

# assert



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## assert

### assert (V [,MESSAGE])

Issues an “assertion failed!” error when its argument V is nil .

Copyright © <Dates> by <Authors>. All Rights Reserved.

## call

### **call (FUNC, ARG [,MODE [, ERRHANDLER]])**

Calls FUNC with arguments in table ARG. FUNC results is returned by call . Error is propagated.MODE “x” protects the call; error is not propagated,nil is returned to signal the error.ERRHANDLERtemporarily sets \_ERRORMESSAGE. A nil disables the error handler

Copyright © <Dates> by <Authors>. All Rights Reserved.

## collectgarbage

### collectgarbage ([LIMIT])

Sets garbage-collection threshold (Kbytes). Default is zero. A garbage collection cycle may be run if the new threshold is smaller.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## copytagmethods

### copytagmethods (TAGTO, TAGFROM)

Copies all tag methods; returns TAGTO.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **doflle**

### **doflle (FILENAME)**

Opens FILENAME and executes it as a Lua chunk. Auto-detects and executes pre-compiled chunks. Defaults to stdin. Returns nil if error, a non-nil value if the chunk returns no values, or the values returned by the chunk. Error if FILENAME is non-string.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dostring

### **dostring (STRING[, CHUNKNAME])**

Executes STRING as a Lua chunk. Returns nil if error, a non-nil value if the chunk returns no values, or the values returned by the chunk. CHUNKNAME is used in error messages and debug information.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **error**

### **error (MESSAGE)**

Calls error handler, then terminates the last protected function called (in C: lua\_callfunction, lua\_dofile, lua\_dostring, lua\_dobuffer, or; in Lua: dofile, dostring, or call in protected mode). If MESSAGE is nil , then error is not called.Never returns.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# foreach



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## foreach

### foreach (TABLE, FUNC)

Executes FUNC over all elements of TABLE. FUNC is called repeatedly with the table's index and value pairs. Loop exits if FUNC returns any non-nil value, and this value is returned as the final value. Behavior is undefined if you change TABLE during traversal.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **foreachi**

### **foreachi (TABLE, FUNC)**

Executes FUNC over numerical indices of TABLE. FUNC is called repeatedly with the index and respective value as arguments, in sequential order, from 1 to getn(table). Loop exits if FUNC returns any non-nil value, and this value is returned as the final value.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# getglobal



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## getglobal

### getglobal (NAME)

Gets value of a global variable, or calls a tag method for “getglobal”. NAME does not need to be a syntactically valid variable name.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getn

### **getn (TABLE**

Returns the “size” of TABLE, seen as a list. If TABLE has an n field with a numeric value, this value is used. Otherwise, “size” is the largest numerical index with a non-nil value in the table.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## gettagmethod

### **gettagmethod (TAG, EVENT)**

Returns the current tag method for a given pair (tag, event). Cannot be used to get a tag method for the “gc” event. (Use C API call for “gc”.)

Copyright © <Dates> by <Authors>. All Rights Reserved.

# globals



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## globals

### globals ([TABLE])

Returns the current table of globals. If TABLE is given, then it also sets TABLE as the table of globals.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **newtag**

### **newtag ()**

Returns a new tag.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## next

### **next (TABLE, [INDEX])**

Traverse all fields of a table. Returns the next index of the table and its associated value. If INDEX is nil , returns the flrst index of the table and its associated value. When called with the last index, or with nil in an empty table, next returns nil . INDEX defaults to nil . next only considers fields with non-nil values. Enumeration order is not specifled, even for numeric indices. (For numeric order, use a numerical for or the function foreach). next is undeflned if you change the table during the traversal.

# print



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## print

### print (E1, E2,...)

Prints arguments using strings returned by tostring. Not intended for formatted output; mainly for quickies, for instance in debugging.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **rawget**

### **rawget (TABLE, INDEX)**

Gets the real value of table[index],without invoking any tagmethod. TABLEmust be a table, and INDEX is non-nil .

Copyright © <Dates> by <Authors>. All Rights Reserved.

## rawset

### **rawset (TABLE, INDEX, VALUE)**

Sets the real value of table[index] to VALUE, without invoking any tag method. TABLE, INDEX, and VALUE must be valid..

Copyright © <Dates> by <Authors>. All Rights Reserved.

# setglobal



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## setglobal

### **setglobal (NAME, VALUE)**

Sets NAME global variable to VALUE, or calls a tag method for “setglobal”.NAME need not be syntactically valid.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## settag

### **settag (T, TAG)**

Sets the tag of a given table T. TAGmust be a value created with newtag. Returns the value of its flrst argument (the table T.) It is impossible to change the tag of a userdata from Lua (for safety.)

Copyright © <Dates> by <Authors>. All Rights Reserved.

## settagmethod

### **settagmethod (TAG, EVENT, NEWMETHOD)**

Sets a new tag method to the given pair (tag, event) and returns the old method. If NEWMETHOD is nil , then the default behavior is restored. Cannot be used for the “gc” event. (Use C API call for “gc”).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## sort



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## sort

### **sort (TABLE [, COMP])**

Sorts TABLE in a given order, in-place, from table[1] to table[n], where n is the result of getn(table). If COMP is given, it must receive two table elements, and returns true (not nil) when the first is less than the second. COMP defaults to the operator <. The sort algorithm is not stable (quicksort is the current algorithm.)

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tag**

### **tag (V)**

Tests the tag of a value V; returns its tag (a number).

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **tonumber**



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## **tonumber**

### **tonumber (E [, BASE])**

Converts E to a number using optional BASE. Returns nil if unsuccessful. BASE can be an integer from 2 to 36 denoting [0-9A-Z]. In decimal, a fractional part and an exponent may be included. In other bases, only unsigned integers are accepted.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tostring**

### **tostring (E)**

Converts E to a string in a reasonable format. See also [format](#).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tinsert**



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## **tinsert**

### **tinsert (TABLE [, POS] , VALUE)**

Inserts element VALUE at table position POS, shifting to open space, if necessary. Default value for POS is getn(table)+1 (insertion at the end of the table, or append.) Table “size” (n field) is then updated.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tremove**

### **tremove (TABLE [, POS])**

Removes from TABLE the element at position POS, shifting to close the space, if necessary. Returns the value of the removed element. Default value for POS is the result of getn(table) (removal of the last element.) Table “size” (n field) is then updated.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# type



QS Informatica - Manuale LUA » Lua Standard Library » Basic Function Library »

## type

### type (V)

Test the type of a value; returns a string, one of: "nil", "number", "string", "table", "function", and "userdata".

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strbyte**

### **strbyte (S [, I])**

Returns internal numerical code of the I-th character of S. Default of I is 1, and may be negative. Not portable.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strchar

### strchar (I1, I2,...)

Receives 0 or more integers. Returns a string with length equal to the number of arguments, wherein each character has the internal numerical code equal to its corresponding argument. Not portable.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# strflnd



QS Informatica - Manuale LUA » Lua Standard Library » String Manipulation Library »

## strflnd

**strflnd (S, PATTERN [, INIT [, PLAIN]])**

Prints MESSAGE to stderr. Function may be reassigned.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# strlen



QS Informatica - Manuale LUA » Lua Standard Library » String Manipulation Library »

## strlen

### strlen (S)

Returns the length of S. An empty string has length 0. Any 8-bit character is counted, including embedded zeros.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **strlower**



QS Informatica - Manuale LUA » Lua Standard Library » String Manipulation Library »

## **strlower**

### **strlower (S)**

Returns a copy of S with all upper case letters changed to lower case, according to the current locale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strrep**

### **strrep (S, N)**

Returns a string that is the concatenation of N copies of the string S.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strsub



QS Informatica - Manuale LUA » Lua Standard Library » String Manipulation Library »

## strsub

### strsub (S, I [, J])

Returns another string, which is a substring of S, starting at I and running until J; I and J may be negative. The default value of J is -1 (the end.) strsub(s,1,j) returns a prefix; strsub(s, -i) returns a suffix.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strupper**

### **strupper (S)**

Returns a copy of S with all lower case letters changed to upper case, according to the current locale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## format

### **format (FORMATSTRING, E1, E2,...)**

Returns a formatted version of E1, E2,...following the description given in FORMATSTRING. Most printf rules are followed. Options/modifiers \*, l, L, n, p, and h are not supported. q formats a string suitable to be safely read back by the Lua interpreter. All double quotes, newlines and backslashes are correctly escaped. Escape sequences are recognized. Conversions specifying the n-th argument uses the sequence %d\$, where d is a decimal digit in the range [1-9], giving the position of the argument. E.g. For instance, %2\$d converts the second argument. The options c, d, E, e, f, g, G, i, o, u, X, and x all expect a number as argument, whereas q and s expect a string. The \* modifier can be simulated by building the appropriate format string. Neither the format string nor %s string values can contain embedded zeros. %q handles string values with embedded zeros

# gsub



QS Informatica - Manuale LUA » Lua Standard Library » String Manipulation Library »

## gsub

### gsub (S, PAT, REPL [, N])

Returns a copy of S in which all occurrences of the pattern PAT have been replaced by a replacement string specified by REPL. The second value returned is the total number of substitutions made. See §28 for the pattern matching format for PAT. The optional parameter N limits the maximum number of substitutions. For instance, when N is 1 only the first occurrence of PAT is replaced.

If REPL is a string, then its value is used for replacement. Any sequence of the form %n with n between 1 and 9 stands for the value of the n-th captured substring, which will be substituted in. If REPL is a function, then REPL is called every time a match occurs, with all captured substrings passed as arguments, in order. If the value returned by this function is a string, then it is used as the replacement string; otherwise, the replacement string is an empty string.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **math\_func**

### **math\_func (N)**

General mathematical functions accept a single number N as the argument.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## trig\_func

### trig\_func (DEG)

Trigonometrical functions accept an angle DEG expressed in degrees, not radians. The functions deg and rad can be used to convert between radians and degrees. (deg =rad \* 180/ $\pi$ )

abs	acos	asin	atan
cos	deg	exp	fioor
max	min	mod	rad
tan	frexp	ldexp	random

# **max**



QS Informatica - Manuale LUA » Lua Standard Library » Mathematical Function Library »

## **max**

### **max (N,...)**

Returns the maximum value of its numeric arguments.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# min



QS Informatica - Manuale LUA » Lua Standard Library » Mathematical Function Library »

## min

### min (N,...)

Returns the minimum value of its numeric arguments.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## random

### **random([N [,M]])**

Interfaces to the ANSI C function rand. Its pseudo-random properties is implementation-dependent; determined by the compiler, operating system and platform. When called without arguments, returns a pseudo-random real number in the range [0,1). When called with a number n, returns a pseudo-random integer in the range [1,n]. When called with two arguments, n and m, returns a pseudo-random integer in the range [n,m].

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **randomseed**

### **randomseed (SEED)**

Interfaces to the ANSI C function srand. Accepts one integer argument that is used to initialize the implementation-dependent random number generator.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## openfile

### **openfile (FILENAME,MODE)**

Opens a file in the mode specified in the string MODE. Broadly equivalent to fopen in standard C. Returns a new file handle, or, in case of errors, nil plus a string describing the error. Does not modify either \_INPUT or \_OUTPUT. The MODE string can contain: append update mode (previous data is preserved, append only at the end of file)

- **r**
  - read mode
- **r+**
  - update mode (all previous data preserved)
- **w**
  - write mode
- **w+**
  - update mode (all previous data erased)
- **a**
  - append mode
- **a+**
  - append update mode (previous data is preserved, append only at the end of file)
- **b**
  - binary mode

## **closeflle**



QS Informatica - Manuale LUA » Lua Standard Library » Input/Output Library »

## **closeflle**

### **closeflle (HANDLE)**

Closes the given flle. Does not modify either \_INPUT or \_OUTPUT.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## readfrom

### readfrom([FILENAME])

When called with a file name, it opens the named file, sets \_INPUT with its handle and returns this value. Does not close the current input file. When called without parameters, it closes the \_INPUT file, and restores stdin as the value of \_INPUT. On error, it returns nil , plus a string describing the error. If FILENAME starts with a | (pipe character), then input is piped via the C function popen. Not all systems implement pipes.

