

Guía de Estudio Racket

Objetivo General

Comprender y aplicar los conceptos de funciones, variables, funciones anónimas (lambda) y operaciones funcionales sobre listas en el lenguaje Racket.

Conceptos Clave

- Función: una relación entre entradas y salidas, definida con `define` o `lambda`.
- Variable: un nombre asociado a un valor, no una celda modificable.
- Lambda: permite definir funciones anónimas, por ejemplo `(lambda (x) (+ x 1))`.
- `map`, `filter`, `foldl`: aplican funciones, filtran o acumulan valores sobre listas.

Ejercicio 1 - Suma de elementos de una lista

Objetivo: Practicar el uso de `foldl` para realizar acumulaciones sobre listas.

Entrada: Entrada: Una lista de números, por ejemplo `'(1 2 3 4 5)`.

Salida esperada: Salida esperada: La suma total de los elementos, en este caso 15.

Ejercicio 2 - Filtrar mayores a un número

Objetivo: Aplicar la función `filter` y el uso de funciones anónimas (lambda).

Entrada: Entrada: Una lista de números y un valor `n`. Ejemplo: `'(5 3 8 1 10)` y `n = 4`.

Salida esperada: Salida esperada: Lista con los elementos mayores a `n`, `'(5 8 10)`.

Ejercicio 3 - Producto de una lista

Objetivo: Practicar la función `foldl` utilizando el operador de multiplicación.

Entrada: Entrada: Una lista de números, por ejemplo `'(2 3 4)`.

Salida esperada: Salida esperada: El producto de los elementos, en este caso 24.

Ejercicio 4 - Aplicar una función a todos los elementos

Objetivo: Comprender el uso de map para transformar listas.

Entrada: Entrada: Una función y una lista, por ejemplo duplicar y '(1 2 3 4).

Salida esperada: Salida esperada: Una lista con los resultados aplicados, '(2 4 6 8).

Ejercicio 5 – Cuadrado de números

Objetivo: Definir y aplicar una función pura que calcule el cuadrado de un número.

Entrada: Entrada: Una lista de números, '(1 2 3 4 5).

Salida esperada: Salida esperada: Una lista con los cuadrados, '(1 4 9 16 25).

Ejercicio 6 – Duplicar los valores de una lista

Objetivo: Reforzar el uso de map con funciones simples.

Entrada: Entrada: Una lista de números, '(2 4 6).

Salida esperada: Salida esperada: '(4 8 12).

Ejercicio 7 – Seleccionar los números pares

Objetivo: Practicar filter con una condición lógica.

Entrada: Entrada: '(1 2 3 4 5 6).

Salida esperada: Salida esperada: '(2 4 6).

Ejercicio 8 – Calcular el promedio de una lista

Objetivo: Combinar foldl con operaciones aritméticas y length.

Entrada: Entrada: '(4 6 8 10).

Salida esperada: Salida esperada: 7.0.

Ejercicio 9 – Elevar los elementos a una potencia n

Objetivo: Aplicar map con una función que reciba un parámetro adicional.

Entrada: Entrada: Lista '(2 3 4) y $n = 2$.

Salida esperada: Salida esperada: '(4 9 16).

Ejercicio 10 – Contar cuántos elementos superan un valor dado

Objetivo: Combinar filter y length para obtener un conteo.

Entrada: Entrada: '(5 7 2 9 1 10) y $n = 6$.

Salida esperada: Salida esperada: 3.

Ejercicio 11 - Aplicar una función dos veces

Objetivo: Comprender la composición de funciones aplicadas secuencialmente.

Entrada: Entrada: Función `sqrt` y valor 16.

Salida esperada: Salida esperada: 2.0.

Ejercicio 12 - Mini reto integrador

Objetivo: Integrar `map`, `filter` y `foldl` en un mismo programa.

Entrada: Entrada: Lista '(2 5 6 8 3 10).

Salida esperada: Salida esperada: Producto de los elementos mayores a 5, es decir 480.

Recomendaciones

- Usa DrRacket para probar cada ejercicio.
- Agrega comentarios con `;;` para describir la función y su propósito.
- Guarda todos los ejercicios en un repositorio de Git para documentar tu progreso.