GUÍA BÁSICA DE SCHEME v.4

Esta guía básica pretende ser una introducción elemental al lenguaje de programación Scheme. Se presenta como una guía de "comienzo rápido" de tal forma que permita conocer de una forma muy esquemática los elementos básicos del lenguaje y posibilite realizar una programación elemental con este lenguaje de programación.

I. SINTAXIS BÁSICA DEL LENGUAJE

expresión → átomo | lista

átomo → número | string | identificador | carácter | boolean

lista → '(' secuencia-expresión ')'

secuencia-expresión → expresión secuencia-expresión | expresión

función → (nombre_función $argumento_1$ $argumento_2$ … $argumento_n$)

II. TIPOS DE DATOS

átomo	Elemento básico → Identificador, letras y dígitos
boléanos	#t (verdadero), #f (falso)
números	42, 2+3i, 3.16, 4/6
listas	(Conjunto de elementos entre paréntesis): formado por átomos o listas.
carácter	#\a, etc.
string	"hola"

III. OPERADORES MATEMÁTICOS

Operador	Operación
+	Suma cada uno de los argumentos
*	Multiplica cada uno de los argumentos
-	Primer argumento menos los restantes o negación del número
/	Primer argumento entre cada uno de los restantes
abs	Valor absoluto del argumento
exp	Exponencial
expt	Dados dos argumentos: *x* e *y*, calcula x^y
max	Máximo de sus argumentos
min	Mínimo de sus argumentos
sqrt	Raíz cuadrada del argumento no negativo

Ejemplos:

(+ 1 2 3) → 6
(- 5.3 2) → 3.3
(- 5 2 1) → 2
(* 1 2 3) → 6
(/ 6 3) → 2
(/ 22 7) → 22/7
(abs -4) → 4
(expt 2 3) → 8
(max 1 3 4 2 3) → 4
(min 1 3 4 2 3) → 1

IV. FUNCIONES DE COMPARACIÓN ARITMÉTICAS

Función	Significado
=	Igual (átomos numéricos)
>	Mayor que
<	Menor que
>=	Mayor o igual
<=	Menor o igual
even?	¿Es Número par?
odd?	¿Es Número impar?
zero?	¿Número cero?

Nota: devuelven #t (verdadero) o #f (falso)

Ejemplos:

(= 5 8) → false
(> 8 5) → true
(< 6 7) → true
(>= 9 1) → true
(<= 5 5) → true
(even? 2) → true
(odd? 8) → false
(zero? 1) → false

V. FUNCIONES PARA MANIPULAR LISTAS

Función	Argumento	Operación
car	Lista	Devuelve el primer elemento de la lista.
cdr	Lista	Devuelve la lista eliminando su primer elemento.
cons	Átomo | Lista , Lista	Lista formada con el primer argumento más los elementos del segundo.
list	Átomo | Lista	Lista con los parámetros pasados en orden.

Ejemplos:

(car '(A B C))	→	'A	(cons 'A '())	→	'(A)
(car '((A B) C D))	→	'(A B)	(cons 'A '(B C D))	→	'(A B C D)
(car 'A)	→	Error	(cons '() '(A B))	→	'(() A B)
(car '(A))	→	'A	(cons '(A B) '(C D))	→	'((A B) C D)
(car '())	→	Error	(cons '(A B) (cons 'C '(D)))	→	'((A B) C D)
(car '(+ 3 4))	→	'+	(cons 5 '(A B C))	→	'(5 A B C)
(car (car (1 2)))	→	Error	(cons '5 '(A B C))	→	'(5 A B C)
(car (car '(1 2)))	→	Error	(list 'A 'B 'C)	→	'(A B C)
(cdr '(A B C))	→	'(B C)	(list 'A '(B C))	→	'(A (B C))
(cdr '((A B) C D))	→	'(C D)	(list '() 'A)	→	'(() A)
(cdr 'A)	→	Error	(list 'A '(BC) 'D)	→	'(A (B C) D)
(cdr '(+ 3 4))	→	'(3 4)	(list 'A (cdr '(A (B C))) (car '(D)))	→	'(A ((B C)) D)
(cdr (cdr '(1 2)))	→	'()			

VI. FUNCIONES DE COMPARACION

Función	Argumento	Operación
eq?	Átomo | Lista, Átomo | Lista	TRUE si ambos argumentos son átomos e iguales. FALSE en caso contrario.
null?	Átomo | Lista	TRUE si su único argumento es una lista vacía. FALSE en caso contrario.
list?	Átomo | Lista	TRUE si su único argumento es una lista. FALSE en caso contrario.
equal?	Átomo | Lista, Átomo | Lista	TRUE si ambos argumentos son iguales. FALSE en caso contrario.

VII. FUNCIONES BOOLEANAS

Función	Operación	sintaxis
and	Devuelve #f tan pronto uno de sus argumentos es falso.	(and arg1 arg2 … argN)
or	Devuelve #t tan pronto uno de sus argumentos es verdadero.	(or arg1 arg2 … argN)
not	Negación de un valor booleano	(not arg1)

VIII. FUNCIONES DE EVALUACIÓN

Función	Operación
eval	Evalúa la expresión que recibe como argumento
quote ó (‘)	No evalúa el parámetro pasado

IX. FUNCIÓN DE ASIGNACIÓN

Función	Operación
let	Asigna nombres a resultados de subexpresiones

Ejemplos:

```
( let ( (a 2) (b 3) ) (+ a b) )
```

X. DEFINICIÓN DE FUNCIONES

(**define** (nombre_función argumentos) sentencia … sentencia)	(**define** nombre_función (**lambda** (argumentos) (expresión)))	(**lambda** (argumentos) (expresión))

Ejemplos:

```
(define (cuadrado num)
        (* num num))
```

XI. ASOCIACIÓN DE EXPRESIONES A SÍMBOLOS

(**define** símbolo expresión)

Ejemplo:

```
(define PI 3.14159)
```

XI. CONTROL DE FLUJO

(**if** condición expresión-entonces expresión_sino)	(**cond** (condición1 expresión {expresión}) … (condición1 expresión {expresión}) (**else** expresión {expresión})) * Sólo evalúa la primera condición válida

XII. FUNCIONES DE ENTRADA Y SALIDA

(**display** lista | elemento | “string”)
(**newline**)

Ejemplos:

```
(display '(A B))
(newline)
```