lisp_01.lsp



```
C:\Fontes_NewLisp\Lisp_01.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_02.lsp Lisp_01X.lsp Lisp_01.lsp
1 (+ 2 3)
2
```

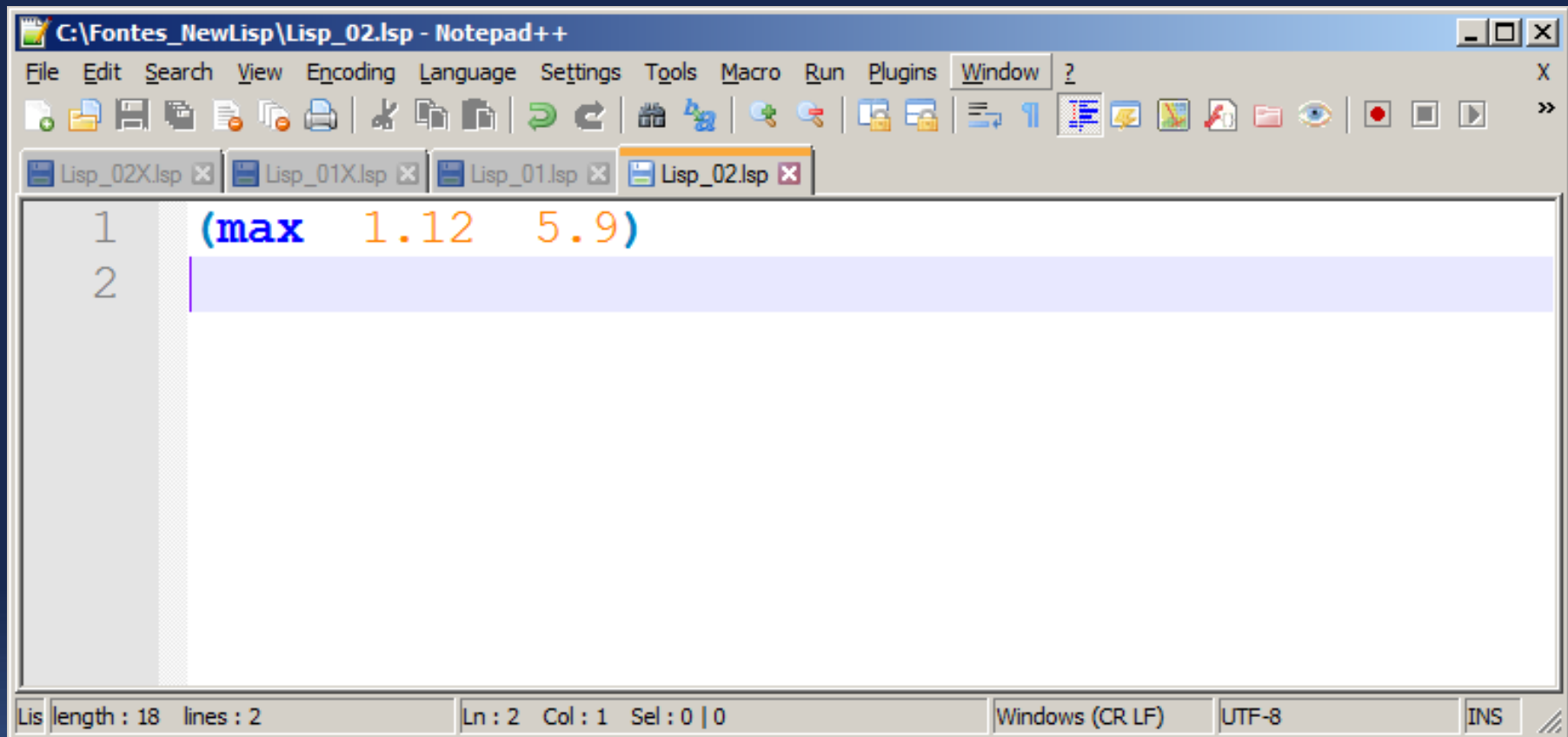


```
Command Prompt - newlisp
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on windows IPv4/6 libffi, options:
> (load "Lisp_01.lsp")
5
> _
```





Exemplo – Lisp_02.lsp



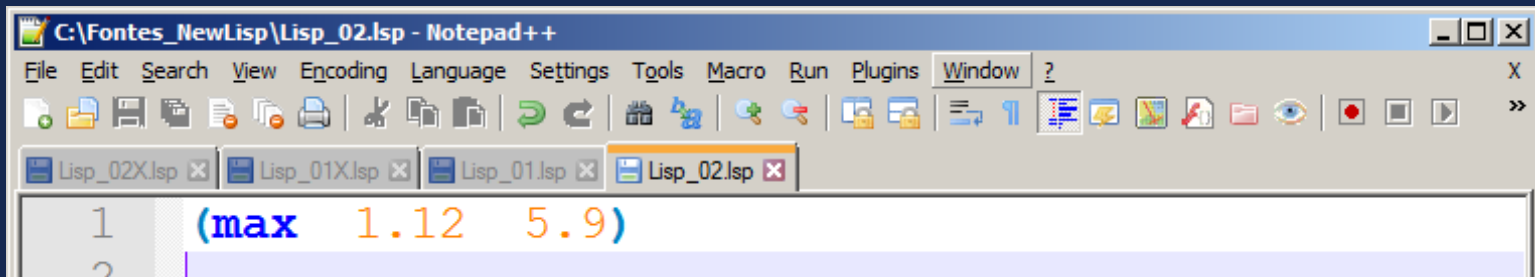
The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_02.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and development. The tab bar shows four open files: Lisp_02X.lsp, Lisp_01X.lsp, Lisp_01.lsp, and Lisp_02.lsp. The main text area contains two lines of code: line 1 is `(max 1.12 5.9)` and line 2 is empty. The status bar at the bottom displays "Ln : 2 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".

```
1 (max 1.12 5.9)
2
```

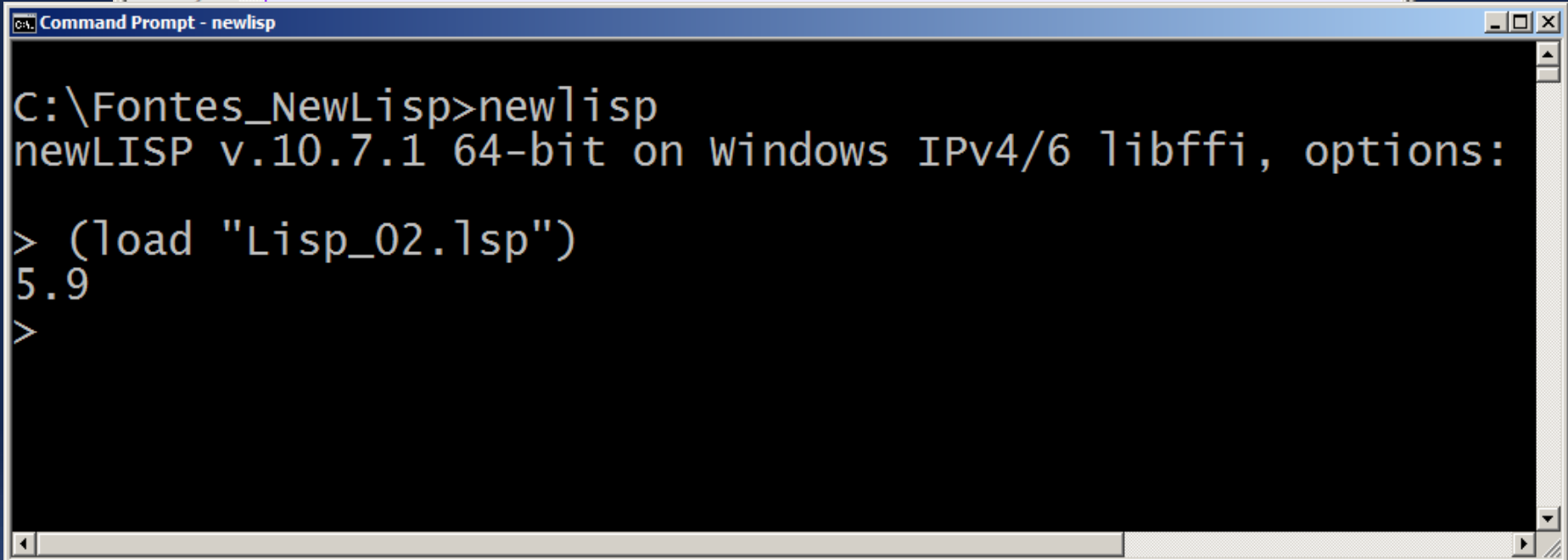




Exemplo – Lisp_02.lsp



```
C:\Fontes_NewLisp\Lisp_02.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_02X.lsp Lisp_01X.lsp Lisp_01.lsp Lisp_02.lsp
1 (max 1.12 5.9)
2
```



```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on windows IPv4/6 libffi, options:
> (load "Lisp_02.lsp")
5.9
>
```