## writeto

### writeto (FILENAME)

With a file name, it opens the named file, sets \_OUTPUT with its handle and returns this value. Does not close the current output file. An existing file will be completely erased. Without parameters, it closes the \_OUTPUT file, and restores stdout as the value of \_OUTPUT. On error, it returns nil , plus a string describing the error. If FILENAME starts with a | (pipe character), then output is piped via the C function popen. Not all systems implement pipes.

## appendto

### appendto (FILENAME)

Opens FILENAME in append mode and sets \_OUTPUT with its handle. On error, it returns nil , plus a string describing the error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# remove



QS Informatica - Manuale LUA » Lua Standard Library » Input/Output Library »

## remove

### remove (FILENAME)

Deletes the file with the given name. On error, it returns nil , plus a string describing the error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# rename



QS Informatica - Manuale LUA » Lua Standard Library » Input/Output Library »

## rename

### rename(NAME1, NAME2)

Renames file named NAME1 to NAME2. On error, it returns nil , plus a string describing the error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **fiush**

### **fiush ([FILEHANDLE])**

Saves any written data to the given file. If FILEHANDLE is not specified, then all open files are flushed. On error, it returns nil , plus a string describing the error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## seek

### **seek (FILEHANDLE [,WHENCE] [,OFFSET])**

Sets and gets the file position (bytes), to the position given by OFFSET from a base specified by the string WHENCE, as follows:

- **set**
- base is position 0 (beginning of the file)
- **cur**
- base is current position and base is end of file If successful
- **seek**
- returns the final file position, measured in bytes from the beginning of the file.

On error, it returns nil , plus a string describing the error. Default for WHENCE is cur; for OFFSET is 0.

seek(file) returns the current file position, without changing it.

seek(file, "set") sets the position to the beginning (returns 0.)

seek(file, "end") sets the position to the end (returns its size.)

## **tmpname**

### **tmpname ()**

Returns a string with a file name that can safely be used as a temporary file. Must be explicitly opened before use; removed when not needed.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# read



QS Informatica - Manuale LUA » Lua Standard Library » Input/Output Library »

## read

### read ([FILEHANDLE,] [FORMAT1,...])

Reads file \_INPUT (or filehandle, if given) according to formats. For each format, a string (or a number) is read, or nil if it cannot read data with that format. The available formats are:

- **\*n**
- reads a number, and returns a number \*\*
- **\*l**
- reads the next line (skipping the end of line), or nil on EOF (default behaviour)
- **\*a**
- reads the whole file, starting at the current position. On EOF, it returns an empty string
- **\*w**
- reads the next word (maximal sequence of non-whitespace characters), skipping spaces if necessary, or nil on EOF.
- **number**
- reads a string with up to that number of characters, or nil on EOF

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **write**



QS Informatica - Manuale LUA » Lua Standard Library » Input/Output Library »

### **write**

#### **write ([FILEHANDLE, ] VALUE1,...)**

Writes the value of each argument to file \_OUTPUT (or filehandle, if given.) Must be strings or numbers, otherwise use tostring or format first. On error, it returns nil , plus a string describing the error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# clock



QS Informatica - Manuale LUA » Lua Standard Library » System Facilities »

## clock

### clock ()

Returns an approximation of CPU time (in sec) used by the program.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **date**



QS Informatica - Manuale LUA » Lua Standard Library » System Facilities »

## **date**

### **date ([FORMAT])**

Returns a string containing date and time formatted according to the given FORMAT, following the rules of the ANSI C function strftime. By default, it returns a reasonable date and time representation that depends on the host system and on the current locale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# execute



QS Informatica - Manuale LUA » Lua Standard Library » System Facilities »

## execute

### execute (COMMAND)

Equivalent to the C function system. It passes COMMAND to be executed by an operating system shell. Returns a system-dependent status code.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# exit



QS Informatica - Manuale LUA » Lua Standard Library » System Facilities »

## exit

### exit ([CODE])

Calls the C function exit, with an optional CODE, to terminate the program. Default value is the success code.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getenv

### getenv (VARNAME)

Returns the value of the process environment variable VARNAME, or nil if the variable is not defined.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setlocale

### setlocale (LOCALE [, CATEGORY])

Interfaces to the ANSI C function setlocale. LOCALE is a string specifying a locale; CATEGORY is an optional string describing which category to change: "all" (the default), "collate", "ctype", "monetary", "numeric", or "time". Returns the name of the new locale, or nil if the request is invalid.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# strlen



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## strlen

### strlen(str)

Ritorna la lunghezza di una stringa.

#### Esempio 1. Esempio strlen()

```
n = strlen('prova') - 5 prova
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strcmp

**strcmp(str1, str2, [numeroCaratteri] )**

Confronta 2 stringhe, considerando uguali le maiuscole e minuscole. Ritorna 0 se sono uguali, un numero minore di 0 se la prima e' minore della seconda, maggiore di 0 altrimenti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strstr



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

### strstr

#### strstr(str, trova)

Cerca la stringa <trova> nella stringa <str>. Se la trova ritorna la posizione (parte da 0), altrimenti ritorna nil. Considera diverse le maiuscole/minuscole. strstr effettua una ricerca che ignora le maiuscole/minuscole.

#### Esempio 2. Esempio strstr()

```
print(strstr('abc123', '123')) -          ritorna 3
print(strstr('abc123', 'ABC')) -          ritorna nil
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strstri**

### **strstri(str, trova)**

Come strstr ma effettua la ricerca non considerando le minuscole/maiuscole.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **substr**



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## **substr**

**substr(str, indice, [numeroCaratteri] )**

Estrae e ritorna dalla stringa str una sottostringa partendo dall'indice <indice> (parte da 0).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strlast



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## strlast

**strlast(str, [numeroCaratteri] )**

Ritorna gli ultimi <numeroCaratteri> caratteri della stringa (default 1).

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **strupr**



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## **strupr**

### **strupr(str)**

Ritorna la stringa <str> convertita in maiuscolo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strlwr**



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

### **strlwr**

#### **strlwr(str)**

Ritorna la stringa <str> convertita in minuscolo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strspn



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## strspn

### strspn(str, delim)

Ritorna il numero di caratteri iniziali di <str> che sono presenti nella stringa <delim>.

### Esempio 3. Esempio strspn()

```
local len = strspn('BBB223344', 'AB') - len vale 3          (numero  
di 'B' iniziali)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strcspn

### **strcspn(str, delim)**

Ritorna il numero di caratteri iniziali di <str> che non sono presenti nella stringa <delim>. Si tratta dell'opposto della funzione *strspn*.

#### **Esempio 4. Esempio strcspn()**

local len2 = strcspn('BBB223344', '4') - len2 vale di 'B23' iniziali)	7 (numero
--	-----------

## strsubst



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## strsubst

### **strsubst(str, find, repl)**

Cerca tutte le occorrenze della stringa <find> nella stringa <str> sostituendole con la stringa <repl>. Ritorna la nuova stringa. Case sensitive.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## strtrans



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## strtrans

### strtrans(str, find, repl)

Sostituisce i caratteri di <find> con quelli di <repl>. Cerca ogni carattere presente in <find> nella stringa <str>; se trovato viene sostituito con il corrispondente carattere di <repl>; se in <repl> non c'e' un carattere corrispondente, il carattere viene eliminato.

#### Esempio 5. Esempio strtrans()

```
a = strtrans("123_456", "34", "AB") - a           diventa "12A_B56" a
= strtrans("a b c-d", " -", "") -
      a diventa "abcd" a = strtrans("pippo", "op",
      "k") - a diventa "ik"
```

## strsxdx

### strsxdx(str, strfind)

Cerca la stringa <strfind> nella stringa <str>. Se trovata ritorna 2 valori: la parte sinistra e la parte destra; se non trovata ritorna nil, nil.

#### Esempio 6. Esempio strsxdx()

```
local sx, dx = strsxdx( 'abcdef', 'bc' ) - sx='a', dx='def'
```

## strfill

### **strfill(larg, strbase)**

Crea una stringa larga <larg> ripetendo la stringa <strbase> fino ad occupare la larghezza data.

#### **Esempio 7. Esempio strfill()**

<code>strfill(5, 'ab')</code> - ritorna	"ababa"
---	---------

## strrev



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

### strrev

#### strrev(str)

Inverte i caratteri di una stringa.

#### Esempio 9. Esempio strrev()

```
a = strrev('abc') - a = 'cba'
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# sprintf



QS Informatica - Manuale LUA » LUA - QS » Stringhe »

## sprintf

### sprintf(fmt, argomenti)

Formatta una stringa, sintassi come in C. Esiste anche la funzione lua format che e' identica.

#### Esempio 10. Esempio sprintf()

```
local prova = sprintf("%02d", 2+2) - ora           prova vale "04"
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## base64enc

### base64enc(str)

- str
- (tipo stringa) stringa in input

Trasforma una stringa in base 64.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## base64dec

### base64dec(str)

- str
- (tipo stringa) stringa in base64 in input

Decodifica una stringa in base 64.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **strquote**

### **strquote(str)**

Ritorna la stringa <str> con i caratteri di escape lua (apici, virgolette, ...) preceduti da backslash.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# match



QS Informatica - Manuale LUA » LUA - QS » Espressioni Regolari (RegExp) »

## match

### <regex>:match(str, pos)

Esegue un match su tutta la stringa e ritorna la posizione del primo match. Se viene fornito <pos> la ricerca parte dalla posizione <pos>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## matchall

**<regex>:matchall(str, pos) -> {s1, s2, ...}**

Esegue un match su tutta la stringa e ritorna i blocchi trovati Se viene fornito <pos> la ricerca parte dalla posizione <pos>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## split



QS Informatica - Manuale LUA » LUA - QS » Espressioni Regolari (RegExp) »

### split

<regex>:split(str) -> {s1, s2, ...}

Come strstr ma effettua la ricerca non considerando le minuscole/maiuscole.

#### Esempio 11. Esempio <regex>:split()

```
local re = RegExp('[ \\\t]*,[ \\\t]*') local str = "prova ,  
prova2,prova3" local lst = re:split(str) -  
    ora  
    lst={[1]='prova',[2]='prova2',[3]='prova3'}
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# get



QS Informatica - Manuale LUA » LUA - QS » Espressioni Regolari (RegExp) »

## get

### <regex>:get(pos)

Dopo avere effettuato un'operazione di match, restituisce il segmento <pos> della stringa. Se <pos> vale 0 viene ritornata l'intera stringa trovata.

#### Esempio 12. Esempio <regex>:get()

```
local re = RegExp('([0-9]+),([0-9]+)') if
  (re:match('xx12,41414xx')) then print (re:get(0)) - scrive
    12,41414 print (re:get(1)) - scrive 12 print (re:get(2)) -
scrive
  41414 end
```

## **lualist.cat**

### **lualist.cat(lista1 [, lista2 ....])**

Concatena tutte le liste e le variabili passate come argomenti, ritornando una nuova lista.

#### **Esempio 14. Esempio lualist.cat()**

<code>lualist.cat( {'a', 'b', 'c'}, {'d', 'e'} ) -</code>	<i>ritorna</i>
<code>{'a', 'b', 'c', 'd', 'e'}</code>	

## **lualist.keys**



QS Informatica - Manuale LUA » LUA - QS » Manipolazione Liste »

## **lualist.keys**

### **lualist.keys(lista)**

Ritorna una lista che ha solo le chiavi di <lista>.

