

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

DEFINICIONES DE FUNCIONES

Objetivo:

Que el alumno aprenda a crear sus propias funciones.

TENER EN CUENTA:

- Las variables nuevas que se definan deben ser locales, tener mensajes descriptivos para su ingreso y las validaciones correspondientes.
- Las funciones deben ser genéricas. Se deben poder ejecutar para cualquier valor que se ingresen en las variables.
- Se recomienda leer el documento REGLAS DE ESTILO LISP que se ha dejado en el Aula dentro de Trabajos Prácticos

Ejercicio Nº 1

Definir una función predicado llamada **palíndromo**, que determine si una lista ingresada por el operador, es una lista palíndromo. (se lee igual de izquierda a derecha y de derecha a izquierda).

Por ejemplo: (I T A T I) es una lista palíndromo

Ejercicio Nº 2

Definir una función llamada **clima**, la que a partir de la temperatura que será ingresada por el operador, me indique el estado del clima. Teniendo en cuenta

Temperatura	Clima
< 0	Helado
= 0 y < 10	Frio
= 10 y < 20	Templado
= 20 y < 30	Cálido
>30	Abrasador

Ejercicio Nº 3

Definir una función llamada **area-triangulo** que devuelve el área de un triángulo a partir de su base y su altura. La base y la altura son ingresadas como parámetros.

Ejercicio Nº 4

Definir la función predicado llamada **es_Cuadrado_Perfecto**, la cual deberá evaluar si un número que es ingresado como parámetro, es un cuadrado perfecto. Un número es un cuadrado perfecto si su raíz cuadrada es un número entero. (Puede ayudarse con una o mas de las siguientes funciones ZEROP, MOD, SQRT, TRUNCATE, INTEGERP)

Ejercicio Nº 5

Definir una función llamada **clasifico-triángulos**, la que a partir de los valores de los lados de un triángulo ingresados por el operador, clasifique el mismo en: isósceles, equilátero o escaleno. Tener en cuenta que todo triángulo debe cumplir que. "Un lado es menor que la suma de los otros dos y mayor que la diferencia".

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

Ejercicio N° 6

A partir la mínima temperatura del día de ayer y de dos listas:

- **min_marzo**: que contendrá las mínimas temperaturas registradas para cada uno de los días de marzo
- **min_abril**: que contendrá las mínimas temperaturas registradas para cada uno de los días de abril

determinar si la temperatura mínima de ayer se registró también en marzo o en abril. (el valor atómico y las dos listas deben ser ingresadas como parámetros)

Ejercicio N° 7

Crear las siguientes funciones de rotación:

- a. una función llamada **derecha** que rote a la DERECHA los elementos de una lista ingresada como parámetro haciendo que su primer elemento pase a ser el último.

Por ejemplo:

>> (rotar-derecha '(1 2 3 4)) ==> (2 3 4 1).

- b. una función llamada **izquierda** que rote a la IZQUIERDA los elementos de una lista ingresada como parámetro haciendo que el último elemento pase a ser el primero. Por ejemplo,

>> (rotar-izquierda '(1 2 3 4)) ==> (4 1 2 3).

Ejercicio N° 8

Definir una función llamada **mediano**, la que a partir de tres valores numéricos ingresados como parámetros, devuelva el valor mediano (puede ayudarse con las funciones max y min). El valor mediano será aquel que no es ni el mayor ni el menor valor.

Ejercicio N° 9

Definir una función llamada **posición**, que reciba como argumentos un elemento y una lista e indique la posición que ocupa el elemento en la lista.

Ejercicio N° 10

Definir una función llamada **mi-segundo** que recibe una lista y un átomo como parámetros. La misma debe devolver una nueva lista donde el átomo ingresado como parámetro ocupará la primera posición de la lista. (Recuerde que las posiciones comienzan a contarse desde la 0)

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

Ejercicio N° 11

Analizar la siguiente función:

```
(defun analizo (l a)
  (cond
    ((not (member a l)) "no se encuentra")
    ( (oddp (length (member a l))) (list (member a l) (length (member a l))))
    ( T "no es par")
  ))
```

y determinar el resultado que arrojaría la misma si la evalúo con los siguientes parámetros:

- a. (analizo '(2 3 4 5 6 7) 8) Resultado:
- b. (analizo '(2 3 4 5 6 7) 5) Resultado:

Ejercicio N° 12

Analizar la siguiente función:

```
(defun calculo (n1 n2)
  (if (> n1 n2) (+ (* n1 n2) ) (/ n2 n1)))
```

y determinar el resultado que arrojaría la misma si la evalúo con los siguientes parámetros:

- a. (calculo 5 3) Resultado:
- b. (calculo 3 18) Resultado:

Ejercicio N° 13

Analizar la siguiente función

```
(defun primero-ultimo (l)
  (if
    (numberp (car (last l))) (cons (car (last L)) (butlast l))
    (car (last L))))
```

y determinar el resultado que arrojaría la misma si la evalúo con los siguientes parámetros:

- a. (primero-ultimo ' ((5 6) 1 2 3 4)) Resultado:
- b. (primero-ultimo '(5 6 7 X Z)) Resultado:

PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN - LENGUAJE LISP -

Ejercicio N° 14

A partir de tres valores numéricos que son ingresados por el operador y representan el nivel de un río en tres días consecutivos, obtener:

- a. La dispersión del nivel del río. Siendo la dispersión, la diferencia entre el valor más alto y el más bajo. Estos valores deben ser ingresados.
- b. Determinar si esta dispersión corresponde a días parejos, locos o normales.
 - Son días parejos si la dispersión es chica (menos de 30 cm)
 - Son días locos si la dispersión es grande (más de un metro)
 - Son días normales si no son ni parejos ni locos.

NOTA: Se aconseja desarrollar los ítems a y b en funciones diferentes y luego llamar en la función definida en el punto a, a la función definida en el punto b.

Ejercicio N° 15:

En una plantación de pinos, de cada árbol se conoce la altura expresada en **metros**. El peso de un pino se puede calcular a partir de la altura así:

- 3 kg por cada centímetro hasta 3 metros,
- 2 kg por cada centímetro arriba de los 3 metros

Por ejemplo:

- 2 metros pesan 600 kg, porque $200 * 3 = 600$
- 5 metros pesan 1300 kg, porque los primeros 3 metros pesan 900 kg y los siguientes 2 pesan los 400 restantes.

Los pinos se usan para llevarlos a una fábrica de muebles, a la que le sirve árboles de entre 400 y 1000 kilos, un pino fuera de este rango no le sirve a la fábrica.

- a. Definir la función **pesoPino**, que recibe la altura de un pino y devuelve su peso.
- b. Definir la función predicado **esPesoUtil**, recibe un peso en kg y responde verdadero si un pino de ese peso le sirve a la fábrica
- c. Definir la función predicado **sirvePino**, recibe la altura de un pino y responde verdadero si un pino de ese peso le sirve a la fábrica.

Ejercicio N° 16

Definir una función **cuantoPagaCadaUno**, que recibe como parámetros el precio de 1 pizza y una cantidad de comensales y devuelva cuánto debe pagar cada uno. Tener en cuenta que cada comensal come 3 porciones y sólo se pueden comprar pizzas enteras (que tiene cada una 8 porciones).

Por ejemplo:

- Son 3 comensales, se necesitan entonces 9 porciones, lo que son dos pizzas, que salen \$ 120 cada una, lo que implica $\$240$ a dividir entre 3 = \$80 cada uno.

Ayuda: Recuerde la diapositiva Otras funciones aritméticas