As três regras básicas de newLISP

Regra 1

- ✓ Uma **lista** é uma sequência de elementos delimitados entre parênteses;
- ✓ **Lista** é a estrutura básica do newLISP;
- ✓ Exemplos:

```
NGOANDSCS.NOTAS x settings.xml x pom.xml x server.xml x Tarefa_T8_Solucao_SCRIPT.sql x Tarefa_T9_Solucao_SCRIPT.sql x Tarefa_T10_Solucao_SCRIPT.sql x synced.rkt x blueboxes.rkt x NewLISP_02.lsp x
```

(1 2 3 4 5)	;uma lista de inteiros
("o" "gato" "triste")	;uma lista de strings
(x y z a)	;uma lista de nomes simbólicos
(sin cos tan)	;uma lista de funções newlisp
(1 2 "teste" x sin)	;uma lista mista
(1 2 (1 2 3) 3 4)	;uma lista com uma lista dentro dela
((1 2) (3 4) (5 6))	;uma lista de listas





As três regras básicas de newLISP

Regra 2

- ✓ Para **newLISP**, o **primeiro elemento** da lista tem um significado especial;
- ✓ Quando **newLISP** recebe uma lista, ele trata sempre o primeiro elemento como uma função e tenta utilizar os demais elementos como argumentos;
- ✓ **(+ 5 4)** é uma lista com três elementos. Uma função chamada +, seguida por 2 valores numéricos;
- ✓ Quando **newLISP** recebe a lista acima, ele a avalia e retorna o valor **9**;





Exemplo – Lisp_03.lsp

The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_03.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and development. The editor has a tab for "Lisp_03.lsp". The code is as follows:

```
1  (+ 1 2 3 4 5 6 7 8 9)
2
```

The status bar at the bottom displays: "Lisp length : 24 lines : 2", "Ln : 2 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".





Exemplo – Lisp_O2A.lsp

ENG0ANDSCS.NOTAS x settings.xml x pom.xml x server.xml x Tarefa_T8_Solucao_SCRIPT.sql x Tarefa_T9_Solucao_SCRIPT.sql x

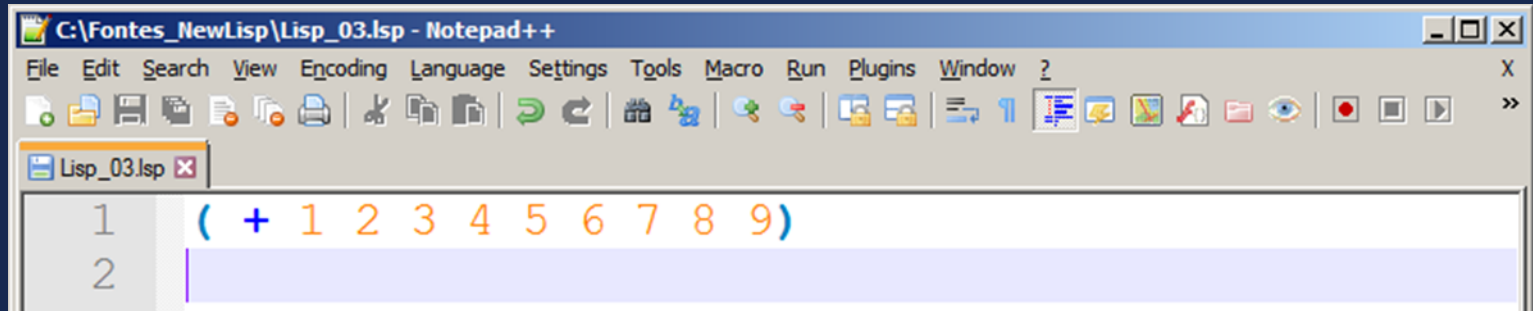
```
(max 1 4.5 9.88 12 999 45.000 65 )
```

```
Command Prompt - newlisp "Lisp_2A.lsp"
>
>
>
>
>
>
> (load "newlisp_02A.lsp")
999
>
```





Exemplo – Lisp_03.lsp

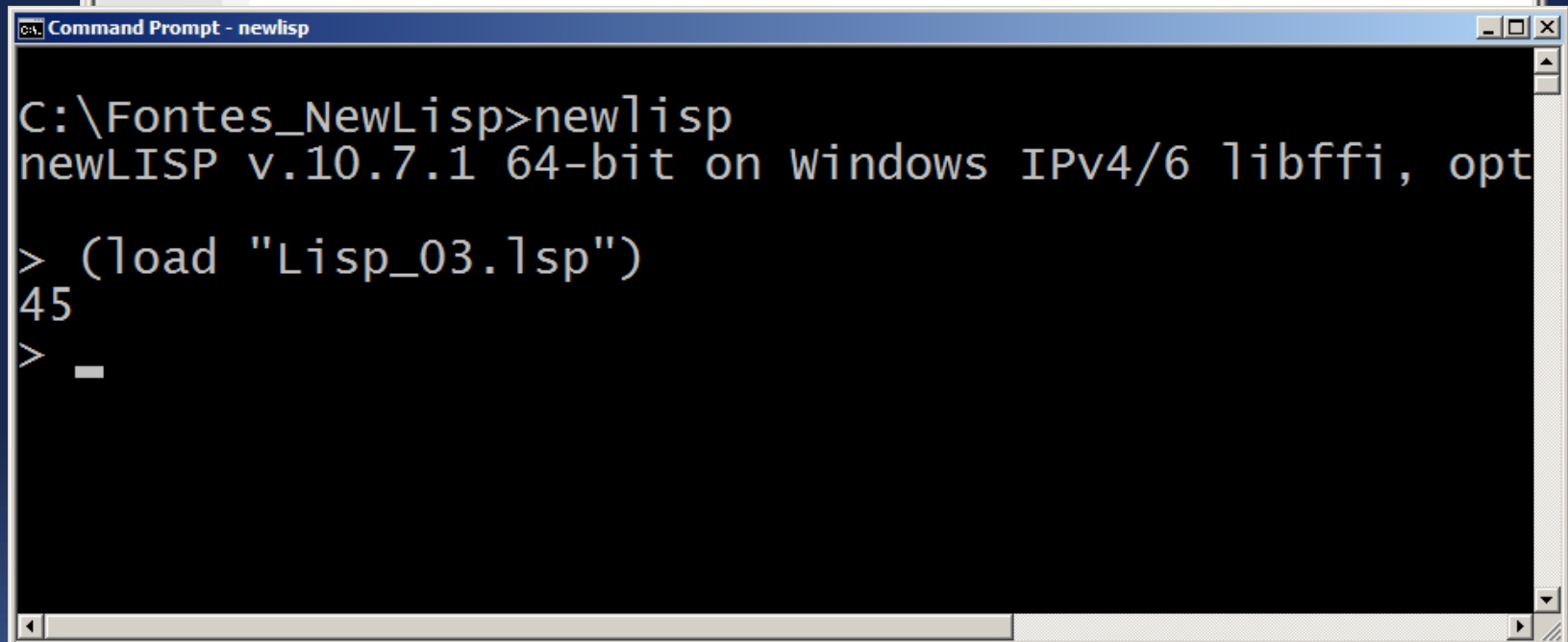


C:\Fontes_NewLisp\Lisp_03.lsp - Notepad++

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
```

Lisp_03.lsp

```
1  ( + 1 2 3 4 5 6 7 8 9 )
2
```



Command Prompt - newlisp

```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt
> (load "Lisp_03.lsp")
45
> _
```





Exemplo – Lisp_O4.lsp