#### **Esempio 15. Esempio lualist.keys()**

<code>lualist.keys{nome=1, valore='xx'} -</code>	<i>ritorna {'nome', 'valore'}</i>
--	-----------------------------------

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lualist.values

### lualist.values(lista)

Ritorna una lista che ha solo i valori <lista>.

#### Esempio 16. Esempio lualist.values()

lualist.values{nome=1, valore='xx'} -	<i>ritorna {1, 'xx'}</i>
---------------------------------------	--------------------------

## lualist.ilookup

### lualist.ilookup(lista, chiave)

Cerca un elemento della lista <lista> con chiave <chiave>. Il confronto viene effettuato ignorando le maiuscole/minuscole.

#### Esempio 17. Esempio lualist.ilookup()

```
lst = {Rosso=1, Giallo=2}           print(lualist.ilookup(lst,
'ROSSO')) -  
          ritorna 1 print(lualist.ilookup(lst,
'ros-so')) - ritorna nil
```

## **lualist.join**

### **lualist.join(lista [, separatore [, pre [, post]]])**

Ritorna una stringa con tutti gli elementi di <lista> concatenati.<separatore> viene inserito tra gli elementi, default stringa vuota. <pre> e' inserito all'inizio, default vuoto. <post> e' inserito alla fine, default vuoto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **lualist.find**

### **lualist.find(lista, valore)**

Cerca <valore> in <lista>. Se lo trova ritorna l'indice, altrimenti nil.

#### **Esempio 18. Esempio lualist.find()**

```
local pos = lualist.find( {'verde', 'giallo',  
'giallo'} ) - pos vale  
2
```

## lualist.indexof

**lualist.indexof(lista, searchitem, [func] )**

- **lista**
- (tipo lista) lista in input
- **searchitem**
- (tipo qualsiasi) elemento da cercare
- **func**
- (tipo funzione) Funzione di test

Cerca un elemento in una lista (sequenza). Restituisce l'indice dell'elemento (se trovato) oppure nil.

### Esempio 19. Esempio lualist.indexof

```
lst = {"gennaio", "febbraio", "marzo", "aprile"}           function
isPluto(el) return el == "pluto" end function
    iniziacon(str, pref) return 0 == stricmp(str, pref,
strlen(pref))
        end print(lualist.indexof(lst, "Dic"), "\n") -- restituisce
NIL
    print(lualist.indexof(lst, "MARZO"), "\n") -- 3
    print(lualist.indexof(lst, "FEB", iniziacon), "\n") --
restituisce 2
```

## lualist.reverse

### lualist.reverse(lista)

Ritorna una lista con tutti gli elementi di <lista> in posizione invertita.

#### Esempio 20. Esempio lualist.reverse()

```
local lst = lualist.reverse{1,2,3} - lst = {3, 2, 1}
```

## **lualist.filter**

### **lualist.filter(lista, funzBool)**

Ritorna una lista con tutti gli elementi di <lista> che soddisfano la funzione di test <funzBool>. <funzBool> riceve un argomento (l'elemento) e ritorna 1 se ok 0 se e' da eliminare.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lualist.map

### **lualist.map(funz, lista)**

Chiama per ogni elemento di <lista> la funzione funz(elem). Ritorna una lista con i valori di ritorno delle chiamate.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lualist.merge

### **lualist.merge(list1 [,list2 ...])**

Crea una lista che contiene tutti gli elementi di list1, di list2, ecc... Nel caso di chiavi ripetute vengono considerati solo gli ultimi elementi.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lualist.select

### lualist.select(list, campo)

Estrae una lista degli elementi <campo> della lista di liste <list>.

#### Esempio 21. Esempio lualist.select()

```
lst = { { altezza = 2, larg=3}, { altezza = 4 , larg=5} }  
lst2 = (lualist.select(lst, "altezza")) - lst2 = {2,  
        4}
```

## **lualist.erase\_n**

### **lualist.erase\_n(lista, idx)**

Cancella l'elemento con indice <idx> dalla lista <lista> scala tutti gli indici verso il basso. Se la lista contiene molti elementi puo' essere lento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## issequence

### issequence(lista)

Ritorna 1 se la <lista> e' una sequenza (le chiavi sono 1,2,3...n); ritorna nil se <lista> e' una tabella associativa o un altro tipo di oggetto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **catpath**



QS Informatica - Manuale LUA » LUA - QS » File Names »

## **catpath**

### **catpath(path, name) -> fullname**

Concatena path e name, mettendo tra i 2 la barra '\' se necessario.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## luafullpath

### luafullpath(basename)

Ritorna un filename composto dal path dell'interprete LUA + il file passato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## tempfilename

**tempfilename(tabOpzioni)**

- **tabOpzioni**
- (tipo tabella)

**Tabella 1. Formato della tabella tabOpzioni.**

Parametro	Tipo	Descrizione
dir	stringa	Directory (se omessa e' la dir temp di sistema)
ext	stringa	Estensione del file da creare (default .tmp)
make	booleano	Se 1 crea il file vuoto; altrimenti ritorna solo il nome
name	stringa	Nome file fisso, opzionale.
prefix	stringa	Prefisso nome file (default "tm")
makedir	booleano	Se true crea una dir invece di un file e restituisce il nome.

Ritorna un nome di file generato automaticamente da usarsi per file temporanei.

# fopen



QS Informatica - Manuale LUA » LUA - QS » Files »

## fopen

### **fopen(fileName, fileOpenMode) -> fileObj**

apre un file, come in C, ritorna un oggetto di tipo file fileName e' il nome del file da aprire. fileOpenMode : 'w':write, 'r':read, 'a':append, 'rb'=read binary

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getstdhandle



QS Informatica - Manuale LUA » LUA - QS » Files »

## getstdhandle

### **getstdhandle(idx) -> fileObj**

Ritorna un oggetto file che rappresenta stdin (idx = 0), stdout (idx=1) oppure stderr (idx=2). E' utile solo in applicazioni di tipo console. Il file restituito non deve essere chiuso.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## popen

### popen(comando, modo)

Come in c popen. Lancia un comando tramite una pipe anonima, ritorna un file. Se modo e' 'r' leggendo dal file si legge l'output del programma. Se comando inizia con il nome di un .EXE contenente spazi, se si e' in modalita' non console, e' necessario usare il formato 8+3, altrimenti non funziona (bug di Win).

#### Esempio 25. Esempio popen()

```
local fn = popen('dir /s /b', 'r') local str = fn:readLine()  
- legge la prima linea dell'output di  
  dir fn:close()
```

## close



QS Informatica - Manuale LUA » LUA - QS » Files »

## close

**<file>:close()**

Chiude il file.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# seek



QS Informatica - Manuale LUA » LUA - QS » Files »

## seek

**<file>:seek(pos)**

Riposiziona il puntatore al file nella posizione <pos>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## puts



QS Informatica - Manuale LUA » LUA - QS » Files »

### puts

```
<file>:puts(s, [s2] , [s3] )  
<file>:writeLine(s [, s2 [, s3]])
```

Scrive la stringa <s> sul file. Puts non aggiunge il CRLF finale, writeLine si.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## serialize



QS Informatica - Manuale LUA » LUA - QS » Files »

## serialize

**<file>:serialize(table, name)**

Salva sul file la table <table> in modo che con un dofile la table possa essere ricostruita. <name> e' il nome della variabile. Utilizzando come nome "return" si ottiene una istruzione di return "return {}" invece di "var={}".

### Esempio 26. Esempio serialize()

```
a = {1,2}; f:serialize(a, "var") -- scrive sul file "var={1,2}".
```

# eof



QS Informatica - Manuale LUA » LUA - QS » Files »

## eof

### <file>:eof()

Ritorna vero (1) se si e' alla fine del file, nil altrimenti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **flush**



QS Informatica - Manuale LUA » LUA - QS » Files »

## **flush**

### **<file>:flush()**

Esegue immediatamente le modifiche non ancora scritte.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **pclose**



QS Informatica - Manuale LUA » LUA - QS » Files »

## **pclose**

**<file>:pclose()**

Chiude una pipe anonima aperta con popen.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getcwd

### getcwd()

Ritorna il nome della dir corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **mkdir**



QS Informatica - Manuale LUA » LUA - QS » Directories »

## **mkdir**

### **mkdir(nomeDir)**

Crea una directory, anche a piu' livelli.Ritorna 1 se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# rmdir



QS Informatica - Manuale LUA » LUA - QS » Directories »

## rmdir

### **rmdir(nomeDir)**

Elimina una directory, che deve essere vuota. Ritorna 1 se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## chdir

**chdir(nomeDir)**

Cambia la directory corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## next



QS Informatica - Manuale LUA » LUA - QS » Directories »

### next

**<dirIterator>:next()**

Avanza al prossimo file

Copyright © <Dates> by <Authors>. All Rights Reserved.

# close



QS Informatica - Manuale LUA » LUA - QS » Directories »

## close

### <dirIterator>:close()

Libera la memoria e le risorse associate all'oggetto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dirscanner

### dirscanner(tabopzioni)

Scorre tutti gli elementi di una cartella, andando anche nelle sottocartelle, ed esegue una funzione per ogni elemento trovato. E' possibile restringere la ricerca utilizzando vincoli sul tipo, dimensioni, profondita' della scansione, ...

La funzione passata la foreach può ritornare un valore e interrompere la scansione

**Tabella 2. dirscanner(): Formato della Tabella.**

Parametro	Tipo	Descrizione
basedir	stringa	Directory di base (default ".")
foreach	funzione	Funzione da eseguire per ogni elemento trovato.
mode	intero	1=solo files, 2=solo directories, 3=files+directories
extension	stringa	Extensione (o estensioni da accettare) esempio: ".dwg" oppure ".bak .tmp"
norecurse	intero	Se 1 => solo 1 livello
size	tabella	Per i files, range della dimensione ({min=bytes, max=bytes})

### Esempio 28. Esempio dirscanner()

```
dirscanner{basedir="c:\\\", foreach=function(fn)
    print(fn, "\n") end, size={1000,2000}} -- stampa il nome
di tutti i files, con dimensione compresa tra 1000 e 2000
```

*bytes,*

*partendo -- dalla radice di C:, visitando anche le sottocartelle.*

Copyright © <Dates> by <Authors>. All Rights Reserved.

## alert



QS Informatica - Manuale LUA » LUA - QS » Windows »

### alert

**alert(msg1, [msg2] )**

- **msg1**
- (tipo stringa) messaggio
- **msg2**
- (tipo stringa) altre stringhe opzionali

Visualizza un message box, con i argomenti separati da spazi.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **clip.clear**

### **clip.clear()**

Cancella il contenuto della clipboard.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## makevariant

### makevariant(**tipo**)

- **tipo**
- (tipo stringa) tipo dell'oggetto da creare.

**Tabella 13. makevariant: tipo**

Valore	Descrizione
"bool.true"	valore boolean TRUE
"bool.false"	valore boolean FALSE
"argnull"	valore nullo DISP_E_PARAMNOTFOUND

Crea un valore di tipo variant (COM) di un determinato tipo.

