

FUNCIONES DE GESTIÓN DE SÍMBOLOS	OBSERVACIONES										
<p align="center"><b>(atom elemento)</b></p> <p><i>Verifica si un elemento es un átomo</i></p>	<ul style="list-style-type: none"> <li>Devuelve nil si elemento es una lista y devuelve T en caso contrario. Todos los elementos que no sean una lista se consideran átomos.</li> </ul>										
<p align="center"><b>(atoms-family formato [listsímb])</b></p> <p><i>Devuelve una lista de los símbolos actualmente definidos</i></p>	<ul style="list-style-type: none"> <li>El argumento formato es un valor entero 0 o 1. Si es 0, atoms-family devuelve los nombres de símbolos en una lista. Si es 1, la función devuelve los nombres de símbolos como una lista de cadenas. La función atoms-family busca una lista específica de nombres de símbolos si se suministra el argumento lista_símbolos. El argumento lista_símbolos es una lista de cadenas que especifican los nombres de símbolos. La función atoms-family devuelve una lista del tipo especificado por formato (símbolos o cadenas) y que contiene los nombres de los símbolos definidos. Devuelve nil para los que no están definidos.</li> </ul>										
<p align="center"><b>(boundp sím)</b></p> <p><i>Verifica si un valor está asociado a un símbolo</i></p>	<ul style="list-style-type: none"> <li>Devuelve T si sím tiene un valor asociado. Si no hay ningún valor asociado a sím (o se ha asociado a nil), boundp devuelve nil. Si sím es un símbolo no definido, se crea y se asocia a nil de forma automática.</li> </ul>										
<p align="center"><b>(null elemento)</b></p> <p><i>Comprueba si un elemento está definido como nil</i></p>	<ul style="list-style-type: none"> <li>Devuelve T si elemento está asociado a nil y devuelve nil en caso contrario.</li> </ul>										
<p align="center"><b>(numberp elemento)</b></p> <p><i>Comprueba si un elemento es un número real o un número entero</i></p>	<ul style="list-style-type: none"> <li>Devuelve T si elemento es un valor real o un entero y devuelve nil en caso contrario.</li> </ul>										
<p align="center"><b>(quote expr)</b></p> <p><i>Devuelve una expresión sin evaluarla</i></p>											
<p align="center"><b>(read [cadena])</b></p> <p><i>Devuelve la primera lista o átomo obtenidos de una cadena</i></p>	<ul style="list-style-type: none"> <li>El argumento cadena no puede contener espacios en blanco excepto en una lista o una cadena. La función read devuelve el argumento convertido en el tipo de datos correspondiente.</li> </ul>										
<p align="center"><b>(set sím expr)</b></p> <p><i>Define el valor de un nombre de símbolo entrecomillado como una expresión</i></p>	<ul style="list-style-type: none"> <li>Si set se utiliza con un nombre de símbolo no entrecomillado, puede asignar un valor nuevo a otro símbolo indirectamente.</li> </ul>										
<p align="center"><b>(setq sim1 expr1 [sim2 expr2] ...)</b></p> <p><i>Define el valor de uno o varios símbolos como una expresión asociada</i></p>	<ul style="list-style-type: none"> <li>Ésta es la función de asignación básica de AutoLISP. La función setq puede asignar varios símbolos en una llamada a la función, pero sólo devuelve la última expr.</li> </ul>										
<p align="center"><b>(type elemento)</b></p> <p><i>Devuelve el tipo de un elemento designado</i></p>	<ul style="list-style-type: none"> <li>Los tipos se devuelven como uno de los átomos mostrados en la tabla siguiente:</li> </ul> <table border="0" data-bbox="798 1568 1433 1630"> <tr> <td>ENAME</td> <td>FILE</td> <td>INT</td> <td>LIST</td> <td>PAGETBPICKSET</td> </tr> <tr> <td>REAL</td> <td>STR</td> <td>SUBR</td> <td>SYM</td> <td></td> </tr> </table>	ENAME	FILE	INT	LIST	PAGETBPICKSET	REAL	STR	SUBR	SYM	
ENAME	FILE	INT	LIST	PAGETBPICKSET							
REAL	STR	SUBR	SYM								

### EJEMPLOS:

*(QUOTE ( 20 30 40 ) ) ----- > DEVUELVE: (20 30 40)*  
*(QUOTE ( SETQ A 23 ) ) ----- > DEVUELVE (SETQ A 23)*  
*( READ "INTRODUCIR OPCION" ) ----- > DEVUELVE INTRODUCIR*  
**( Se pierde parte del literal, a partir del espacio en blanco )**