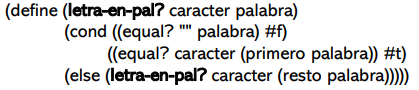
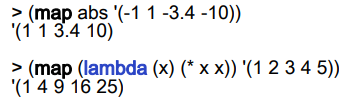
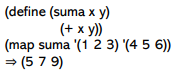
Las **funciones de orden superior**son funciones que cumplen al menos una de las siguientes condiciones:

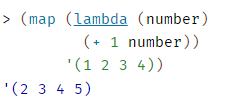
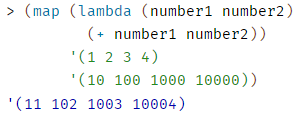
1. Tomar una o más funciones como entrada
2. Devolver una función como salida

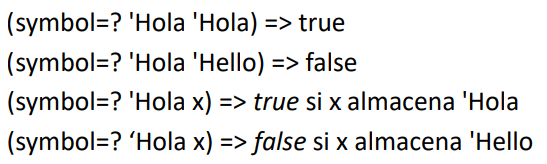
Pueden usarse como parámetros para otras funciones. El uso de funciones de orden superior permite realizar un código muy conciso y expresivo. Ya no es necesario escribir la recursión de forma explícita, sino que la función de orden superior es la que se encarga de abstraerla.

**Filter** *proc* *lst*) → list?

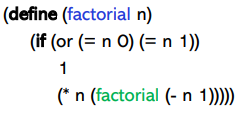
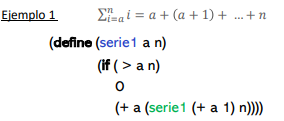
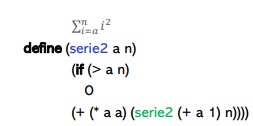
 

La función **map** aplica una función a todos los elementos de una lista y devuelve la lista resultante *proc* *lst* ...+ → list?

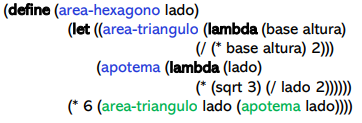
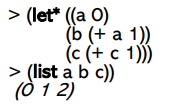


**Symbol**

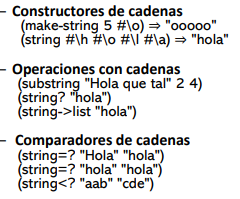
Un símbolo es una secuencia de caracteres precedida por un signo de comillas simple. Es una operación de comparación, recibe dos símbolos y produce true si y sólo si los dos símbolos son idénticos.

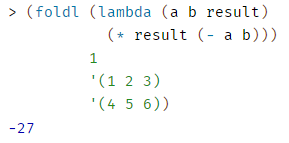
**Funciones recursivas**

**Uso del let**

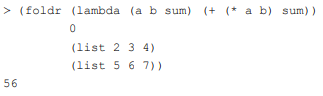


**Tratamiento de cadenas Cons**



**Foldl** *proc* *init* *lst 🡪*[any/c](https://docs.racket-lang.org/reference/data-structure-contracts.html#%28def._%28%28lib._racket%2Fcontract%2Fprivate%2Fmisc..rkt%29._any%2Fc%29%29) recorre las listas de derecha a izquierda

https://cdn.discordapp.com/attachments/630846797904085012/666376305801035824/unknown.png

**Foldr** *proc* *init* *lst 🡪* [any/c](https://docs.racket-lang.org/reference/data-structure-contracts.html#%28def._%28%28lib._racket%2Fcontract%2Fprivate%2Fmisc..rkt%29._any%2Fc%29%29) de izquierda a derecha