## proprieta

**<obj>.proprieta = valore**

Imposta il valore di una proprieta'.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# call



QS Informatica - Manuale LUA » LUA - QS » Windows COM »

## call

**<obj>.call('nome\_funz'[ , argomenti])**

Chiama un metodo di un oggetto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# get



QS Informatica - Manuale LUA » LUA - QS » Windows COM »

## get

**<obj>:get('nome\_prop'[ , argomenti])**

Legge una proprieta', con argomenti opzionali.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## localtime

### localtime()

Ritorna la data/ora corrente in una tabella. Nomi dei campi: year, mon, day, hour, min, sec.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **bitxor**

### **bitxor(a, b [, ...])**

Esegue l'XOR tra gli argomenti (interi).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **bitand**

### **bitand(a, b [, ...])**

Esegue l'AND tra gli argomenti (interi).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **bitor**

### **bitor(a, b [, ...])**

Esegue l'OR tra gli argomenti (interi).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **dofile**

### **dofile(fileName)**

Include ed esegue un file LUA esterno. Algoritmo di ricerca del file <fileName>:

- Se il fileName e' completo (contiene almeno un carattere '/' oppure '\') viene usato solo quel percorso;
- Se fileName non include un path la ricerca avviene prima nella directory corrente poi nella directory restituita dalla funzione LuaSearchPath.

## **dofilemulti**

### **dofilemulti(pattern)**

Esegue un dofile per ogni file compreso in pattern. (es 'c:\\temp\\\*.lua').

Copyright © <Dates> by <Authors>. All Rights Reserved.

# require



QS Informatica - Manuale LUA » LUA - QS » Bit/Bytes »

## require

### require(file1, ...)

Include ed esegue i file LUA elencati, caricandoli una sola volta setta per ogni elemento included\_files[file] = 1.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **loadlib**

### **loadlib(file)**

Carica un modulo dll <file> BINARIO, <file> e' una dynamic link library (DLL) che per essere caricabile da LUA deve esportare un simbolo "LuaExtInit(lua\_state\*)".

Copyright © <Dates> by <Authors>. All Rights Reserved.

# globals



QS Informatica - Manuale LUA » LUA - QS » Bit/Bytes »

## globals

### globals() -> tab

Ritorna una tabella con tutti i simboli globali (variabili, funzioni).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## gc

### gc()

Esegue una garbage collection manuale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **assert**



QS Informatica - Manuale LUA » LUA - QS » Bit/Bytes »

## **assert**

### **assert(expression)**

Mostra un messaggio se expression e' falsa (vedi man lua)

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tagname**

### **tagname(var)**

Ritorna il nome del tipo con tag = <var> o se var e' un oggetto il nome della classe.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## beep

### beep([val])

Messaggio acustico standard del sistema. Passando '\*' oppure '!' oppure '?' partono altri suoni.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# delay



QS Informatica - Manuale LUA » LUA - QS » Bit/Bytes »

## delay

### delay(nSeconds)

Sospende il programma per un tempo di <nSeconds> secondi

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **abs**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## **abs**

### **abs(n)**

Valore assoluto di <n>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**sin**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## **sin**

### **sin(ang)**

Ritorna il seno dell'angolo <ang>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**COS****cos(ang)**

Ritorna il coseno dell'angolo <ang>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**tan**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

**tan**

**tan(ang)**

Ritorna la tangente dell'angolo <ang>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# atan



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## atan

### atan(num)

Ritorna l'arcotangente (in gradi) di <num>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**rad**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## **rad**

### **rad(ang)**

Ritorna <ang> convertito da gradi a radienti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# deg



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## deg

### deg(ang)

Conversione di un angolo da gradi a radianti e viceversa.

#### Esempio 37. Esempio deg()

```
local ang = rad(90) - ang = 1.57           (pi/2)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## floor



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## floor

### floor(val)

Tronca la parte decimale di un numero verso il basso.

#### Esempio 38. Esempio floor()

```
print(floor(2.0)) - 2 print(floor(2.8)) - 2  
print(floor(-2.1)) - (-2)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# ceil



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## ceil

### ceil(val)

Tronca la parte decimale di un numero verso l'alto

#### Esempio 39. Esempio ceil()

```
print(ceil(2.0)) -- 2 print(ceil(2.1)) -- 3  
print(ceil(-2.1)) -- (-2)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **sqrt**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## **sqrt**

### **sqrt(n)**

Ritorna la radice quadrata del numero <n>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**log**



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## **log**

### **log(n)**

Ritorna il logaritmo naturale del numero <n>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# log10



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## log10

### log10(*n*)

Ritorna il logaritmo deciamle del numero <*n*>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# mod



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## mod

### mod(a, b)

Ritorna il resto della divisione a/b.

#### Esempio 40. Esempio mod()

```
local resto = mod(5, 3) - resto = 2
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# max



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## max

**max(a, b, [c] )**

Ritorna il massimo di una serie di valori passati come argomenti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## min



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

### min

**min(a, b, [c] )**

Ritorna il minimo della serie di valori passati come argomenti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## modf



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

### modf

#### modf(num)

Restituisce la parte intera e decimale del numero <num>, ritorna 2 valori.

#### Esempio 41. Esempio modf()

```
local intera, decimale = modf(1.23) -- ora           intera=1 e  
decimale=0.23
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **random**

### **random(n)**

Ritorna un numero pseudocasuale compreso tra 0 e max.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## round



QS Informatica - Manuale LUA » LUA - QS » Matematiche »

## round

**round(num, [numDigits] )**

Restituisce il numero <num> arrotondato a <numDigits> decimali. Se <numDigits> e' omesso vale 0.

### Esempio 42. Esempio round()

```
print(round(2.15,1)) -- scrive 2.2          print(round(2.149,1)) --
scrive 2.2  print(round(-1.475, 2)) --
scrive -1.48
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## itoa

**itoa(*num*, [*base*] )**

Ritorna il numero <num>, intero, come stringa. Il parametro <base> indica la base di conversione.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getenv

### **getenv(nomeVar)**

Ritorna il valore di una variabile di ambiente nomeVar, oppure nil se non e' definita.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **setenv**

### **setenv(nomeVar, valore)**

Imposta il valore di una variabile di ambiente. La modifica vale solo nel processo corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **dofile**

### **dofile(fileName)**

Include ed esegue un file LUA esterno.

Algoritmo di ricerca del file <fileName>:

- Se il fileName e' completo (contiene almeno un carattere '/' oppure '\') viene usato solo quel percorso;
- Se fileName non include un path la ricerca avviene prima nella directory corrente poi nella directory restituita dalla funzione LuaSearchPath.

## **dofilemulti**

### **dofilemulti(pattern)**

Esegue un dofile per ogni file compreso in pattern. (es 'c:\\temp\\\*.lua')

Copyright © <Dates> by <Authors>. All Rights Reserved.

# require



QS Informatica - Manuale LUA » LUA - QS » Mix »

## require

### require(file1, ...)

Include ed esegue i file LUA elencati, caricandoli una sola volta setta per ogni elemento included\_files[file] = 1.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **loadlib**

### **loadlib(file)**

Carica un modulo dll <file> BINARIO, <file> e' una dynamic link library (DLL) che per essere caricabile da LUA deve esportare un simbolo "LuaExtInit(lua\_state\*)".

Copyright © <Dates> by <Authors>. All Rights Reserved.

# globals



QS Informatica - Manuale LUA » LUA - QS » Mix »

## globals

### globals() -> tab

Ritorna una tabella con tutti i simboli globali (variabili, funzioni).

Copyright © <Dates> by <Authors>. All Rights Reserved.

**gc****gc()**

Esegue una garbage collection manuale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **setgclimit**

### **setgclimit(num)**

Funzione che setta il limite di dati che possono entrare nel garbage collection. In ingresso vuole un numero maggiore di 0.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **assert**



QS Informatica - Manuale LUA » LUA - QS » Mix »

## **assert**

### **assert(expression)**

Mostra un messaggio se expression e' falsa (vedi man lua).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tagname**

### **tagname(var)**

Ritorna il nome del tipo con tag = <var> o se var e' un oggetto il nome della classe.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## beep

### beep([val])

Messaggio acustico standard del sistema. Passando '\*' oppure '!' oppure '?' partono altri suoni.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# delay



QS Informatica - Manuale LUA » LUA - QS » Mix »

## delay

### delay(nSeconds)

Sospende il programma per un tempo di <nSeconds> secondi.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **perror**

### **perror(str)**

Scrive la stringa <str> su standard error.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **print**



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## **print**

### **print(str1...)**

Scrive una stringa nella finestra di testo di AutoCAD.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# command



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## command

**command(comando, arg1, arg2, ...)**

Lancia un comando di AutoCAD. E' possibile passare piu' argomenti oppure una lista.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **usrbrk**

### **ritorno = usrbrk()**

Ritorna true se l'utente ha premuto "esc", altrimenti false.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **dwgopen**

### **dwgopen(nomeDisegno)**

Apre un disegno con sync-file-open.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## menucmd

**menucmd(str)**

Come in LISP.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## foreach\_block



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## foreach\_block

### foreach\_block(pattern-name, func)

Cerca i blocchi con nome simile a pattern-name ed esegue func(blocco) per ogni INSERT trovato. La ricerca degli insert avviene nello spazio corrente (modello o carta).

#### Esempio 45. Esempio foreach\_block()

```
foreach_block("TAB-*", function(e)      print("Trovato blocco: \n")
end)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## wcmatch

### wcmatch(stringa, pattern)

Esegue un pattern match tra stringa e pattern. Pattern puo' contenere \*=quals.car  
- [abc]=da 'a' a 'c'

#### Esempio 46. Esempio wcmatch()

```
wcmatch('prov123', 'prov*') -- ritorna 1
```

## getfiled

### getfiled{parametri}

Chiede all'utente di selezionare il nome di un file. parametri e' una tabella con il seguente formato:

**Tabella 15. getfiled(): Tabella Parametri**

Nome Campo	Tipo	Descrizione
title	stringa	Titolo del dialog box di richiesta
default	stringa	Filename di default
ext	stringa	Filtro estensione.
flags	numero	Opzioni: vedi tabella seguente, sommare i valori

**Tabella 16. getfiled(): Tabella valori "flags"**

Valore	Descrizione
1	Indicates a request for a new file to be created.
2	Disables the Type it button.
4	Enables the user to enter an arbitrary file-name extension.
8	Performs a library search for the file name entered.
16	Interprets the second argument to acedGetFileD as a path or directory name, not a file name.
32	Inhibits display of the alert box, which warns that a file exists when a new file of the same name is opened.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# xload



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## xload

### xload(filearx)

Carica un'applicazione arx. filearx e' il filename completo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## acad\_colordlg



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## acad\_colordlg

### acad\_colordlg([init\_color])

Dialog box di selezione colore di Autocad. Ritorna il colore selezionato o nil su cancel.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## acad\_truecolordlg



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## acad\_truecolordlg

**colore = acad\_truecolordlg([init\_color])**

Dialog box di selezione colore di Autocad, gestisce anche i truecolor. Ritorna il colore selezionato o nil su cancel.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## rgb

**color = rgb(RR, GG, BB)**

Funzione che trasforma la codifica RGB nel corrispondente valore intero. In ingresso vuole 3 valori interi (colori). Ritorna un numero.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **snvalid**

### **snvalid(testName)**

Ritorna vero (1) se il nome testName e' valido come symbol table entry, nil altrimenti.

## **textscr**

### **textscr()**

Attiva la finestra testo in AutoCAD.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **graphscr**

### **graphscr()**

Attiva la finestra grafica in AutoCAD.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **acad\_torgb**



QS Informatica - Manuale LUA » AutoCAD » Funzioni Globali »

## **acad\_torgb**

### **acad\_torgb(indicecol)**

Dato un colore di AutoCAD ne ritorna il valore rgb nel formato (32 bit)  
00rrggbb

Copyright © <Dates> by <Authors>. All Rights Reserved.

# getvar



QS Informatica - Manuale LUA » AutoCAD » Variabili »

## getvar

**getvar(nomeVar)**

Legge una variabile di AutoCAD e Ritorna il valore.

**Esempio 48. Esempio getvar()**

```
local cmdecho = getvar("CMDECHO")
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setvar



QS Informatica - Manuale LUA » AutoCAD » Variabili »

## setvar

**setvar(nomeVar, valoreVar)**

Imposta una variabile.