```

1 (directory "/" )
2

```

Notepad++ window title: *C:\Fontes_NewLisp\Lisp_O4.lsp - Notepad++

Menu: File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Toolbar: [Icons for file operations, editing, and development]

Tab: Lisp_O4.lsp

Status bar: Lis length : 18 lines : 2 | Ln : 2 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS





Exemplo – Lisp_O4.lsp

)

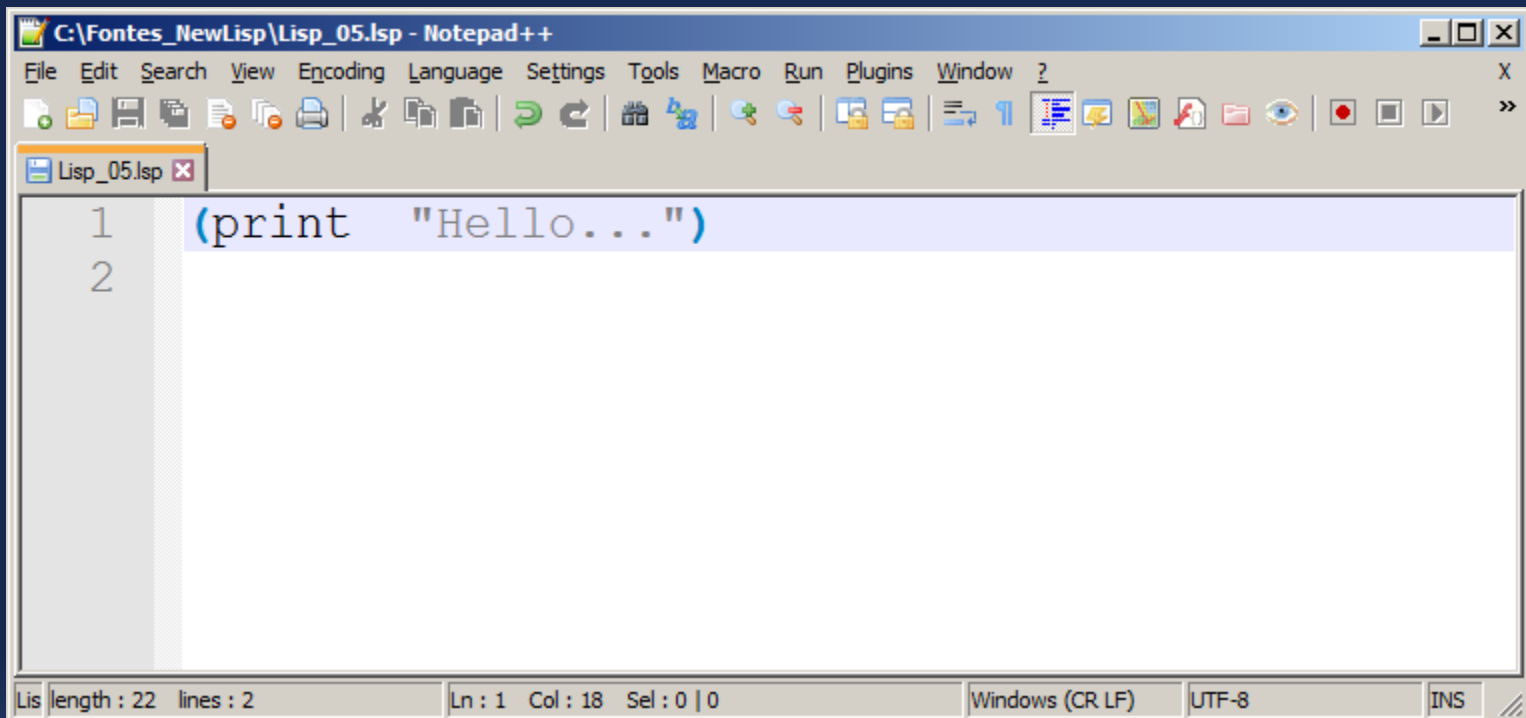
```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt

> (load "Lisp_O4.lsp")
("$Recycle.Bin" "Documents and Settings" "Fontes_NewL
cao"
"MSOCache" "NVIDIA" "pagefile.sys" "PerfLogs" "Progr
86)"
"ProgramData" "Recovery" "Softwares" "System Volume
t"
"Windows")
>
```





Exemplo – Lisp_O5.lsp



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_O5.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations and editing. The editor area shows a single line of Lisp code: `(print "Hello...")`. The status bar at the bottom displays "Lis | length : 22 | lines : 2", "Ln : 1 | Col : 18 | Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".

```
1 (print "Hello...")
2
```





Exemplo – Lisp_O5.lsp

The image shows two overlapping windows. The top window is Notepad++ with the file 'C:\Fontes_NewLisp\Lisp_O5.lsp'. The code in the editor is: `1 (print "Hello...")`. The bottom window is a Command Prompt titled 'Command Prompt - newlisp'. It shows the following interaction:
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt
> (load "Lisp_O5.lsp")
Hello..."Hello..."
> _

Observação: Além de imprimir o **string**, a função também o retorna.





Exemplo – Leitura de Arquivo

(`read-file` "Arquivo.txt")

Ouviram do Ipiranga as margens plácidas
De um povo heroico o brado retumbante
E o sol da liberdade, em raios fúlgidos
Brilhou no céu da pátria nesse instante

Se o penhor dessa igualdade
Conseguimos conquistar com braço forte
Em teu seio, ó liberdade

```
Command Prompt - newlisp
E:\NewLisp>newlisp
newLISP v.10.7.5 64-bit on windows IPV4/6 UTF-8 libffi, optio
> (read-file "Arquivo.txt")
"Ouviram do Ipiranga as margens plácidas\r\nDe um povo heroi
te\r\nE o sol da liberdade, em raios fúlgidos\r\nBrilhou no
se instante\r\n\r\nSe o penhor dessa igualdade\r\nConseguimos
o forte\r\nEm teu seio, ó liberdade\r\nDesafia o nosso pei
!\r\n\r\nô Pátria amada\r\nIdolatrada\r\nSalve! Salve!\r\n\
intenso, um raio vívido\r\nDe amor e de esperanãa á terra
formoso céu, risonho e límpido\r\nA imagem do Cruzeiro res
ante pela própria natureza\r\nés belo, s forte, impívido
```





Exemplo – Leitura de Arquivo



```
( println ( read-file "Arquivo.txt") )
```

Ouviram do Ipiranga as margens plácidas
De um povo heroico o brado retumbante
E o sol da liberdade, em raios fúlgidos
Brilhou no céu da pátria nesse instante

Se o penhor dessa igualdade
Conseguimos conquistar com braço forte
Em teu seio, ó liberdade

Desafia o nosso peito a própria morte!

```
Command Prompt - newlisp

>
>
(println (read-file "Arquivo.txt"))

Ouviram do Ipiranga as margens plácidas
De um povo heroico o brado retumbante
E o sol da liberdade, em raios fúlgidos
Brilhou no céu da pátria nesse instante

