



Using the List Utility Library (listutils.tns)

Forest W. Arnold
9-24-2018

listutils is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

listutils is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with *listutils*. If not, see <https://www.gnu.org/licenses/>.

listutils is a library of functions for list manipulation

Library History:

Date Created: 22 May 2017

Functions:

The functions in the library are:

append()

The function `append(lst, elem)` is a convenience function that adds an element to the end of a list.

The element must be the same type of element as the other elements in the list.

Examples:

```
append({"red", "green", "blue"}, "gray") ▶ {"red", "green", "blue", "gray"}
```

insert()

The function `insert(lst, elem, index)` inserts an element into a list after the element at the input index. If the input index is less than 1, the element is added to the beginning of the list. If the input index is greater than the list dimension, the element is added to the end of the list.

The element must be the same type of element as the other elements in the list.

Examples:

```
insert({1,2,3},0,0) ▶ {0,1,2,3}
insert({1,2,3},4,3) ▶ {1,2,3,4}
insert({1,2,3,4,5},1.5,1) ▶ {1,1.5,2,3,4,5}
insert({1,2,3,4,5},2.5,2) ▶ {1,2,2.5,3,4,5}
insert({},1,0) ▶ {1}
insert({},1,2) ▶ {1}
```

qsort()

The function `qsort(lst)` sorts a list in ascending order.

The input argument 'lst' must be a LIST containing numeric or string values.

The implementation is based on the quicksort algorithm described in *The Design and Analysis of Computer Algorithms* by Aho, Hopcroft, and Ullman.

Examples:

```
lst:={54,26,93,17,77,31,44,55,20}  
qsort(lst) ▶ {17,20,26,31,44,54,55,77,93}
```

```
slist:={"red","green","gray","gray","grey","blue","violet","yellow"} ▶  
{ "red","green","gray","gray","grey","blue","violet","yellow"}  
qsort(slist) ▶ {"blue","gray","gray","green","grey","red","violet","yellow"}
```

remove()

The function `remove(lst, index)` removes the element at the position 'index'.

Examples:

```
remove({4},1) ▶ {}  
remove({4},0) ▶ {4}  
remove({4},2) ▶ {4}  
remove({8,7,6,5},1) ▶ {7,6,5}  
remove({8,7,6,5},4) ▶ {8,7,6}  
remove({8,7,6,5},3) ▶ {8,7,5}  
remove({8,7,6,5},2) ▶ {8,6,5}
```

reverse()

The function `reverse(lst)` reverses the order of elements in a list.
The input argument 'lst' must be a LIST containing numeric or string values.

Examples:

```
nlist:={1,2,3,4,5,6,7,8,9} ▶ {1,2,3,4,5,6,7,8,9}  
reverse(nlist) ▶ {9,8,7,6,5,4,3,2,1}  
reverse(qsort(lst)) ▶ {93,77,55,54,44,31,26,20,17}
```