**Esempio 49. Esempio setvar()**

```
setvar("CMDECHO", 0)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setvarmulti



QS Informatica - Manuale LUA » AutoCAD » Variabili »

## setvarmulti

**tab = setvarmulti({tab})**

Setta contemporaneamente piu variabili di AutoCAD. In ingresso vuole una tabella nel formato: tab = {Var1, valore1, Var2, valore2,...}. Ritorna una tabella nello stesso formato con i valori precedenti.

**Esempio 50. Esempio setvarmulti()**

```
tab = setvarmulti({"CMDECHO", 0, "OSMODE", 1})
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## acad\_setdwgreadonly



QS Informatica - Manuale LUA » AutoCAD » Variabili »

## acad\_setdwgreadonly

**ritorno = acad\_setdwgreadonly()**

Funzione che setta il dwg in modalità di sola lettura. Ritorna true se l'operazione è andata a buon fine, altrimenti false.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## acad\_setdwgwriteable



QS Informatica - Manuale LUA » AutoCAD » Variabili »

### acad\_setdwgwriteable

**ritorno = acad\_setdwgwriteable()**

Funzione che setta il dwg in modalità di scrittura. Ritorna true se l'operazione è andata a buon fine, altrimenti false.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **getpoint**

### **getpoint(prompt, ptRiferimento)**

Chiede un punto, prompt e' il messaggio, ptRiferimento e' opzionale.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getcorner

### **getcorner(punto, prompt)**

Chiede l'angolo di un rettangolo. <punto> e' l'altro angolo. <prompt> e' opzionale

Copyright © <Dates> by <Authors>. All Rights Reserved.

## inters

### **inters(p1, p2, p3, p4, flag)**

Come in lisp. Trova l'intersezione tra p1-p2 e p3-p4. Se <flag> e' assente o nil le 2 linee vengono considerate infinite e il punto puo' essere esterno.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## clone



QS Informatica - Manuale LUA » AutoCAD » Punti »

## clone

**<punto>:clone()**

Ritorna la copia di un punto con gli stessi valori di x, y, z.

**Esempio 52. Esempio <punto>:clone()**

```
local pt2 = pt1:clone()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **polar**

**<punto>:polar(angolo, dist)**

Ritorna un punto relativo a <punto> distante <dist> con angolo <angolo> in radianti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## moverel



QS Informatica - Manuale LUA » AutoCAD » Punti »

### **moverel**

**<punto>:moverel(dx, [dy, [dz]])**

Somma alle coordinate (x,y,z) i valori dati. Ritorna l'oggetto di partenza.

**Esempio 53. Esempio moverel()**

```
pt1:moverel(0, 3) – somma 3 alla componente Y del punto <pt1>local p2 =  
pt1:clone():moveRel(10, 0, 0)
```

# angle



QS Informatica - Manuale LUA » AutoCAD » Punti »

## angle

**<punto>:angle(pt2)**

Ritorna l'angolo tra <punto> e il punto <pt2>

Copyright © <Dates> by <Authors>. All Rights Reserved.

## osnap

### <punto>:osnap(modo)

Ritorna un punto usando l'object snap di AutoCAD. <modo> e' una stringa che contiene uno o piu' modi di osnap di AutoCAD (end,mid,...).

#### Esempio 55. Esempio osnap()

```
local endpt = pt1:osnap( "_end" )
```

## **entsel**

**entsel(prompt) -> entity**

**entsel(prompt) -> entity, point**

Chiede all'utente di selezionare un'entita' e la ritorna.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entlast**

### **entlast([lastSubEntity]) -> entity**

Ritorna l'ultima entita' inserita nel disegno. Se e' presente il flag <lastSubEntity> di valore maggiore di 0, ritorna l'ultima sub-entita' (nel caso di INSERT o POLYLINE e' un SEQEND).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## handent

**handent(handle) -> entity**

Dato un handle (stringa) restituisce l'entita' corrispondente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## entmake

### entmake(dxflist) -> entity

Crea un'entita', come in LISP. Ritorna l'entita' creata oppure nil.

#### Esempio 56. Esempio entmake()

```
- se OK <ent> sara' la linea appena creatalocal ent = entmake{ {0,  
"LINE"}, {10, Point(1,0,0)}, {11, Point(2,1,0)} }
```

## **entmakex**

### **entmakex(dxflist) -> entity**

Crea un'entità non residente nel DB, ad esempio un XRECORD. Ritorna l'entità.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tblobjname**

**tblobjname(tablename, symbolname) -> entity**

Come in LISP.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entnext**

**<entity>:entnext() -> <entity>**

Ritorna la prossima entita' nel database di AutoCAD. Ritorna nil se e' l'ultima.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## erase



QS Informatica - Manuale LUA >> AutoCAD >> Entità >>

### erase

**<entity>:erase()**

Cancella l'entità.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entupd**

### **<entity>:entupd()**

Effettua un aggiornamento a video dell'entita' specificata. Necessaria in caso di aggiornamento di un'entita' complessa tipo polilinea o blocco con attributi.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **redraw**



QS Informatica - Manuale LUA >> AutoCAD >> Entità >>

## **redraw**

**<entity>:redraw([mode])**

Ridisegna un'entita'; mode (0 se omesso) indica la modalita' di ridisegno  
(0=disegna, 1=nascondi,...)

Copyright © <Dates> by <Authors>. All Rights Reserved.

## grdraw

### **grdraw(p1, p2, colore, hl)**

Funzione che disegna un' entità nel navigator. In ingresso vuole 2 punti, il colore(numero compreso tra 0 e 255) e hl (se hl diverso da 0 la disegna evidenziata, altrimenti la disegna in modalità normale).

Copyright © <Dates> by <Authors>. All Rights Reserved.

# explode



QS Informatica - Manuale LUA >> AutoCAD >> Entità >>

## explode

**<entity>:explode( {[ [returnss = 1] [, attr2text = 1] ]])**

Eplode un' entita' (blocco). Se attr2text = 1, gli attributi eventualmente presenti nel blocco vengono convertiti in testi (a parte quelli invisibili). Se returnss = 1 viene restituito un selection set contenente tutte le entita' create dal comando ESPLODI.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# entget



QS Informatica - Manuale LUA » AutoCAD » Entità »

## entget

**<entity>:entget() -> entlist**

**<entity>:entget(xAppName) -> entlist**

Ritorna la lista con i codici dell'entita'. Nella seconda forma, ritorna anche i dati estesi dell'applicazione xAppName.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## textbox



QS Informatica - Manuale LUA >> AutoCAD >> Entità >>

## textbox

**<entity>:textbox()**

Solo per entita' testo. Ritorna 2 punti, sintassi come LISP textbox.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## regapp

### **app = regapp()**

Funzione che registra una application name nel disegno. L'application name è usata per raggruppare, immagazzinare, recuperare e modificare i dati estesi. In ingresso vuole una stringa(nome app). Ritorna rtnorm o rterror a seconda del risultato dell'operazione.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **command\_entpoint**



QS Informatica - Manuale LUA » AutoCAD » Entità »

## **command\_entpoint**

**ritorno = command\_entpoint(entità , punto)**

Funzione che vuole in ingresso un entità e un punto. Serve in funzione ad un altro command(taglia, filetta..) per andare a modificare l'entità passata. Se l'operazione è andata a buon fine ritorna 1, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entget**

### **entget(entity) -> enlist**

Legge la lista di associazione di un'entita' <ent>. Meglio usare <ent>:entget()

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entmod**

### **entmod(entlist)**

Vedi <entity>:entmod().

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entfirst**

**entfirst()** -> <entity>

Restituisce la prima entita' del disegno, nil se il db e' vuoto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **tblnext**

### **tblnext(nometable, [rewind = 0]) -> entlist**

Permette di scorrere una tabella dei simboli di autocad ("block", "layer", ...). La prima volta si chiama con rewind=1, le altre senza il parametro. Ritorna ogni volta la lista di associazione per l'elemento. Ritorna nil quando non ci sono ulteriori simboli.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## tblsearch

**tblsearch(nometable, simbolname, [setnext = 0])**

Ricerca un elemento in una tabella dei simboli. Come in LISP. Ritorna ogni volta la lista di associazione per l'elemento. Ritorna nil quando non ci sono ulteriori simboli.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## count



QS Informatica - Manuale LUA » AutoCAD » Entità \ Liste di associazione »

## count

**<entlist>:count()**

Ritorna il numero di elementi di una lista.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## shift



QS Informatica - Manuale LUA » AutoCAD » Entità \ Liste di associazione »

## shift

### <entlist>:shift() -> dxf, valore

Elimina il primo elemento di una lista DXF e lo ritorna come coppia di valori. Il primo e' il codice DXF, il secondo il valore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## assoc

**<entlist>:assoc(code)**

Cerca il codice <code> in una lista di associazione. Ritorna il valore dell'elemento trovato o nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **nth**

### **<entlist>:nth(index)**

Ritorna il tipo e il valore dell'elemento numero <index> della lista. Ritorna nil se index non e' valido.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setnth



QS Informatica - Manuale LUA » AutoCAD » Entità \ Liste di associazione »

### setnth

**<entlist>:setnth(index, tp, value)**

Imposta l'elemento n-esimo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# set



QS Informatica - Manuale LUA » AutoCAD » Entità \ Liste di associazione »

## set

### <entlist>:set(**code**, **value**)

Imposta un elemento della lista di associazione nella lista. <code> è il codice, <value> il valore. Se esiste viene sostituito, altrimenti viene aggiunto in coda.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## append



QS Informatica - Manuale LUA » AutoCAD » Entità \ Liste di associazione »

## append

**<entlist>:append(code, value [, code, value ...])**

Aggiunge uno o più codici DXF ad una lista di associazione; non controlla se i codici sono già presenti; può essere usata per aggiungere dati estesi ad un'entità.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **entmod**

### **<entlist>:entmod()**

Modifica un'entita' leggendo le proprieta' da <entlist>. Ritorna 1 se ok. nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## del

### <entlist>:del(code)

Elimina un elemento di gruppo <code> dalla lista.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**new**



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

**new**

**selset.new()**

Ritorna un selection set vuoto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **usersel**

### **selset.usersel()**

Chiede una lista di selezione in modo interattivo (l'utente deve selezionare le entita'). Ritorna un oggetto <selset>

Copyright © <Dates> by <Authors>. All Rights Reserved.

**all**



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## **all**

### **selset.all()**

Ritorna un selection set contenente l'intero disegno.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## filter

### **selset.filter(lista\_filtro)**

Seleziona un selection set in base alla lista filtro, funziona come ssget("X") in lisp (vedi manuale). Ritorna un oggetto <selset>

#### **Esempio 59. Esempio selset.filter()**

```
-- crea un selection set con i punti nel layer "0"local ss =
selset.filter { {0, "POINT"}, {8, "0"} }
```

## ssget

### **selset.ssget(opzioni)**

Versione generale di ssget da utilizzare in casi particolari. Ritorna un oggetto <selset> o nil.

Formato opzioni:

**Tabella 17. selset.ssget(): Formato Opzioni**

Nome Campo	Tipo	Descrizione
mode	stringa	modo 'W'=window, '_C'=crossing, '_F'=fence...
pt1	punto	(punto 1 per window,crossing)
pt2	punto	(punto 2 per window,crossing)
filter	lista	lista_filtro (stesso formato di selset.filter)
pointlist	lista	Punti per fence, wp e cp, lista di oggetti di tipo <punto>.

## count



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

### count

**<selset>:count()**

Ritorna il numero di elementi di un selection set

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **add**



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

### **add**

**<selset>:add(ent)**

**<selset>:add(ss)**

Aggiunge al selection set l'entita' <ent>, oppure il selection set <ss>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **del**



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## **del**

**<selset>:del(ent)**

**<selset>:del(ss)**

Toglie dal selection set l'entita' <ent>, oppure il selection set <ss>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**free**



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## **free**

**<selset>:free()**

Libera le risorse associate ad un selection set.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# has



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## has

### <selset>:has(ent)

Ritorna 1 se l'entita' <ent> e' presente nel selection set.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## item

**<selset>:item(indice)**

Ritorna l'elemento di indice <indice> di un selection set. Come ssname.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# apply



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## apply

**<selset>:apply(funz)**

Chiama la funzione <funz> per ogni elemento del selection set, passandole l'entita'.

**Esempio 60. Esempio selset.apply()**

```
ss:apply(function(ent) ent:setColor(2)
end)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# draggen



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## draggen

**<selset>:draggen(arg)**

Effettua il draggen della selection set

**Tabella 18. selset.draggen(): Formato Opzioni**

Nome Campo	Tipo	Descrizione
refpoint	punto	punto di riferimento per il trascinamento e la rotazione rispetto alla selection-set
rotate	booleano	Stabilisce se eseguire anche la rotazione della selezione dopo il trascinamento. (Default = falso)

Copyright © <Dates> by <Authors>. All Rights Reserved.

