

FUNCIONES DE MANEJO DE CICLOS	OBSERVACIONES
<p>(repeat ent expr ...) <i>Evalúa cada expresión para todos los miembros de una lista, y devuelve el valor de la última expresión</i></p>	<ul style="list-style-type: none"> • El argumento ent debe ser, además de entero, positivo.
<p>(while texpr_prueba expr ...) <i>Evalúa una expresión de prueba y, si ésta no da como resultado nil, evalúa otras expresiones para volver de nuevo a la expresión de prueba</i></p>	<ul style="list-style-type: none"> • La función while continúa hasta que expr_prueba es nil. Entonces devuelve el valor más reciente de la última expresión.
<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.

EJEMPLOS DE FUNCIONES DE MANEJO DE CICLOS

Considérese la variable *l* valorada con la siguiente lista de puntos:

l = ((104.489 186.88 0.0) (238.529 185.446 0.0) (149.89 178.272 0.0) (179.436 177.554 0.0) (163.582 108.685 0.0) (104.489 161.772 0.0))

```
(setq i 0)
(command "pol")
(repeat (length l)
  (setq pto (nth i l))
  (command pto)
  (setq i (+ i 1))
)
(command "")
```

(I)

```
(command "pol")
(repeat (length l)
  (setq pto (car l))
  (command pto)
  (setq l (cdr l))
)
(command "")
```

(II)

```
(setq i 0)
(command "pol")
(while (setq pto (nth i l))
  (command pto)
  (setq i (+ i 1))
)
(command "")
```

(III)

```
(command "pol")
(while (setq pto (car l))
  (command pto)
  (setq l (cdr l))
)
(command "")
```

(IV)

```
(command "pol")
(foreach pto l
  (command pto)
)
(command "")
```

(V)

Cualquiera de las cinco variantes de ciclo dibuja la polilínea que pasa por todos los puntos de la lista *l*. Las variantes I y III necesitan un contador; las variantes II y IV no, pero dejan la variable *l* vacía al final del ciclo. En todo caso, es importante tener en cuenta que se ha separado el inicio de la polilínea en un primer Command (tras el cual el área de órdenes queda en estado de espera de dato) del envío de puntos (que se efectúa en el Command que está dentro del ciclo). Esto es perfectamente posible, pero hay que recordar que en ese segundo Command no hay que incorporar nombre de comando, porque ya hay uno activo (polilínea en este caso). Una vez finalizado el ciclo, aún queda la orden polilínea activa; por eso se incorpora el tercer Command con la cadena vacía, para cerrar la polilínea y finalizar la orden.