se o penhor dessa igualdade
Conseguimos conquistar com braço forte
```





Funções newLISP

- ✓ Há cerca de **380** funções **newLISP**;
- ✓ Toda função **newLISP** **retorna** um **valor**;
- ✓ Mesmo um **print** retorna um valor;
- ✓ Diz-se que a ação de imprimir um valor é um "**side-effect**";
- ✓ Assim, quando se executa a função **print** serão vistos 2 valores: um que é o valor impresso e outro o valor retornado.





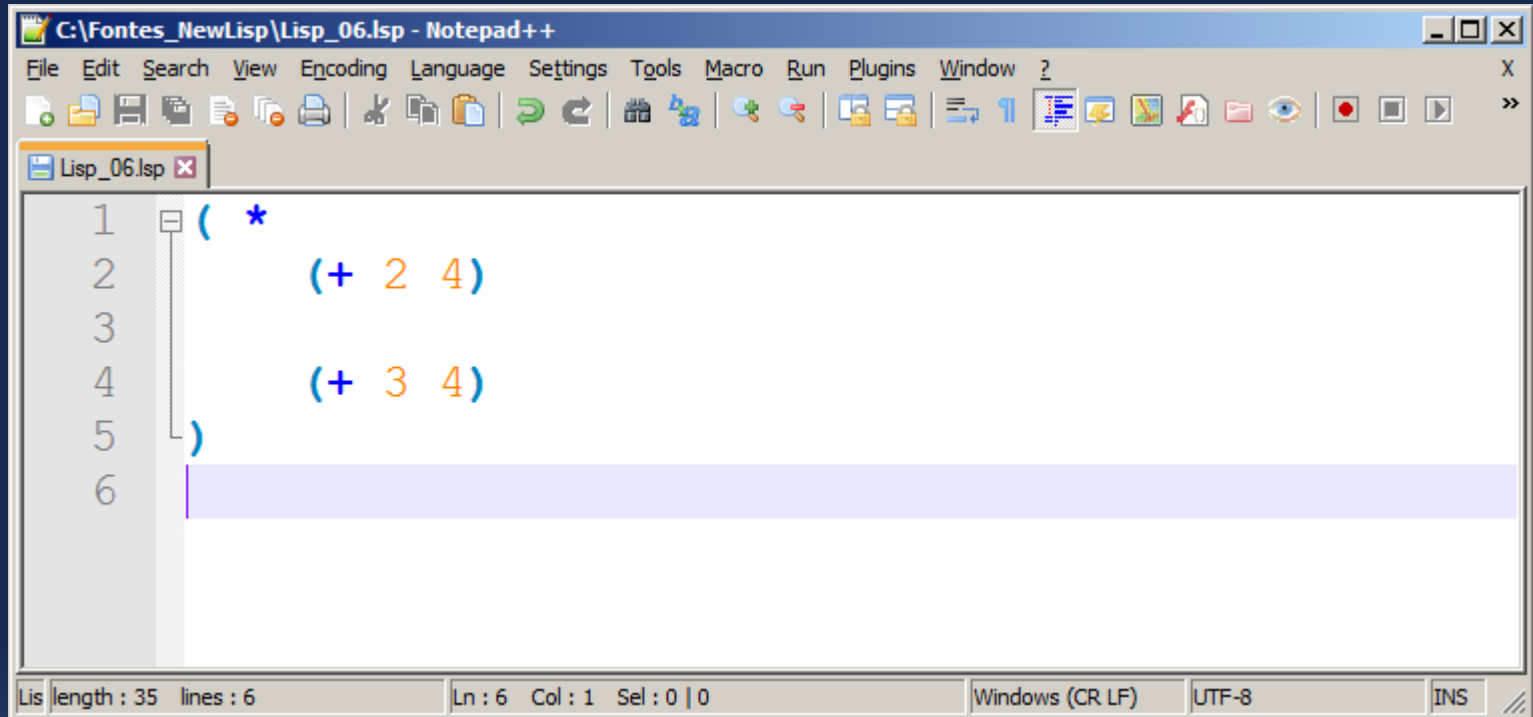
Listas Aninhadas

- ✓ Listas podem conter elementos que também são listas;
- ✓ Exemplo: $(* (+ 2 4) (+ 3 4))$
- ✓ `newLISP` avalia inicialmente a função $(+ 2 4)$ o qual retorna **6**;
- ✓ Em seguida avalia a função $(+ 3 4)$ o qual retorna **7**;
- ✓ Finalmente a expressão $(* 6 7)$ é avaliada, retornando o valor **42**.





Exemplo – Lisp_O6.lsp



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_O6.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and development. The editor displays the following Lisp code:

```
1  ( *  
2      (+ 2 4)  
3  
4      (+ 3 4)  
5  )  
6
```

The status bar at the bottom shows "Lis length : 35 lines : 6", "Ln : 6 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".





Exemplo – Lisp_06.lsp

```
C:\Fontes_NewLisp\Lisp_06.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_06.lsp
1 ( *
2   (+ 2 4)
3
4   (+ 3 4)
5 )
```

```
Command Prompt - newlisp
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt
> (load "Lisp_06.lsp")
42
> _
```





As três regras básicas de newLISP

Regra 3



- ✓ **Apóstrofe** inibe a avaliação da expressão;
- ✓ Assim, **(+ 2 3)** será avaliada pelo interpretador **newLISP** retornando o valor 5;
- ✓ A expressão **'(+ 2 3)** é precedida por **'** o qual previne a avaliação. Nesse caso, a expressão será retornada sem processar a avaliação;
- ✓ Esse procedimento permite que a **lista** seja **tratada** como **dado** e não como chamada de **função** para ser avaliada;
- ✓ Exemplo: **(31 11 2017)** representa uma data e portanto deve ser escrita **'(31 11 2017)**.





Exemplo – Lisp_07.lsp

The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_07.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and development. The tab bar shows two tabs: "Lisp_06.lsp" and "Lisp_07.lsp", with "Lisp_07.lsp" being the active tab. The editor area contains two lines of code: line 1 is `' (10 11 2017)` and line 2 is empty. The status bar at the bottom displays "Ln : 2 Col : 1 Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".

```
1 ' ( 10 11 2017 )
2
```





Exemplo – Lisp_07.lsp

The image shows two windows. The top window is Notepad++ editing C:\Fontes_NewLisp\Lisp_07.lsp. It contains two lines of code: line 1 is `' (10 11 2017)` and line 2 is empty. The bottom window is a Command Prompt titled 'newlisp'. It shows the command `newlisp` being executed, followed by the version information `newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt`. Then, the command `> (load "Lisp_07.lsp")` is entered, and the output `(10 11 2017)` is displayed. A prompt `>` and a cursor are shown on the next line.

```
C:\Fontes_NewLisp\Lisp_07.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_06.lsp x Lisp_07.lsp x
1 ' ( 10 11 2017 )
2

C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, opt
> (load "Lisp_07.lsp")
(10 11 2017)
> _
```





Exemplo – Lisp_08.lsp

The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_08.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The tab bar shows two open files: "Lisp_06.lsp" and "Lisp_08.lsp". The main text area contains three lines of code, with the first line selected:

```
1 '( "Aparecido V. de Freitas")
2
3
```

The status bar at the bottom displays: "Lis | length : 34 | lines : 3 | Ln : 2 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS |".





Exemplo – Lisp_08.lsp

The screenshot shows two windows. The top window is Notepad++ editing C:\Fontes_NewLisp\Lisp_08.lsp. It contains the following code:

```
1 ' ( "Aparecido V. de Freitas")
2
3
```

The bottom window is a Command Prompt titled "newlisp". It shows the following commands and output:

```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on windows IPv4/6 libffi, opt