# intersect



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## intersect

**<selset>:intersect(ss2) -> ss3**

Ritorna un selection set contenente le entita' che sono anche in <ss2>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## extent



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## extent

### <selset>:extent()

Ritorna le dimensioni (x,y,z) del parallelepipedo piu' piccolo che puo' contenere le entita'.

### Esempio 62. Esempio selset.extent()

```
local dx, dy, dz = ss:extent()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## insert



QS Informatica - Manuale LUA » AutoCAD » Selection-Set »

## insert

```
<selset>:insert{ [refPoint = <punto>,] [scale = <n>,][insPoint = <punto>,]  
[rotate = <n> }
```

Inserisce le entita' del selection set (una copia) nel disegno, in modo simile all'inserimento di un blocco. *<refPoint>* e' il punto di definizione, *<insPoint>* il punto di inserimento. Tutti i parametri sono opzionali; se *<insPoint>* viene omesso il comando e' interattivo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## line



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

### line

**<stile>:line(punto1, punto2 [, punto3....])**

Traccia una linea che congiunge i punti dati.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## line2



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

### line2

**<stile>:line2(tab, punto1, punto2 [, punto3....])**

Traccia una linea che congiunge i punti dati.

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:line2({ltype = "Continuous", layer = "alfa", color = 2},  
punto1, punto2 [, punto3....])
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## arc

**<stile>:arc(centro, raggio, angoloIniziale, angoloFinale)**

Disegna un arco. L'angolo e' espresso in radianti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## arc2

**<stile>:arc2(tab, centro, raggio, angoloIniziale, angoloFinale)**

Disegna un arco. L'angolo e' espresso in radianti.

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:arc2({ltype = "Continuous", layer = "alfa", color = 2},  
centro, raggio, angoloIniziale, angoloFinale)
```

## arc3

**<stile>:arc3(puntoInizio, puntoRif, puntoFinale)**

Disegna un arco passante per 3 punti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## arc32

**<stile>:arc32(tab, puntoInizio, puntoRif, puntoFinale)**

Disegna un arco passante per 3 punti.

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:arc32({ltype = "Continuous", layer = "alfa", color = 2},  
puntoInizio, puntoRif, puntoFinale)
```

## circle



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

### circle

**<stile>:circle(centro, raggio)**

Disegna un cerchio in base al centro e al raggio passato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## circle2



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## circle2

**<stile>:circle2(tab, centro, raggio)**

Disegna un cerchio in base al centro e al raggio passato.

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:circle2({ltype = "Continuous", layer = "alfa", color = 2},  
centro, raggio)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## polyline



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## polyline

**<stile>:polyline({points = listaPunti [,elev = nnn] [, nolw = 1]})**

Disegna una polilinea; points e' una lista di punti; elev indica l'elevazione (0 se omessa), nolw fa creare una polilinea tradizionale (non lightweight).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## polyline2



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## polyline2

<stile>:polyline2(tab, { points = listaPunti [,elev = nnn] [, nolw = 1]})

Disegna una polilinea; points e' una lista di punti; elev indica l'elevazione (0 se omessa), nolw fa creare una polilinea tradizionale (non lightweight).

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:polyline2({ltype = "Continuous", layer = "alfa", color = 2},  
{ points = listaPunti [,elev = nnn] [, nolw = 1]})
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# clone



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## clone

**<stile>:clone()**

Ritorna una copia dello stile.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# apply



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## apply

**<stile>:apply(ent [,ent2...])**

Applica ad una o piu' entità le proprietà dello stile.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## text2



QS Informatica - Manuale LUA » AutoCAD » Funzioni per il disegno delle entità »

## text2

**<stile>:text2(tab, punto, testo, stile, altezza, rotazione)**

Funzione che inserisce un testo <testo> nel punto specificato <punto>, con lo stile <stile>, l'altezza <altezza> e rotazione <rotazione> specificati in ingresso.

**tab** è una tabella (che può essere omessa) per modificare lo stile (layer\ colore\ tipo di linea) dell'entità da disegnare.

```
Es:<stile>:text2({ltype = "Continuous", layer = "alfa", color = 2},  
punto, testo, stile, altezza, rotazione)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## insert



QS Informatica - Manuale LUA » AutoCAD » Blocchi »

## insert

<block>:insert{parametri} -> <entita'>

Inserisce un blocco nel disegno.

**Tabella 21. insert(): Formato tabella parametri:**

Nome Campo	Tipo	Descrizione
pos	punto	punto di inserimento (default: origine)
scale	numero	scala (default: 1)
angle	numero	rotazione (default: 0)

Copyright © <Dates> by <Authors>. All Rights Reserved.

## append



QS Informatica - Manuale LUA » AutoCAD » Dati estesi (DXF Code) »

## append

**<entlist>:append(code, value [, code, value ...])**

Aggiunge uno o più codici DXF ad una lista di associazione; non controlla se i codici sono già presenti; può essere usata per aggiungere dati estesi ad un'entità. Una volta finite le modifiche, salvare l'entità con le funzioni "entmod" e "entupd".

es.

```
edata = <ent>:ent:entget()  
edata:entmod()  
edata:append(-3, 1)  
edata:append(codiceDxf, valore)  
edata:entmod()  
edata:entupd()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## shift



QS Informatica - Manuale LUA » AutoCAD » Dati estesi (DXF Code) »

### shift

**<entlist>:shift() -> dxf, valore**

Elimina il primo elemento di una lista DXF e lo ritorna come coppia di valori. Il primo e' il codice DXF, il secondo il valore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## regapp

### **regapp(nomeApplicazione)**

Registra una applicazione xData. nomeApplicazione è una tabella.

es.

```
tabApp = {}  
tabApp.xDataApp = "apcLegdis"  
regapp(tabApp.xDataApp)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **namedobjdict**

### **namedobjdict() -> entity**

Ritorna l'oggetto (unico) <named object dictionary>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dictsearch

### **dictsearch(dict, sym [, new]) -> enlist**

Cerca nel dizionario `<dict>` l'oggetto identificato dal nome `<sym>`. Il parametro `<new>`, booleano, indica se la ricerca e' nuova (se 1) oppure si sta continuando una ricerca precedente. Ritorna la lista di associazione (come `entget`) dell'oggetto trovato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dictnext

**dictnext(dict[ , rewind]) -> entlist**

Avanza alla prossima entità nel dizionario <dict>. Se <rewind> è presente e diverso da 0, riparte. Se ok, restituisce la lista dei codici dxf, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dictrename



QS Informatica - Manuale LUA » AutoCAD » Dizionari »

## dictrename

### **dictrename(dict, oldname, newname)**

Rinomina una voce del dizionario. <dict> e' il dizionario che contiene la voce. <oldname> e' il vecchio nome, <newname> e' il nuovo nome. Ritorna 1 se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dictremove



QS Informatica - Manuale LUA » AutoCAD » Dizionari »

## dictremove

### **dictremove(dict, symtodelete)**

Cancella una voce di un dizionario. <dict> e' il dizionario che contiene la voce. <symtodelete> e' il nome della voce da cancellare. Ritorna 1 se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## dictadd

### **dictadd(dict, symname, objtoadd)**

Aggiunge un simbolo al dizionario. <dict> e' il dizionario, <symname> e' il nome della voce da aggiungere, <objtoadd> e' l'oggetto da aggiungere (normalmente di tipo xrecord). Ritorna 1 se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## group2selset

**group2selset(groupname) -> selset**

Restituisce un selection set contenente tutte le entita' del gruppo <groupname>.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupList

**lista = groupList()**

Ritorna sottoforma di lista i gruppi presenti nel dizionario.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupnew

**ritorno = groupnew(strNome, strDescr, ent1/selset)**

Funzione che crea un nuovo gruppo. In ingresso vuole nome, description, entità/selectionset. Ritorna 1 se l'operazione è andata a buon fine, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupaddent

**ritorno = groupaddent(strNomeGroup, ent1/selset)**

Funzione che in ingresso vuole il nome del gruppo (la stringa e non l'oggetto gruppo) e un entità/selectionset. Aggiunge l'entità/selectionset ad un gruppo. Ritorna 1 se l'operazione è andata a buon fine, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupdelent

**ritorno = groupdelent(strNomeGroup, ent1/selset)**

Funzione che in ingresso vuole il nome del gruppo (la stringa e non l'oggetto gruppo) e un entità/selectionset. Rimuove l'entità/selectionset da un gruppo. Ritorna 1 se l'operazione è andata a buon fine, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## grouprename

**ritorno = grouprename(strOldGroup, strNewGroup, strNewDescr)**

Funzione che rinomina un gruppo e cambia la sua descrizione. In ingresso vuole oldGruppo(str), newGruppo(str), newDescr(str). Ritorna 1 se l'operazione è andata a buon fine, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupsetcolor

QS Informatica - Manuale LUA » AutoCAD » Gruppi »



## groupsetcolor

**ritorno = groupsetcolor(strNomeGroup, colore)**

Funzione che in ingresso vuole il nome del gruppo (la stringa e non l'oggetto gruppo) e un numero rappresentante il colore (l'intervallo ammesso è tra 0 e 256). Imposta il colore delle entità del gruppo. Ritorna 1 se l'operazione è andata a buon fine, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupgetDescription

QS Informatica - Manuale LUA » AutoCAD » Gruppi »



## groupgetDescription

**descr = groupgetDescription(strNomeGroup)**

Funzione che ritorna la descrizione di un gruppo. In ingresso vuole il nome del gruppo (la stringa e non l'oggetto gruppo)

Copyright © <Dates> by <Authors>. All Rights Reserved.

