

1	Introduction	6
2	List Suite	7
2.1	create list _	7
2.1.1	Syntax	7
2.1.2	Treatment of errors	7
2.1.3	Example	7
2.2	exchange item _ with item _ in list _	8
2.2.1	Syntax	8
2.2.2	Treatment of errors	8
2.2.3	Tip	8
2.2.4	Example	9
2.3	remove item _ of list _	9
2.3.1	Syntax	9
2.3.2	Treatment of errors	9
2.3.3	Example	10

Title Page

Contents



Page 1 of 22

Go Back

Full Screen

Close

Quit

2.4	remove head of list _	10
2.4.1	Syntax	10
2.4.2	Treatment of errors	10
2.4.3	Example	10
2.5	remove tail of list _	11
2.5.1	Syntax	11
2.5.2	Treatment of errors	11
2.5.3	Example	11
2.6	insert _ in list _ at index _	11
2.6.1	Syntax	12
2.6.2	Treatment of errors	12
2.6.3	Tip	12
2.6.4	Example	12
2.7	remove item _ from list _	13
2.7.1	Syntax	13
2.7.2	Treatment of errors	13
2.7.3	Tip	13

Title Page

Contents



Page 2 of 22

Go Back

Full Screen

Close

Quit

2.7.4	Example	13
2.8	merge list _ with list _	14
2.8.1	Syntax	14
2.8.2	Example	14
3	Computer Suite	15
3.1	log out	15
3.1.1	Example	15
3.2	restart	15
3.2.1	Example	15
3.3	shutdown	15
3.3.1	Example	16
3.4	sleep	16
3.4.1	Example	16
3.5	kill me	16
3.5.1	Example	16
4	Cursor Suite	17

Title Page

Contents



Page 3 of 22

Go Back

Full Screen

Close

Quit

4.1	mouse x	17
4.1.1	Syntax	17
4.1.2	Example	17
4.2	mouse y	17
4.2.1	Syntax	17
4.2.2	Example	18
4.3	mouse location	18
4.3.1	Syntax	18
4.3.2	Example	18
4.4	temp hide cursor	18
4.4.1	Example	18
4.5	mouse click	19
4.5.1	Example	19
4.6	double mouse click	19
4.6.1	Syntax	19
4.6.2	Example	19
4.7	move mouse	19

Title Page

Contents



Page 4 of 22

Go Back

Full Screen

Close

Quit

4.7.1	Treatment of errors	20
4.7.2	Example	20
5	Screen Suite	21
5.1	screen resolution	21
5.1.1	Example	21
6	Version History	22
6.1	Version 1.1 – March 2003	22
6.2	Version 1.0 – March 2003	22

Title Page

Contents



Page 5 of 22

Go Back

Full Screen

Close

Quit

1.

Introduction

This is a manual for the use of XTool. This document describes the syntax, the type of the arguments of each function, the type of the result, and what kind of result can be expected in when the user enters a bad argument. I will be glad to hear all your complaints, suggestions, remarks in order to help me to make it more functional and intuitive for the user. In order to contact me send an email at : xtool@tiscali.fr.

I would like to thank the applescript-implementors list for helping me and taking the time to explain me some very important things (and there is still a lot more to learn), all the staff of macscripter.net for making me enjoy applescript since I started using it, and Julifos for testing my OSAX.

[Title Page](#)[Contents](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 6 of 22](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

2. List Suite

2.1. create list _

Aim : create a list of a given size

2.1.1. Syntax

- argument1 : this is an argument of type *integer* that indicates the number of items in the list you want to create
- result : the result is of type *list*

2.1.2. Treatment of errors

- a zero value or a negative value for the argument will return an empty list
- a non integer argument will be round to the nearest integer

2.1.3. Example

```
set myList to create list 5
--> {0,0,0,0,0}
```

2.2. exchange item _ with item _ in list _

Aim : make an exchange between two items of a list

2.2.1. Syntax

- argument1 : this is an argument of type *integer*
- argument2 : this is an argument of type *integer*
- argument3 : this is an argument of type *list*
- result : the result is of type *list*

2.2.2. Treatment of errors

- if one of the argument is -1, it will refer to the last item of the list
- if one of the argument is not an integer but a real, the argument will be round to the nearest integer
- if one of the argument is invalid , no change will occur

2.2.3. Tip

- if one of the argument is -1, it will refer to the last item of the list
- if one of the argument is not an integer but a real, the argument will be round to the nearest integer

2.2.4. Example

```
set myList to {1,2,{5,6,7},"a","t"}  
set myList to exchange item 1 with item 2 in list myList  
--> {2,1,{5,6,7},"a","t"}
```

2.3. remove item _ of list _

Aim : remove an item from a list

2.3.1. Syntax

- argument1 : this is an argument of type *integer* that indicates the number of items in the list you want to create
- argument2 : this is an argument of type *list*
- result : the result is of type *list*

2.3.2. Treatment of errors

- if the first argument is -1, it will remove the last item of the list
- if the first argument is not an integer but a real, the argument will be round to the nearest integer
- if the first argument is invalid (greater than the number of items in the list), no change will occur

2.3.3. Example

```
set myList to {1,2,{5,6,7},"a","t"}  
set myList to remove item 3 of list myList  
--> {1,2,"a","t"}
```

2.4. remove head of list _

Aim : remove the first item of a list

2.4.1. Syntax

- argument1 : this is an argument of type *list*
- result : the result is of type *list*

