

FUNCIONES DE MANEJO DE LISTAS	OBSERVACIONES
<p>(foreach nombre lista expr ...) <i>Evalúa cada expresión para todos los miembros de una lista</i></p>	<ul style="list-style-type: none"> • Avanza por la lista, asignando un nombre a cada elemento y evalúa cada expr para todos los elementos de la lista. Puede especificar tantas expr como desee. La función foreach devuelve el resultado de la última expr evaluada.
<p>(length lista) <i>Devuelve un número entero que indica el número de elementos de una lista</i></p>	
<p>(listp elemento) <i>Comprueba si un elemento es una lista</i></p>	<ul style="list-style-type: none"> • Devuelve T si elemento es una lista y devuelve nil en caso contrario.
<p>(mapcar función lista1 ... listan) <i>Devuelve una lista como resultado de ejecutar una función con los elementos individuales de una o varias listas proporcionadas como argumentos de dicha función</i></p>	<ul style="list-style-type: none"> • El número de listas debe coincidir con el número de argumentos requeridos por función. • La función lambda puede especificar una función sin nombre para ejecutarse con mapcar. Esto resulta útil cuando algunos de los argumentos de función son constantes o se proporcionan mediante otros métodos.
<p>(member expr lista) <i>Busca una expresión en una lista y devuelve el resto de la lista, desde la primera ocurrencia de la expresión encontrada</i></p>	<ul style="list-style-type: none"> • Si expr no aparece en la lista, member devuelve nil.
<p>(reverse lista) <i>Devuelve una lista con sus elementos colocados en orden inverso</i></p>	
<p>(subst elemento_nuevo elemento_antiguo lista) <i>Busca un elemento antiguo en una lista y devuelve una copia de ésta con un elemento nuevo en lugar de cada aparición del elemento antiguo</i></p>	<ul style="list-style-type: none"> • Si el elemento_antiguo no se encuentra en la lista, subst devuelve ésta intacta. • Cuando se utiliza conjuntamente con assoc, subst proporciona una forma útil de sustituir el valor asociado a una clave de una lista de asociaciones.

EJEMPLOS DE FUNCIONES DE MANEJO DE LISTAS

```
; Supógase que la variable l = ( ( 10 10 ) ( 20 20 ) ( 30 30 ) . . . ( 100 100 ) )
```

```
( setq longitud ( length l ) )
```

```
10
```

```
( setq elem ( member ( list 30 30 ) l ) )
```

```
((10 10) (20 20) (30 30) . . . (100 100))
```

```
( setq l2 ( reverse l ) )
```

```
((100 100) . . . (30 30) (20 20) (10 10))
```

```
( setq lmod ( subst ( list 0 0 ) ( list 30 30 ) l ) )
```

```
((10 10) (20 20) (0 0) (40 40) . . . (100 100))
```

```
( setq l ( list 1 2 3 4 5 6 ) )
```

```
(1 2 3 4 5 6)
```

```
( foreach elem l ( setq suma ( + suma elem ) ) )
```

```
21
```

```
; La función mapcar se estudiará junto con las funciones monolínea APPLY y LAMBDA.
```