# initget



QS Informatica - Manuale LUA » AutoCAD » Input Utente »

## initget

### initget(flags, keywords)

Inizializza lo stato della prossima funzione di input. Come in Lisp.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# getstring



QS Informatica - Manuale LUA » AutoCAD » Input Utente »

## getstring

### getstring(prompt)

Chiede una stringa come in autolisp/ads. Ritorna la stringa o nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **getreal**

### **getreal(prompt) -> num**

Chiede un numero reale dal prompt di AutoCAD. Ritorna il numero o nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# getkeyword



QS Informatica - Manuale LUA >> AutoCAD >> Input Utente >>

## getkeyword

### getkeyword(prompt)

Chiede una keyword come in autolisp/ads. Ritorna la stringa o nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **getdist**

### **getdist([prompt], [ptRiferimento])**

Chiede l'immissione di una distanza, come numero oppure come coppia di punti.  
Il parametro opzionale <ptRiferimento> permette di indicare un punto di riferimento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getangle

### **getangle(prompt, ptRiferimento)**

Chiede l'immissione di un angolo. Passando nil come ptRiferimento oppure prompt si utilizza il valore di default.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **mklayer**

**ritorno = mklayer(str)**

Funzione che crea un nuovo layer. Vuole in ingresso una stringa che sarà il nome del layer. Impedisce la creazione di layer con lo stesso nome. Se l'operazione è andata a buon fine ritorna 1, altrimenti nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_ptop



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## top\_ptop

### top\_ptop() -> parte

Ritorna la parte TOP del disegno attivo

#### Esempio 66. Esempio top\_ptop()

```
top_ptop():setAttr("codice", "2")local top = top_ptop()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.



## **top\_pcur**

### **top\_pcur() -> parte**

Ritorna la parte CORRENTE del disegno attivo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## ancestors



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### ancestors

#### <parte>:ancestors()

Restituisce una lista con tutte le parti incontrate fino alla parte TOP. La parte top e' l'ultima della lista.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## activate



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### activate

**<parte>:activate()**

Attiva una parte.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## new



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### new

**<parte>:new(nome)**

Crea una sottoparte, ritorna la parte creata.

**Esempio 69. Esempio new()**

```
local prt = prt:new('Nuova parte')prt:activate()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **xform**

**<parte>:xform(matrix)**

- **matrix**
- (tipo matrice) matrice di trasformazione (ss:xform)

Applica la matrice di trasformazione <matrix> alle entita' della parte.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## redraw



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## redraw

<parte>:redraw(modoRedraw)

- modoRedraw
- (tipo intero) modalita' di ridisegno

**Tabella 26. redraw: Valori parametro <modoRedraw> (1 se omesso)**

Valore	Descrizione
1	Ridisegna
2	Nasconde (undraw)
3	Evidenzia (highlight)
4	Dis-Evidenzia (unhighlight)

Ridisegna tutte le entita' di una parte (e delle sue sottoparti).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## aggr

### <parte>:aggr(ent)

Aggrega alla parte gli oggetti di <ent>. E' possibile passare sia entita' singole che gruppi di selezione.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **explode**



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## **explode**

**<parte>:explode()**

Eplode la parte. Non chiede conferme.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## erase



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### erase

#### <parte>:erase()

Cancella la parte e le sue sottoparti, non chiede conferme.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## foreach



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## foreach

### <parte>:foreach(func)

Scorre tutte le parti ad ogni livello, e chiama la funzione <func> per ogni parte, passandola come parametro.

Se <func> ritorna vero (non nil) la scansione si interrompe e foreach ritorna il valore.

#### Esempio 70. Esempio foreach()

```
-- Esempio cerca una parte con attributo altezza=2
local parteTrovata =
top_ptop():foreach(function(prt)
    if (prt:getAttr("altezza")=="2") then
        return prt
    end
end)
```

## foreach1



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### foreach1

#### <parte>:foreach1(func)

Scorre tutte le parti al primo livello (riferito a <parte>), e chiama la funzione <func> per ogni parte, passandola come parametro. Se <func> ritorna vero (non nil) la scansione si interrompe e foreach1 ritorna il valore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## find



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### find

#### <parte>:find(nome)

Cerca una sottoparte di <parte> con nome <nome>. Se sono presenti piu' parti con lo stesso nome, viene ritornata la prima.

#### Esempio 71. Esempio find()

```
-- Esempio: verifica se esiste la parte cartiglio.local
cart = top_ptop():find(".cart")
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# load



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## load

### <parte>:load(params)

Carica una parte da un file. <params> e' una tabella con il seguente formato:

**Tabella 27. <parte>:load(): Formato della Tabella.**

Nome	Tipo	Descrizione
file	stringa	nome del file da caricare
opendir	stringa	directory di partenza per l'apertura file
angle	numero	angolo di rotazione, se omesso lo chiede
point	punto	punto di inserimento, se omesso lo chiede
scale	numero	scala, se omesso 1.0

Copyright © <Dates> by <Authors>. All Rights Reserved.

## save



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

### save

#### <parte>:save(params)

Salva una parte <params> e' una tabella con il seguente formato:

**Tabella 28. <parte>:save(): Formato della tabella params.**

Nome	Tipo	Descrizione
file	stringa	nome del file da salvare, se manca lo chiede all'utente
subprt	0 oppure 1	Se 1 (default) salva le sottoparti, se 0 solo la parte.
opendir	Stringa	Directory di partenza per il dialogbox salva...

Copyright © <Dates> by <Authors>. All Rights Reserved.

## copy



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## copy

### <parte>:copy(params)

Versione lua del comando "copia parte". <params> e' una tabella con il seguente formato:

**Tabella 29. <parte>:copy(): Formato della tabella params.**

Nome	Tipo	Descrizione
to	parte	la nuova parte padre
single	bool	Se vero non copia le sottoparti (default falso).
structOnly	bool	Se vero non copia le entita' ma solo la struttura (default falso).
noUnlock	bool	Se vero non effettua l'unlock dei layers (e' pero' necessario che i layers siano stati precedentemente sbloccati). (default falso).

## hide

### <parte>:hide(flag)

Spegne la parte <parte> e a seconda del flag passatogli può nascondere anche le parti figlie.

**Tabella 30.**

Valore Flag	Conseguenza
Null	Spegne solo la parte <parte>
0	Spegne solo la parte <parte>
1	Spegne la parte <parte> e le sue parti figlie



#### Nota

Per spegnere le parti figlie non è necessario passargli un 1, il numero passato può essere anche un qualsiasi numero positivo o negativo.

# show



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

## show

### <parte>:show(flag)

Accende la parte <parte> e a seconda del flag passatogli può accendere anche le parti figlie.

**Tabella 31.**

Valore Flag	Conseguenza
Null	Accende solo la parte <parte>
0	Accende solo la parte <parte>
1	Accende la parte <parte> e le sue parti figlie



### Nota

Per accendere le parti figlie non è necessario passargli un 1, il numero passato può essere anche un qualsiasi numero positivo o negativo.

\_p



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto Parte »

**\_p**

**<parte>:\_p()**

Ritorna il puntatore della parte *parte*.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **\_fromp**

**<parte>:\_fromp([puntatore])**

Funzione che in ingresso vuole un puntatore ad una parte e ti ritorna quest'ultima.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## add



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

### add

**<prtArray>:add(parte [, figli])**

Aggiunge all'array una parte singola (con sottoparti se <figli> = 1).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## count



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

### count

**<prtArray>:count()**

Ritorna il numero di parti contenute nel PrtArray.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# apply



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

## apply

### <prtArray>:apply(funz)

Chiama la funzione funz per ogni parte del PrtArray, passandole la parte come parametro.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# show



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

## show

### <prtArray>:show()

Visualizza solo le parti del prtArray.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **explode**



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

## **explode**

**<prtArray>:explode()**

Esplode tutte le parti del prtArray.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **lman**

### **<prtArray>:lman(comando)**

Esegue il comando sui layer specificato per ogni parte del prtArray  
(0=congela,1=blocca,2=sbloccca).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## diff



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Oggetto PrtArray »

### diff

**<prtArray>:diff(prtArray2) -> prtArray**

Ritorna un nuovo prtArray che contiene le parti del prtArray sorgente che non sono nel prtArray2.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_prtfroment**

### **top\_prtfroment(entity)**

Cerca la parte che contiene l'entita' <entity>. Ritorna la parte o nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_prtselect



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Globali »

### top\_prtselect

**top\_prtselect{prompt=<stringa>, title=<stringa>} -> prt**

Chiede all'utente di selezionare una parte da un dialogo che visualizza la struttura ad albero. Ritorna la parte selezionata o nil se l'utente annulla.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_spotlight

QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Globali »



## top\_spotlight

### top\_spotlight(modo)

Imposta la visualizzazione delle "linee luminose" secondo la stringa <modo>  
Modo può contenere "ON", "FIGLI", "TEMP" (non salva la configurazione).

#### Esempio 75. Esempio: top\_spotlight()

```
local old = top_spotlight("OFF+TEMP").... -- operazioni senza spotlight  
top_spotlight(ol)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.



## **top\_updspotlight**

### **top\_updspotlight()**

Scorre tutte le entita' del model space ed aggiorna le spotlight.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_prtfullpath



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Globali »

## top\_prtfullpath

### top\_prtfullpath(prtName)

Ritorna il percorso della parte specificata in ingresso della funzione.

#### Esempio 76. Esempio: top\_prtfullpath()

```
LUA>print(top_prtfullpath('parte1'))c:\programmi\qs\top_prt\parte1
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_envfullpath



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Globali »

## top\_envfullpath

### top\_envfullpath(envName)

Ritorna il percorso dell'ambiente specificato in ingresso della funzione.

#### Esempio 77. Esempio: top\_envfullpath()

```
LUA>print(top_envfullpath('ambiente1'))c:\programmi\qs\env\top_base\ambi
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_prtselectmulti



QS Informatica - Manuale LUA » Modulo Parti (TopPrt.arx) » Globali »

## top\_prtselectmulti

<prtArray>:top\_prtselectmulti(tabopzioni)

- **tabopzioni**
- (tipo tabella)

**Tabella 38. Formato della tabella tabopzioni.**

Parametro	Tipo	Descrizione
title	stringa	(opz.) Titolo della finestra
initIcon	funzione	funzione che riceve in ingresso la parte e ritorna l'indice dell'icona da impostare. Sono disponibili 10 icone standard (pallini colorati) che possono essere utilizzate per default. L'indice 0 ritorna un'icona bianca (nessuna icona).
selfunc	funzione	(opz.) funzione che riceve in ingresso la parte e ritorna 1 se la parte deve essere preselezionata nell'albero parti.
sorted	booleano	(opz.) se true, ordina l'albero

		alfabeticamente;
initToggle	funzione	(opz.) riceve la parte e restituisce true se la parte deve essere inizialmente selezionata.
initText	funzione	(opz.) riceve la parte e restituisce il testo da visualizzare al posto del nome.
imageList	punt.imagelist	puntatore alla image list (HIMAGELIST) da usare per le icone
prtBase	parte	parte di base (radice dell'albero).
expandLevels	intero	(default.1) livelli da espandere

Mostra l'albero parti di TOP con una casella di selezione a lato della parte.  
Ritorna l'elenco delle parti selezionate.

## **top\_discart**



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »

## **top\_discart**

### **top\_discart([fullpath], [scala])**

Permette di inserire a disegno il cartiglio. E' possibile passare come primo parametro il percorso completo del cartiglio da inserire altrimenti viene richiesta all'utente la selezione del file nell'ambiente corrente. Il secondo parametro consente di definire la scala di inserimento sottoforma di stringa (Es "1:4"). Se viene omesso, la scala viene richiesta all'utente tramite una finestra di dialogo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_updcart**

QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »



## **top\_updcart**

### **top\_updcart()**

Aggiorna gli attributi visualizzati nel cartiglio.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## top\_distab



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »

## top\_distab

### **top\_distab([ptPosa, ptRif])**

Disegna la tabella di pallinatura. Se non vengono passati i parametri relativi ai punti di riferimento equivale al comando top. In caso contrario non comtepla l'interazione con l'utente ma disegna direttamente la tabella.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_aggtab**



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »

## **top\_aggtab**

### **top\_aggtab([showMsg])**

Aggiorna la tabella di pallinatura. Se non si specifica alcun parametro i messaggi non vengono visualizzati.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_updtab**



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »

## **top\_updtab**

### **top\_updtab([prt])**

Aggiorna gli attributi visualizzati nella tabella. Se è presente il parametro <prt> aggiorna solo la riga corrispondente a quella parte, altrimenti tutta la tabella. Se <prt> coincide con la parte top, viene aggiornato anche il cartiglio.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_pallprt**

### **top\_pallprt(prt, [listaPunti])**

Permette di pallinare la parte passata. Richiede all'utente solo la selezione dei punti di posa. Usa le impostazioni dello stile corrente.