2.4.2. Treatment of errors

- if the argument is not a valid list, an empty list will be returned

2.4.3. Example

```
set myList to {1,2,{5,6,7},"a","t"}  
set myList to remove head of list myList  
--> {2,{5,6,7},"a","t"}
```

2.5. remove tail of list _

Aim : remove the last item of a list

2.5.1. Syntax

- argument1 : this is an argument of type *list*
- result : the result is of type *list*

2.5.2. Treatment of errors

- if the argument is not a valid list, an empty list will be returned

2.5.3. Example

```
set myList to {1,2,{5,6,7},"a","t"}  
set myList to remove tail of list myList  
--> {1,2,{5,6,7},"a"}
```

2.6. insert _ in list _ at index _

Aim : insert a given item after a given index after a given list

2.6.1. Syntax

- argument1 : this is an argument of type *integer*
- argument2 : this is an argument of type *list*
- argument3 : this is an argument of type *integer*
- result : the result is of type *list*

2.6.2. Treatment of errors

- if one of the argument is -1, it will refer to the last item of the list
- if one of the argument is not an integer but a real, the argument will be round to the nearest integer
- if one of the argument is invalid , no change will occur

2.6.3. Tip

- if one of the argument is -1, it will refer to the last item of the list
- if one of the argument is not an integer but a real, the argument will be round to the nearest integer

2.6.4. Example

```
set myList to {1,2,{5,6,7},"a","t"}  
set myList to insert "bye" in list myList at index -1  
--> {1,2,{5,6,7},"a","t","bye"}
```

2.7. remove item _ from list _

Aim : remove an item at a given index from a given list

2.7.1. Syntax

- argument1 : this is an argument of type *integer*
- argument2 : this is an argument of type *list*
- result : the result is of type *list*

2.7.2. Treatment of errors

- if the first argument is not an integer but a real, the argument will be round to the nearest integer
- if one of the argument is invalid , no change will occur

2.7.3. Tip

- if one of the argument is -1, it will refer to the last item of the list
- if one of the argument is not an integer but a real, the argument will be round to the nearest integer

2.7.4. Example

```
set myList to {1,2,{5,6,7},"a","t"}
```

```
set myList to remove item 2 from list myList
--> {1,{5,6,7},"a","t"}
```

2.8. merge list _ with list _

Aim : merge two lists into a single list

2.8.1. Syntax

- argument1 : this is an argument of type *list*
- argument2 : this is an argument of type *list*
- result : the result is of type *list*

2.8.2. Example

```
set myList to {1,2,{5,6,7},"a","t"}
set arg2 to "nice"
set myList to merge list myList with list arg2
--> {1,2,{5,6,7},"a","t","nice"}
```

3. Computer Suite

3.1. log out

Aim : log out the current user (asking to save unsaved document)

3.1.1. Example

```
log out
```

3.2. restart

Aim : restart the computer

3.2.1. Example

```
restart
```

3.3. shutdown

Aim : shut down the computer

Title Page

Contents



Page **15** of **22**

Go Back

Full Screen

Close

Quit

3.3.1. Example

`shutdown`

3.4. sleep

Aim : put the computer to sleep

3.4.1. Example

`sleep`

3.5. kill me

Aim : kill the application that is running this command

3.5.1. Example

`kill me`

[Title Page](#)[Contents](#)

Page 16 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

4.

Cursor Suite

4.1. mouse x

Aim : get the x-coordinate of the mouse

4.1.1. Syntax

— the result is an integer

4.1.2. Example

```
mouse x  
--> 226
```

4.2. mouse y

Aim : get the y-coordinate of the mouse

4.2.1. Syntax

— the result is an integer

[Title Page](#)[Contents](#)

Page 17 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

4.2.2. Example

```
mouse y
--> 125
```

4.3. mouse location

Aim : get the x and y coordinate of the mouse

4.3.1. Syntax

— the result is a list of two integers (a point)

4.3.2. Example

```
mouse location
--> {226,125}
```

4.4. temp hide cursor

Aim : hide the cursor until the user move the mouse

4.4.1. Example

```
temp hide cursor
```

[Title Page](#)[Contents](#)

Page 18 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

4.5. mouse click

Aim : emulates a simple mouse click

4.5.1. Example

```
mouse click
```

4.6. double mouse click

Aim : emulates a double mouse click

4.6.1. Syntax

— the argument is a list of two items

4.6.2. Example

```
double mouse click
```

4.7. move mouse

Aim : move the mouse to the specified coordinates

[Title Page](#)[Contents](#)

Page 19 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

4.7.1. Treatment of errors

- giving a null list will put the mouse cursor at the top right of the screen
- giving a list of one argument will consider that the second argument is 0
- any additional item will be ignored

4.7.2. Example

```
move mouse {40,40}
```

[Title Page](#)[Contents](#)

Page 20 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

5. Screen Suite

5.1. screen resolution

Aim : get the bounds of your main screen

5.1.1. Example

```
screen resolution  
-> {1024,768}
```

[Title Page](#)[Contents](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)

Page 21 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

6.

Version History

6.1. Version 1.1 – March 2003

- three new suites have been added (Cursor Suite, Computer Suite, Screen Suite)
- the syntax in the 'list suite' as been modified so that every commans as the same type of syntax.

6.2. Version 1.0 – March 2003

- initial release, I'm waiting for bugs, suggestions, remarks!

[Title Page](#)[Contents](#)

Page 22 of 22

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)