> (load "Lisp_08.lsp")
("Aparecido V. de Freitas")
> _
```





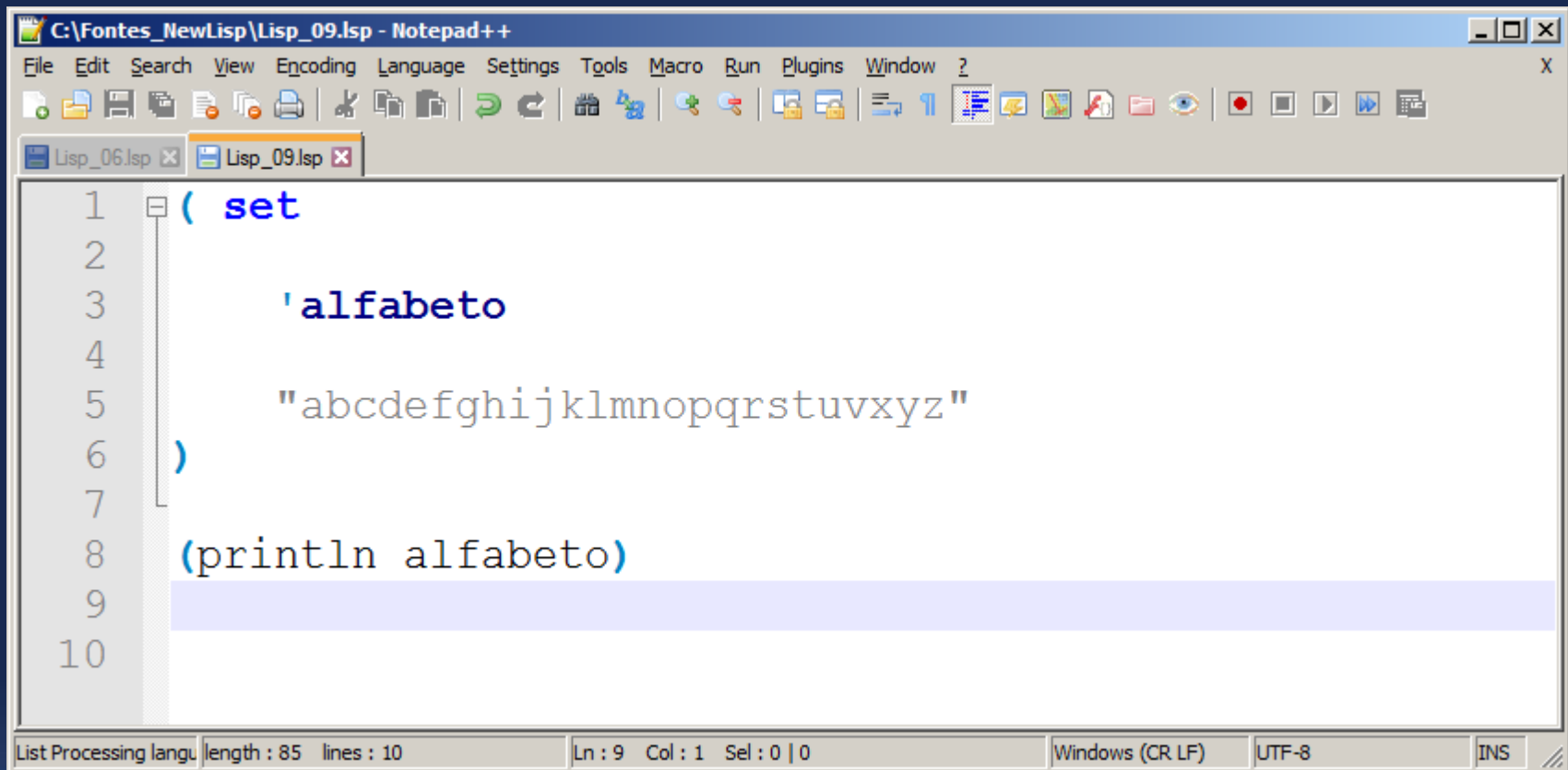
Símbolos

- ✓ Em **newLISP**, um **símbolo** representa algo associado a um **nome**;
- ✓ Pode-se definir esse **nome** representando alguma coisa no código e, posteriormente, usar-se o nome ao invés de seu conteúdo;
- ✓ Por exemplo: (**set** 'alfabeto "abcdefghijklmnopqrstuvxyz")
- ✓ No exemplo, definiu-se por meio da função **set** um nome chamado 'alfabeto cujo valor é o string que contém as 26 letras do alfabeto;
- ✓ A partir dessa definição, o símbolo **alfabeto** pode ser usado em qualquer ponto do programa e será avaliado sempre que for chamado ;





Exemplo – Lisp_09.lsp



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_09.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and running. The tab bar shows "Lisp_06.lsp" and "Lisp_09.lsp". The editor content is as follows:

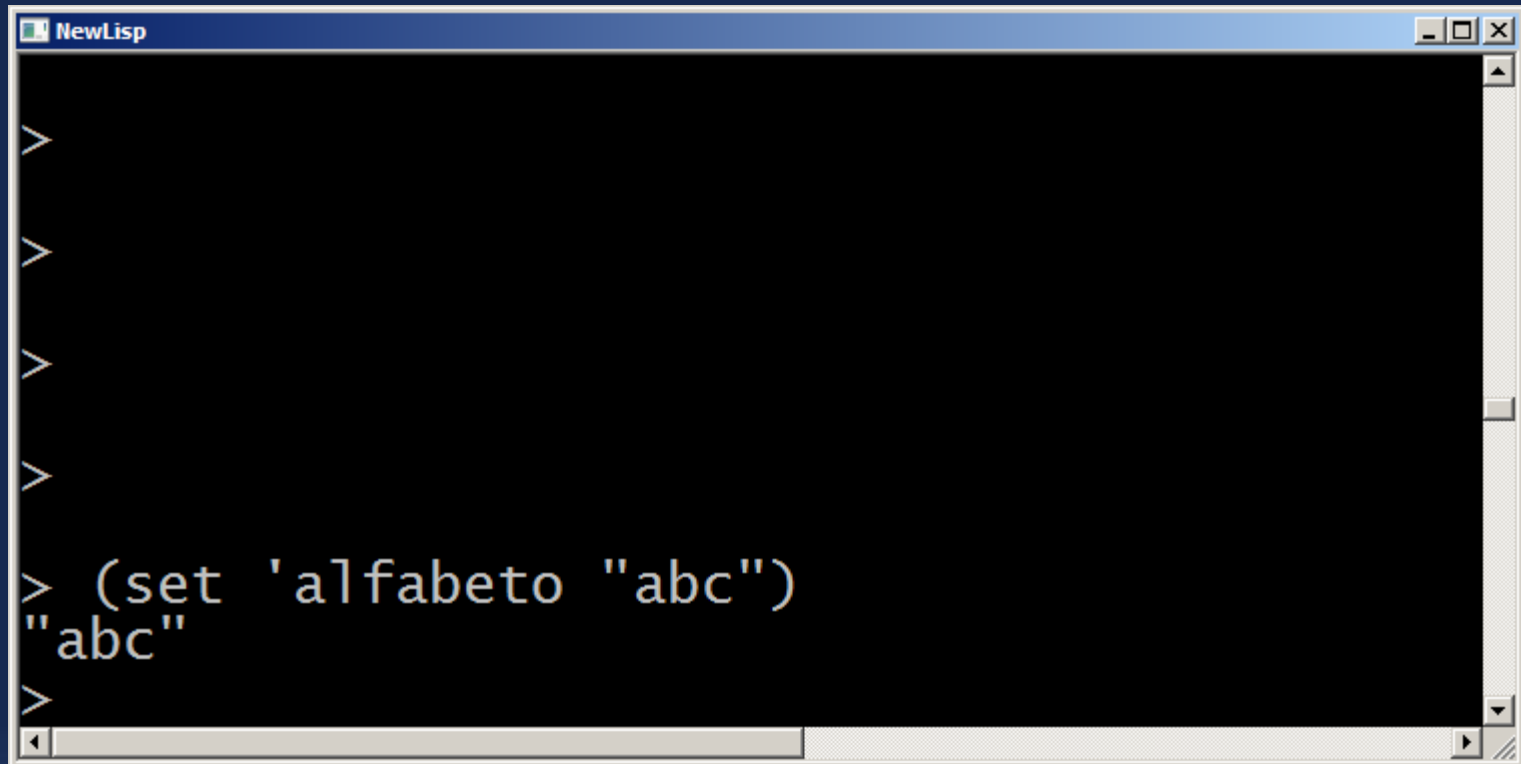
```
1  ( set
2
3      'alfabeto
4
5      "abcdefghijklmnopqrstuvxyz"
6  )
7
8  (println alfabeto)
9
10
```

The status bar at the bottom displays: "List Processing langu", "length : 85", "lines : 10", "Ln : 9", "Col : 1", "Sel : 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".





Exemplo – Lisp_10.lsp

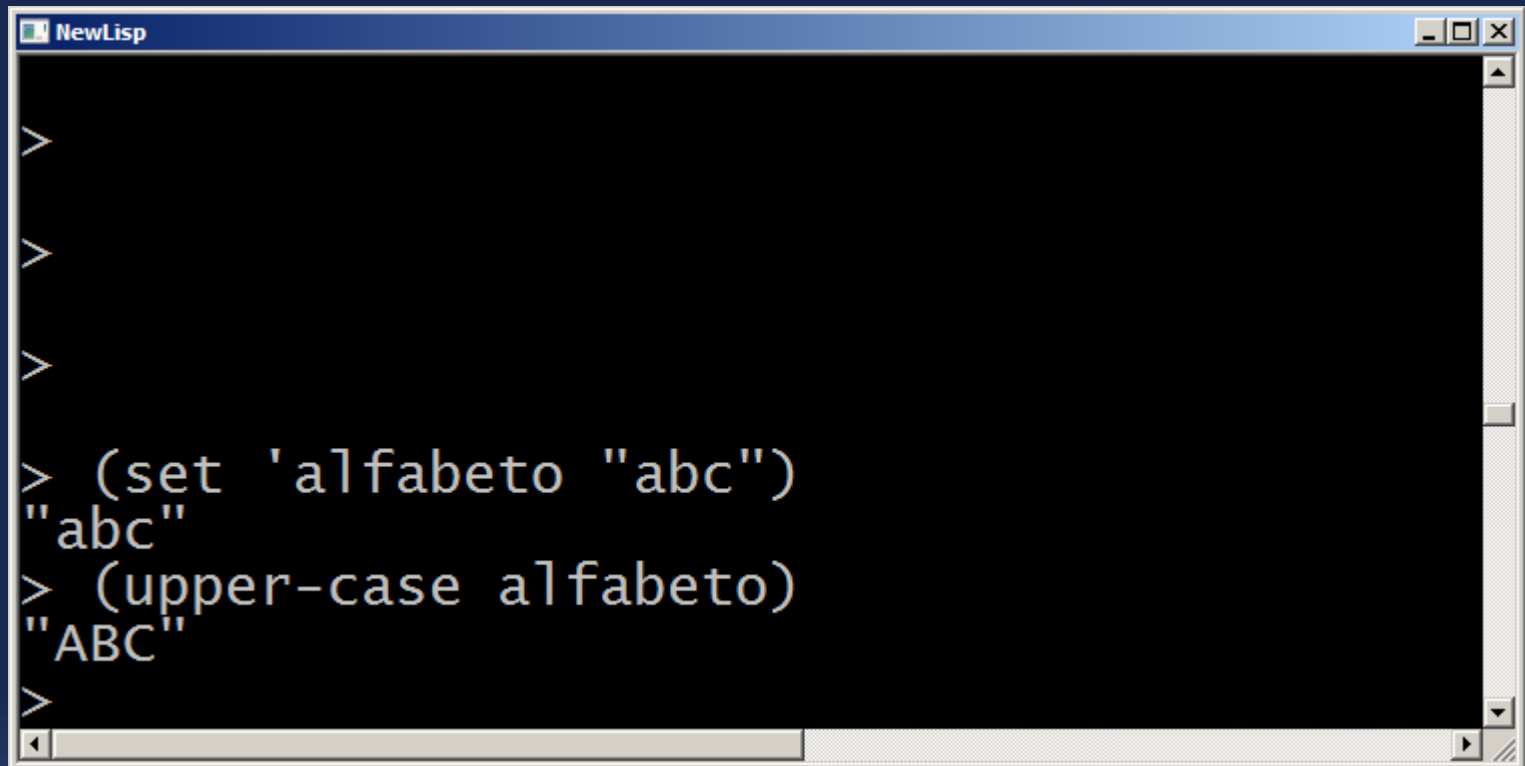
A screenshot of a NewLisp window. The window has a title bar that says "NewLisp" and standard window controls (minimize, maximize, close). The main area is black with white text. On the left side, there are four prompt characters ">". The first three are followed by a space. The fourth prompt is followed by the code "(set 'alfabeto \"abc\")". Below this code, the string "\"abc\"" is displayed, indicating the result of the set operation. A horizontal scrollbar is visible at the bottom of the text area.

