

Paradigmas de Programação

Prof. Maicon R. Zatelli

LISP - Programação Funcional Introdução

Universidade Federal de Santa Catarina
Florianópolis - Brasil

LISP - Introdução

LISP (LISt Processor) - 1958

Funcional

Criada por John McCarthy

Desenvolvida para processamento de listas

Puramente recursiva e não iterativa

<https://common-lisp.net/>

https://www.tutorialspoint.com/execute_lisp_online.php

LISP - Compilando e Executando

Faça o download do clisp e instale em seu computador:

- <https://clisp.sourceforge.io/>

Para compilar use o seguinte comando:

- `clisp -c hello.lisp`

Para executar, basta executar: `clisp hello.fas`

LISP - CLISP

É o cliente (e compilador) LISP.

Pode ser aberto por meio do comando `clisp`

Por meio do CLISP, pode-se diretamente interpretar/executar código LISP.

LISP

Considerados brancos: espaços, <cr>, <lf>, tabulações, comentários.

Comentário de linha: ;

Comentário de bloco: #| |#

LISP

Operadores

Aritméticos

- +, -, *, /, mod, expt

Lógicos

- or, and, not

Relacionais

- =, /=, >, >=, <=, <, eq (para símbolos)

Operadores

LISP utiliza notação pré-fixada, ou seja $(+ 5 2)$ é o mesmo que somar 5 e 2.

$(- (+ 5 2) 3)$ é o mesmo que $(5 + 2) - 3$

$(/ (- (+ 5 2) 3) 2)$ é o mesmo que $((5 + 2) - 3) / 2$

$(/ (+ 5 2) (- 7 3))$ é o mesmo que $(5 + 2) / (7 - 3)$

Entrada e Saída de Dados

Entrada

`(read)` lê até encontrar espaço

`(read-char)` lê exatamente um caracter

`(read-line)` lê até encontrar quebra de linha

Saída

`(write-line x)` imprime e quebra linha

`(write x)` imprime e não quebra linha

Valores lógicos

T = True

NIL = False

Funções básicas

Converte x para string

```
(write-to-string x)
```

Concatena sequências (strings, vetores, listas)

```
(concatenate 'string "ufsc " "paradigmas " "hoje")  
(concatenate 'list '(1 2 3) '(4 5) "abc")  
(concatenate 'vector '(1 2 3) '(4 5) "abc")  
(concatenate 'string "Resultado: " (fatorial x))
```

Funções básicas

Atribuição

```
(setq z (read))  
(setq z (fatorial x))
```

\sqrt{x}

```
(sqrt x)
```

LISP

```
(defun hello()  
  "Hello world!"  
)  
  
(defun main()  
  (write-string (hello))  
)  
  
(main)
```

LISP

```
(defun ehpar (n)
  (= (mod n 2) 0)
)

(defun main()
  (setq x (read))
  (write-line (write-to-string (ehpar x)))
  (setq y (read))
  (write-line (write-to-string (ehpar y)))
)

(main)
```

LISP

```
(defun fatorial (n)
  (if (= n 0)
      1
      (* n (fatorial (- n 1)))
  )
)

(defun main()
  (setq y (read))
  (write-line (write-to-string (fatorial y)))
)

(main)
```

LISP

```
(defun selecciona (n)
  (cond
    ((= n 1) "Caso 1")
    ((= n 2) "Caso 2")
    ((= n 3) "Caso 3")
    ((= n 4) "Caso 4")
  )
)

(defun main()
  (setq y (read))
  (write-line (write-to-string (selecciona y)))
)

(main)
```

LISP

```
(defun seleciona (c)
  (cond
    ((eq c '@) "Arroba")
    ((eq c '!) "Exclamação")
    (t "Outro")
  )
)

(defun main ()
  (setq x (read))
  (write-line (seleciona x))
)

(main)
```


LISP - Alguns Links Úteis

- <https://www.tutorialspoint.com/lisp/index.htm>
- https://www.tutorialspoint.com/lisp/lisp_input_output.htm
- https://www.tutorialspoint.com/lisp/lisp_characters.htm
- https://www.tutorialspoint.com/lisp/lisp_strings.htm

Ver atividade no Moodle