

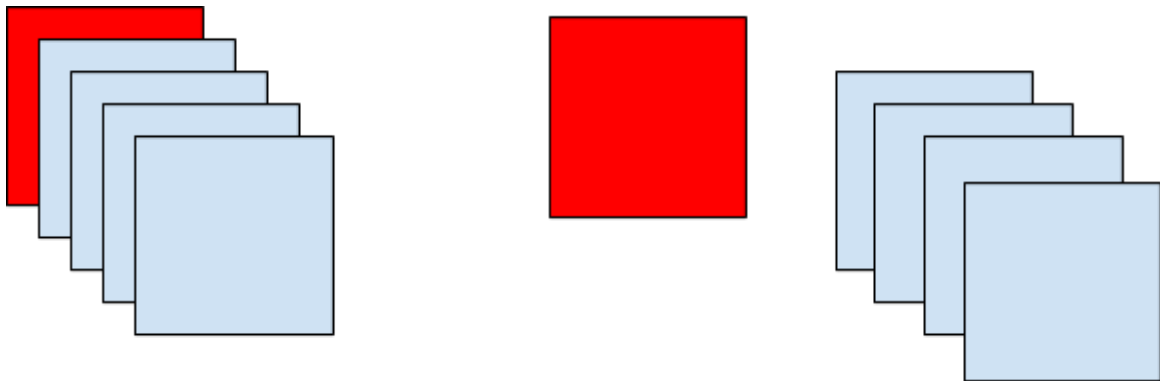
002. Sumar los elementos de una lista

Escriba una función **suma** que toma una lista como argumento y devuelve un número, que corresponde a la suma de todos los elementos de la lista. La lista es una lista simple, que no tiene listas anidadas. Ejemplos de invocaciones:

```
(suma '()) -> 0  
(suma '(1 2 3 4 5)) -> 15
```

Podemos ver que hay dos casos bien diferenciados: la lista vacía y la lista que no está vacía.

- Si una lista está vacía, el resultado de su suma es 0.
- Si una lista no está vacía, implica que hay al menos un elemento que es la cabeza de la lista, y además hay una cola. Es claro que la suma de la lista completa se obtiene adicionando el valor de la cabeza más la suma de la cola:



Vamos a codificar esa idea en Scheme:

Entonces sabemos que la lista es vacía o no lo es, son los dos casos que nos interesan, podemos usar una forma **if** para diferenciarlos:

```
(define suma  
  (lambda (lst)  
    (if (null? lst)  
        0  
        (+ (car lst) (suma (cdr lst))))))
```

Compare **suma** con **largo** del ejercicio 01.

Si usamos los superpoderes de **apply**, podemos escribir **(apply + lst)** como cuerpo de la definición. La comprensión de esto se deja para cuando veamos funciones de orden superior.