```
>  
>  
>  
>  
> (set 'alfabeto "abc")  
"abc"
```





Exemplo – Lisp_10.lsp

A screenshot of a NewLisp window. The window has a title bar that says "NewLisp" and standard window controls (minimize, maximize, close). The main area is black with white text. It shows a series of prompt characters ">" followed by Lisp code. The code defines a variable 'alfabeto' with the value "abc" and then uses the 'upper-case' function to convert it to "ABC".

```
>  
>  
>  
> (set 'alfabeto "abc")  
"abc"  
> (upper-case alfabeto)  
"ABC"  
>
```





Exemplo – Lisp_10.lsp



```
NewLisp
>
>
> (set 'alfabeto "abc")
"abc"
> (upper-case alfabeto)
"ABC"
> (first (upper-case alfabeto) )
"A"
>
```





Exemplo – Lisp_11.lsp



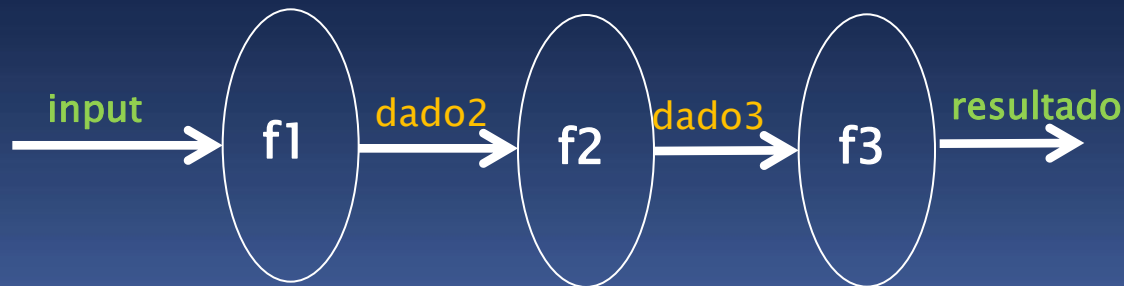
```
NewLisp
>
>
> (set 'alfabeto "abc")
"abc"
> (upper-case alfabeto)
"ABC"
> (first (upper-case alfabeto) )
"A"
> (println (first (upper-case alfabeto)))
A
"A"
>
```





Símbolos – Mais observações

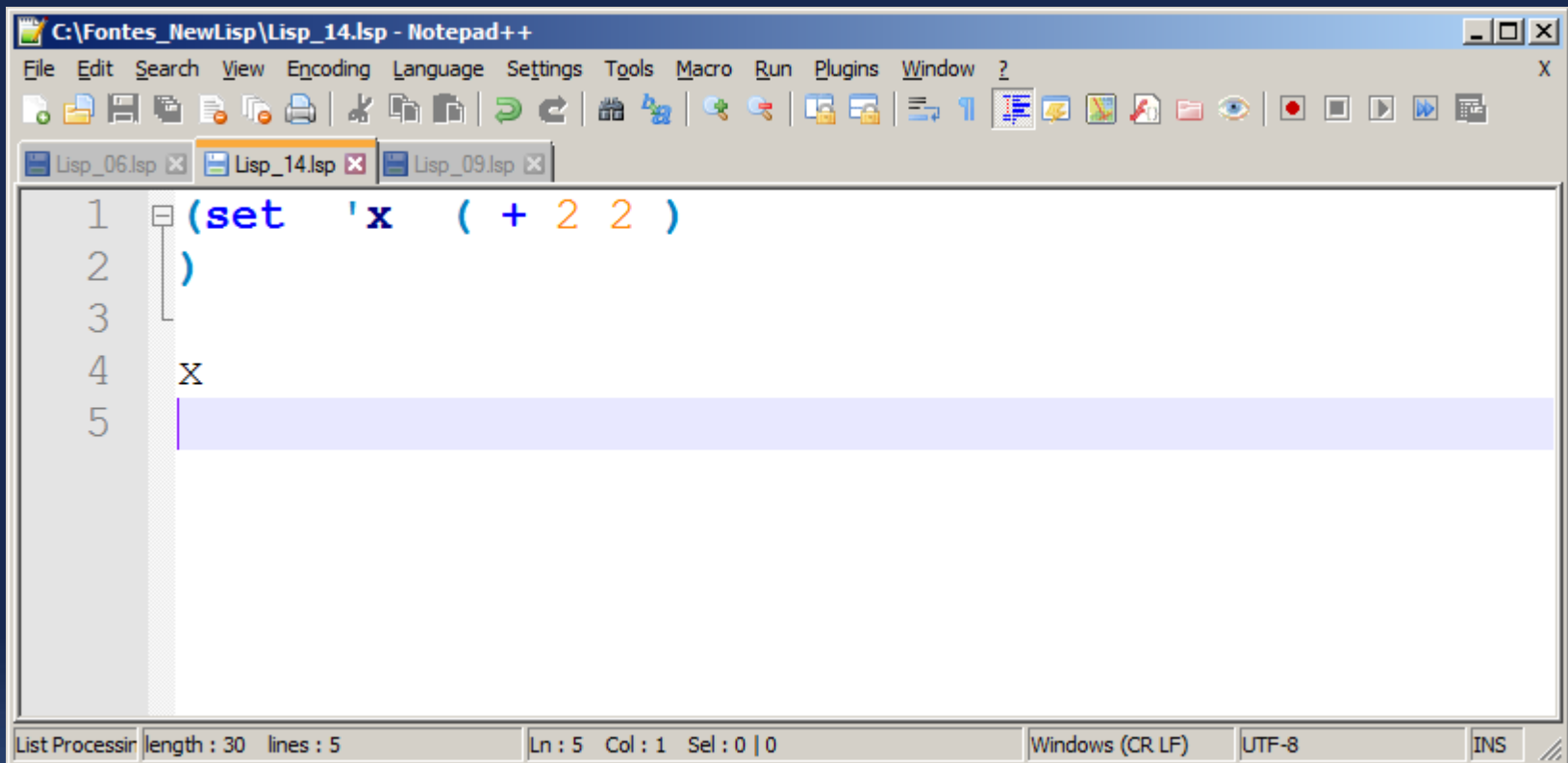
- ✓ **Símbolos** correspondem à “variáveis” em outras linguagens de programação;
- ✓ Na verdade, **newLISP** **não** usa símbolos de forma extensiva como se usa em outras linguagens de programação;
- ✓ A razão disto é que valores retornados por uma função estarão sendo continuamente entrados para outras funções, sem portanto a necessidade de armazená-los;
- ✓ No paradigma funcional o foco está na **transformação dos dados** de forma sucessiva até atingir-se o resultado final esperado.





Exemplo – Lisp_14.lsp

Exemplo



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_14.lsp - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations and editing. The tab bar shows three open files: Lisp_06.lsp, Lisp_14.lsp (active), and Lisp_09.lsp. The editor content is as follows:

```
1 (set 'x ( + 2 2 )  
2 )  
3  
4 x  
5
```

The status bar at the bottom displays: List Processor | length : 30 | lines : 5 | Ln : 5 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS.





Exemplo – Lisp_14.lsp



The image shows two overlapping windows. The top window is Notepad++ editing 'C:\Fontes_NewLisp\Lisp_14.lsp'. It contains the following code:

```
1 (set 'x ( + 2 2 )  
2 )  
3  
4 x  
5
```

The bottom window is the NewLisp REPL. It shows the execution of the code from the file above. The prompt '>' is followed by the code '(set 'x (+ 2 2))'. The next prompt shows the variable 'x' and its value '4'.

```
>  
>  
>  
>  
> (set 'x (+ 2 2) )  
4  
> x  
4  
>
```

A red jagged line points from the text 'x tem um binding ao valor 4' to the '4' output in the NewLisp window.





Exemplo - Lisp_15.lsp

C:\Fontes_NewLisp\Lisp_15.lsp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Lisp_06.lsp x Lisp_15.lsp x Lisp_09.lsp x

```
1 (set 'y '( + 2 2 )  
2 )  
3  
4 y  
5
```

NewLisp

```
>  
>  
>  
>  
>  
> (set 'y '(+ 2 2) )  
(+ 2 2)  
> y  
(+ 2 2)  
>
```

y tem um binding à lista (+ 2 2)





Definindo símbolos



- ✓ **Símbolos** também podem ser definidos pela função **setq**;
- ✓ A função **setq** considera que primeiro argumento é um **símbolo**, e dessa forma, não se exige que o argumento seja quoted (com apóstrofe).

```
NewLisp
>
>
>
>
> (setq PI 3.14159)
3.14159
> PI
3.14159
>
```





Definindo símbolos

- ✓ **Símbolos** também podem ser definidos pela função **define**;
- ✓ A função **set** vista anteriormente exige que o símbolo seja definido com **'**;
- ✓ Mas, **define** não exige que o argumento seja **quoted**;
- ✓ Define também é usado para definição de funções.

```
Command Prompt - newlisp
>
>
>
>
>
> (define PI 3.14159)
3.14159
> PI
3.14159
```





Definindo símbolos com define

```
Command Prompt - newlisp
>
> (define trab (+ 4 3) )
7
> trab
7
> _
```

```
Command Prompt - newlisp
>
>
> (define trab '(+ 4 3) )
(+ 4 3)
> trab
(+ 4 3)
>
>
>
>
```





Controlando o Fluxo de Execução

- ✓ Há diferentes formas de se controlar o Fluxo de Execução de nosso código;
- ✓ Todo o controle de fluxo em **newLISP** seguem determinadas regras da linguagem;
- ✓ A forma geral de definir o controle de fluxo é por meio de uma **lista** no qual o primeiro elemento é uma **keyword** (palavra reservada) seguida por uma ou mais expressões para serem avaliadas.

```
( keyword      expressao2      expressao3 ... )
```





if



- ✓ Consiste de um teste e uma ação;
- ✓ A grande maioria das coisas já conhecidas em **newLISP** são **true**;
- ✓ Há duas importantes coisas em **newLISP** que são **falsas**: **nil** e a lista vazia **()**.
- ✓ Qualquer coisa que **newLISP** não conhece também é retornado como **falso**.

```
Command Prompt - newlisp
>
> (define PI 3.14)
3.14
> (if PI)
3.14
> (if (= PI 3.14) "Verdade" "Falso")
"Verdade"
> (define A 3.14)
3.14
> (if A)
3.14
> (if (= A PI) "Verdade" "Falso")
"Verdade"
>
```





if



- ✓ O símbolo **x** é conhecido, portanto (if x) retorna **999** que é **true**;
- ✓ O símbolo **y** é desconhecido, portanto (if y) retorna **nil**, que corresponde a **false**.

```
Command Prompt - newlisp
C:\Users\VB_Aparecido>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, options:

> (set 'x 999)
999
> x
999
> (if x)
999
> (if a)
nil
> _
```





if



- ✓ O símbolo **var1** é conhecido, portanto **(if var1 "ola" "mundo")** retorna **"ola"** que é **true**;
- ✓ Mas, o símbolo **var1** após o **if**, continua com o valor original, **"teste"**
- ✓ A terceira expressão, representa uma ação **else**.

```
Command Prompt - newlisp

C:\Users\VB_Aparecido>(newlisp)
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, options:

> (define var1 "teste")
"teste"
> var1
"teste"
> (if var1 "ola" "mundo")
"ola"
> var1
"teste"
> _
```





if

- ✓ O símbolo **var3** não é conhecido, portanto (if **var3** "ola" "mundo") retorna "**mundo**";
- ✓ O símbolo **var2** após o if, continua com o valor original, "**uscs**";
- ✓ O símbolo **var3** não está definido e, portanto, ao acessá-lo será retornado **nil**.

```
Command Prompt - newlisp
C:\Users\VB_Aparecido>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, options: n
> (define var2 "uscs")
"uscs"
> var2
"uscs"
> (if var3 "ola" "mundo")
"mundo"
> var3
nil
> var2
"uscs"
>
```





Testes com if

```
Command Prompt - newlisp
C:\Users\VB_Aparecido>(newlisp)
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, options: n
> (if (> 4 3) "hello" "uscs")
"hello"
>
> (if (> 2 9) "hello" "uscs")
"uscs"
> _
```





Testes com if

- ✓ Se um símbolo é avaliado como **nil** (talvez porque ele **não** exista ou não tenha sido assinalado a algum valor) **newLISP** o considera **falso** e o teste retorna **nil**.

```
Command Prompt - newlisp
C:\Users\VB_Aparecido>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, options:
> (define x "uscs")
"uscs"
> (if x "sim" "nao")
"sim"
> (if y "sim" "nao")
"nao"
> (if y)
nil
>
```





If com diversos testes

- ✓ Pode-se empregar **if** com um número ilimitado de testes e ações;
- ✓ Neste caso, a lista de **if**'s consiste de uma série de pares teste-ação;

```
C:\Fontes_NewLisp\Lisp_16.lisp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_06.lisp x Lisp_16.lisp x Lisp_09.lisp x
1 (define x 99)
2
3 ( if
4
5     ( < x 0) (define a "impossivel")
6
7     ( < x 10) (define a "pequeno")
8
9     ( < x 99 ) (define a "medio")
10
11     ( < x 100) (define a "grande")
12
13     ( < x 40) (define a "exagerado")
14 )
15
16
List Processing language file length : 230 lines : 21 Ln : 4 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS
```





If com diversos testes

- ✓ Pode-se empregar **if** com um número ilimitado de testes e ações;
- ✓ Neste caso, a lista de **if**'s consiste de uma série de pares teste-ação;

The image shows a Notepad++ window with the file `C:\Fontes_NewLisp\Lisp_16.lsp` open. The code defines a variable `x` to 99 and then uses an `if` statement with multiple conditions. The Command Prompt window shows the execution of `newlisp` and `(load "Lisp_16.lsp")`, resulting in the value of `x` being printed as 99.

```
(define x 99)

(if
  (< x 0) (do)
  (< x 10) (do)
  (< x 99) (do)
  (< x 100) (do)
  (< x 40) (do)
)
```

```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6

> (load "Lisp_16.lsp")
"grande"
> x
99
>
```



Função when

- ✓ A função **when** permite que duas ou mais ações sejam processadas se um teste for satisfeito;
- ✓ Pode-se usar **when** como um **if** sem a parte 'else'.

```
C:\Fontes_NewLisp\Lisp_17.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Lisp_06.lsp x Lisp_17.lsp x Lisp_09.lsp x
1 (define x 5)
2
3 ( when ( < x 10)
4
5     ( define a 20 )
6
7     ( define b 30 )
8
9     ( define c 40 )
10
11    ( define d 50 )
12
13    ( define e 60 )
14 )
15
16
List Processing language file length : 162 lines : 21 Ln : 2 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS
```





Função when



The screenshot shows a Notepad++ window titled "C:\Fontes_NewLisp\Lisp_17.lsp - Notepad++" with the following Lisp code:

```
1 (define x 5)
2
3 (when (< x 10)
4
5   (define a 20)
6
7   (define
8
9   (define
10
11   (define
12
13   (define
14 )
15
16
```

Below the code editor is a Command Prompt window titled "Command Prompt - newlisp" showing the execution of the Lisp code:

```
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 libffi, optio
> (load "Lisp_17.lsp")
60
> a
20
> b
30
> c
40
> d
50
> e
60
>
```

The status bar at the bottom of the Notepad++ window indicates "List Processing language file | length : 162 | lines : 21".





Trabalhando com Listas

- ✓ A função **sequence** gera uma **sequência** de números em um intervalo, com um valor opcional definindo o passo da sequência.
- ✓ Quando o passo da sequência for omitido, o valor **1** é assumido.

```
Command Prompt - newlisp
C:\Fontes_NewLisp>newlisp
newLISP v.10.7.1 64-bit on Windows IPv4/6 1

> (sequence 1 10)
(1 2 3 4 5 6 7 8 9 10)
> (sequence 10 2)
(10 9 8 7 6 5 4 3 2)
> (sequence 1 5 0.5)
(1 1.5 2 2.5 3 3.5 4 4.5 5)
>
```





Função dolist

```
E:\NewLisp\dolist2.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tarefa_T9_Solucao_SCRIPT.sql Tarefa_T10_Solucao_SCRIPT.sql synced.rkt blueboxes.rkt Arquivo.txt lsp_12.lsp Lista_Presenca_Palestra_Navarro_16_05_2020_TXT.txt dolist2.lsp

1 ; ---- inicio: dolist2.lsp
2 (define lista '(1 3 5 9 10) )
3
4 (dolist (elementoLista lista )
5       (* 2 elementoLista)
6       (println "Elemento:  " (* 2 elementoLista) )
7 )
8 ; A lista continua com os mesmos valores
9 lista
10 ; --- Fim: dolist2.lsp
11
12
```

```
Command Prompt - newlisp
>
>
(load "dolist2.lsp")

Elemento:  2
Elemento:  6
Elemento: 10
Elemento: 18
Elemento: 20
(1 3 5 9 10)
>
```

List Processing language file | length : 252 | lines : 12 | Ln : 12 Col : 5 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS





Função dolist

```
E:\NewLisp\dolist3.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tarefa_T9_Solucao_SCRIPT.sgl Tarefa_T10_Solucao_SCRIPT.sgl synced rktl blueboxes rktl Arquivo.txt lisp_12.lsp Lista_Presenca_Palestra_Navaro_16_05_2020_TXT.txt dolist3.lsp

1 ; ---- inicio: dolist3.lsp
2 (define lista '(1 3 5 9 10) )
3 (define contador 0)
4
5 (dolist (elementoLista lista)
6
7     (println "Elemento " contador ": " elementoLista)
8     (define contador (+ contador 1) )
9 )
10 ; --- Fim: dolist2.lsp
11
12
```

```
Command Prompt - newlisp
>
>
(load "dolist3.lsp")

Elemento 0: 1
Elemento 1: 3
Elemento 2: 5
Elemento 3: 9
Elemento 4: 10
5
>
```

List Processing language file | length : 243 lines : 12 | Ln : 12 Col : 5 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS





Contador de Loop mantido por newLISP

- ✓ newLISP automaticamente mantém um contador de loop chamado **\$idx**

The screenshot shows a Notepad++ window with a newLISP script and a Command Prompt window showing the execution results.

```
1 ; ---- inicio: dolist4.lsp
2
3 (define lista '(1 3 5 9 10) )
4
5 (dolist (elementoLista lista )
6
7     (println "Elemento " $idx ": " elementoLista)
8 )
9 ; --- Fim: dolist4.lsp
10
11
```

The Command Prompt window shows the following output:

```
>
>
> (load "dolist4.lsp")
Elemento 0: 1
Elemento 1: 3
Elemento 2: 5
Elemento 3: 9
Elemento 4: 10
10
>
```





Função map

- ✓ A função **map** pode ser usada para se aplicar uma **função** à todo elemento de uma lista;
- ✓ A **função** a ser aplicada aos elementos da lista é definida de forma anônima por **fn**.

```
E:\NewLisp\map01.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tarefa_T9_Solucao_SCRIPT.spl Tarefa_T10_Solucao_SCRIPT.spl synced.rktdd blueboxes.rktdd Arquivo.txt lsp_12.lsp Lista_Presenca_Palestra_Navaro_16_05_2020_TXT.txt map02.lsp map01.lsp x3

1 ; ---- inicio: map01.lsp
2 ; ---- fn significa definição de função anônima (lambda)
3 ; Esse exemplo soma 2 à todo elemento da lista
4
5 (define lista '(1 3 5 9 10) )
6
7 (define listasaida (map
8   (fn (x) (+ x 2)) ;função anônima
9
10  lista ;lista
11 )
12 )
13
14 listasaida
15
16 ; --- Fim: map01.lsp
17
18
```

```
Command Prompt - newlisp
>
>
>
(load "map01.lsp")
(3 5 7 11 12)
>
```





Função map

- ✓ A função **map** pode ser usada para se aplicar uma **função** à todo elemento de uma lista;
- ✓ A **função** a ser aplicada aos elementos da lista pode ser anonimamente definida por expressão **lambda** (que equivale ao **fn**)

```
1 ; ---- inicio: map02.lsp
2 ; ---- fn significa definição de função anônima
3 ; Esse exemplo multiplica por 10 todo elemento da lista
4
5 (define lista '(1 3 5 9 10) )
6
7 (define listasaida (map
8   (lambda (x) (* x 10)) ;função anônima
9   lista                  ;lista
10 )
11 )
12
13 listasaida
14
15
16 ; --- Fim: map02.lsp
17
18
```

```
Command Prompt - newlisp
>
>
>
> (load "map02.lsp")
(10 30 50 90 100)
>
```





Função map

```
E:\NewLisp\map03.lsp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tarefa_T9_Solucao_SCRIPT.sql Tarefa_T10_Solucao_SCRIPT.sql synced rkttd blueboxes rkttd Arquivo.txt lsp_12.lsp Lista_Presenca_Palestra_Navarro_16_05_2020_TXT.txt map02.lsp map03.lsp

1 ; ---- inicio: map03.lsp
2 ; ---- fn significa definição de função anônima (lambda)
3 ; Esse exemplo soma os elementos de duas listas
4
5 (define lista1 '(1 3 5 9 10) )
6
7 (define lista2 '(2 4 6 8 10) )
8
9 (define listasaida (map
10                     (fn (x y) (+ x y))           ;função anônima
11                     lista1
12                     lista2
13                     )
14 )
15
16
17 listasaida
18
19
```

```
Command Prompt - newlisp
>
>
> (load "map03.lsp")
(3 7 11 17 20)
>
```