*ListaPunti* è un argomento opzionale e rappresenta la lista di punti atti a costruire la leader.



## **top\_pallprtmulti**

### **top\_pallprtmulti(prtArray)**

Permette di pallinare in sequenza tutte le parti presenti nel prtArray. Usa le impostazioni dello stile corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_aggbblkpall**

QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »



### **top\_aggbblkpall**

#### **top\_aggbblkpall()**

Aggiorna in modo forzato l'attributo del blocco associato a tutti i pallini dello stile corrente presenti sul disegno.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_delpallpart**



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Pallinatura & Cartiglio »

## **top\_delpallpart**

### **top\_delpallpart(prt)**

Elimina la pallinatura associata alla parte passata. Opera sulla pallinatura associata allo stile corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_eticprt**

### **top\_eticprt(prt, Attr)**

Permette di etichettare la parte a seconda dell'attributo di riferimento passato. Richiede all'utente solo la selezione dei punti di posa. Usa le impostazioni dello stile corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_eticprtmulti**

### **top\_eticprtmulti(prtArray, Attr)**

Permette di etichettare in sequenza tutte le parti presenti nel prtArray a seconda dell'attributo di riferimento passato. Usa le impostazioni dello stile corrente.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## <style>:rename



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Stili di Pallinatura »

## <style>:rename

### <style>:rename(styleName)

Rinomina lo stile con il nuovo nome passato. Se fallisce ritorna un nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## <style>:activate



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Stili di Pallinatura »

### <style>:activate

#### <style>:activate()

Rende corrente l'oggetto stile.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## <style>:close



QS Informatica - Manuale LUA » Pallinatura\Cartiglio (PallCart.arx) » Stili di Pallinatura »

## <style>:close

### <style>:close()

Chiude l'oggetto e libera tutte le risorse.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# open



QS Informatica - Manuale LUA » LUA-ADO » Ado\_Database »

## open

**<adodatabase>:open(connection\_string)**

Si collega al database definito da connection\_string. Ritorna nil in caso di errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## execute



QS Informatica - Manuale LUA » LUA-ADO » Ado\_Database »

## execute

### <adodatabase>:execute(sqlstring)

Esegue una stringa sql che non ritorna risultati. Ritorna vero (1) se ok, nil in caso di errore.

#### Esempio 100. Esempio <adodatabase>:execute()

```
db:execute("delete * from tab1 where codice=2")local ok =
db:execute("update tab1 set dim=3 where codice=2")
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## close



QS Informatica - Manuale LUA » LUA-ADO » Ado\_Database »

## close

### <adodatabase>:close()

Chiude la connessione. Dopo avere chiamato questa funzione l'oggetto db non e' piu' utilizzabile.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# open



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

## open

### <adorecordset>:open(str)

Apre il recorset, str puo' essere del codice sql, oppure il nome di una table. In caso di errore ritorna nil.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## close



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

## close

### <adorecordset>:close()

Chiude il recorset. Per riutilizzarlo e' necessario eseguire l'open di nuovo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## edit



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

### edit

#### <adorecordset>:edit()

Restituisce il punto di inserimento della parte, come entità punto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# update



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

## update

### <adorecordset>:update()

Conferma un'operazione di modifica come addNew. Ritorna vero (1) se ok, nil se errore. Nel caso ritorni nil e' necessario annullare l'operazione di modifica tramite cancelUpdate.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **delete**



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

## **delete**

### **<adorecordset>:delete()**

Elimina il record attualmente selezionato nel recorset. Ritorna vero (1) se ok, nil se errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## find



QS Informatica - Manuale LUA » LUA-ADO » Ado\_RecordSet »

## find

**<adorecordset>:openXML(filename)**

Come open, ma legge il contenuto del recordset dal file XML filename.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# dbpath



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedType »

## dbpath

**<dedtype>:dbpath()**

Ritorna il dbpath dell'tipo..

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **desc**



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedType »

### **desc**

**<dedtype>:desc()**

Ritorna la descrizione del tipo .

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **name**



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedType »

### **name**

**<dedtype>:name()**

Come DED.getFile ma ritorna il filename completo del path.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **prompt2name**

**<dedtype>:prompt2name(prompt)**

Converte da prompt a nome. Ritorna una stringa. Passando una tabella con coppie (prompt, valore) viene restituita una nuova tabella (nome, valore).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **name2prompt**

**<dedtype>:name2prompt(name)**

Converte da nome a prompt. Ritorna una stringa. Passando una tabella con coppie (nome, valore) viene restituita una nuova tabella (prompt, valore).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## name



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedAttr »

### name

#### <**dedattr**>:name()

Ritorna il nome (nome del campo nel database) dell'attributo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **prompt**

### **<dedattr>:prompt()**

Ritorna il prompt (nome visualizzato) dell'attributo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## type



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedAttr »

## type

**<dedattr>:type()**

Ritorna il tipo (stringa, num) dell'attributo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# length



QS Informatica - Manuale LUA » Data & Drawings » Oggetto DedAttr »

## length

### <dedattr>:length()

Ritorna la lunghezza (numero di caratteri) dell'attributo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lua

### <**dedattr**>:lua()

Ritorna una tabella con gli attributi aggiuntivi per l'edita attributi.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## refresh



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

## refresh

### <DedAttachList>:refresh()

Aggiorna il contenuto della lista rileggendo da file system, nel caso di gestione standard o ricaricandola dal file .qsa.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## count



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

### count

#### <DedAttachList>:count()

Restituisce il numero degli allegati presenti.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## item

### <DedAttachList>:item(idx)

Restituisce l'oggetto DedAttachListItem all'indice specificato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **save**



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

## **save**

### **<DedAttachList>:save()**

Serializza su file l'oggetto. Restituisce 1 se riesce altrimenti nil ed un messaggio di errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## find



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

## find

### <DedAttachList>:find(item)

Cerca un oggetto DedAttachListItem nella lista e ne restituisce l'indice oppure nil se non viene trovato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## add



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

## add

### <DedAttachList>:add(filename [, ded\_dis])

Aggiunge un allegato solo se non esiste già. Nel caso di gestione standard è sufficiente passare il filename del file da allegare (DedAttachUseQsaFile = nil) altrimenti si deve indicare anche il ded\_dis di provenienza dell'allegato (e deve quindi trattarsi di file già codificato). Restituisce 1 oppure nil ed un messaggio di errore se non riesce nell'operazione.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## remove



QS Informatica - Manuale LUA » Gestione allegati » DedAttachList »

## remove

### <DedAttachList>:remove(item)

Elimina un allegato. Restituisce 1 se riesce altrimenti nil ed un messaggio di errore.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## start

### **DEDDIST.start([codice, tabella, flags])**

Lancia il dialog-box distinta. codice (opzionale) indica il codice iniziale da visualizzare in alto. tabella (opzionale) e' una lista di liste con i valori iniziali da visualizzare nella griglia. flags (opzionale) e' una stringa che contiene delle parole chiave che indicano opzioni aggiuntive:

**Tabella 50. DEDDIST.start() Tabella Valori "flags"**

“silent”	non fa domande (tipo “vuoi sovrascrivere ?”);
“skip”	non sovrascrive mai il record; solo append;
“autoclose”	chiude automaticamente il dialogo appena aperto;
“autoload”	all'avvio lancia automaticamente “carica”
“autosave”	dopo l'inizializzazione lancia “Save”, non chiude il dialogo;

Se non vengono forniti i parametri opzionali il dialogo parte vuoto.

## addcol



QS Informatica - Manuale LUA » Editor Distinte (DED\_DIST) » Funzioni esterne »

## addcol

### **DEDDIST.addcol(tabParam)**

Funzione da chiamare solo durante l'inizializzazione. Aggiunge una colonna alla griglia. Formato *tabParam*

**Tabella 51. DEDDIST.addcol() Tabella Opzioni**

Nome Campo	Tipo	Descrizione
title	stringa	Titolo della colonna.
width	intero	Larghezza colonna (pixel).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **addrow**

### **DEDDIST.addrow()**

Aggiunge una riga vuota alla griglia. Gli elementi vanno poi impostati con .set().

Copyright © <Dates> by <Authors>. All Rights Reserved.

## set



QS Informatica - Manuale LUA » Editor Distinte (DED\_DIST) » Funzioni esterne »

### set

#### **DEDDIST.set(riga, colonna, valore)**

Imposta il valore contenuto in una cella. riga e' la il numero di riga (parte da 1) colonna e' il numero di colonna (parte da 1) valore e' il valore (testo).

Copyright © <Dates> by <Authors>. All Rights Reserved.

**clear**



QS Informatica - Manuale LUA » Editor Distinte (DED\_DIST) » Funzioni esterne »

## **clear**

### **DEDDIST.clear()**

Cancella tutto il contenuto della griglia.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# init



QS Informatica - Manuale LUA » Editor Distinte (DED\_DIST) » CallBacks »

## init

### **DEDDIST.init()**

Chiamato appena dopo la visualizzazione del dialogo (initdialog). Puo' aggiungere colonne o modificare elementi della griglia.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## onload



QS Informatica - Manuale LUA » Editor Distinte (DED\_DIST) » CallBacks »

## onload

### **DEDDIST.onload(codice)**

Chiamato quando l'utente preme il pulsante 'carica'. codice e' il codice da caricare.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **onsave**

### **DEDDIST.onsave()**

Chiamato quando l'utente preme il pulsante 'salva'. Dovrebbe salvare la tabella in ded\_bom.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **onexport**

### **DEDDIST.onexport()**

Chiamato quando l'utente preme il pulsante 'esporta'.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## count



QS Informatica - Manuale LUA » Utility » LOG »

### count

#### <LOG>:count()

Restituisce il numero di errori registrati.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## close



QS Informatica - Manuale LUA » Utility » LOG »

## close

### <LOG>:close([tabparam])

Chiude la registrazione degli errori ed apre il file di Log in base all'impostazione attuale di *showmode* oppure in base a quella passata.

**Tabella 53. Tabella Opzioni per tabparam**

Nome	Tipo	Descrizione
showmode	numero	<ul style="list-style-type: none"><li>• nil : apre solo se con errori oppure visualizza il messaggio di procedura conclusa con successo (compatibilità col passato);</li><li>• 1 : apre solo se con errori e comunque non visualizza mai il messaggio;</li><li>• 2 : apre sempre senza visualizzare alcun messaggio;</li><li>• 3 : sia che siano avvenuti o no degli errori, non visualizza messaggi e non apre il file di</li></ul>

		log;
msgok	stringa	Messaggio personalizzato da visualizzare in caso di procedura senza errori (viene ignorato ne casi in cui il messaggio non viene visualizzato).
delfile	booleano	Se ha valore 0 non cancella il file, altrimenti lo cancella: di default è 1. Questo flag viene ignorato se il file deve essere visualizzato all'utente o se sono presenti degli errori registrati nel log.

### Esempio 109. Esempio newDlgCancOp()

```
-- Argomenti precedente versione (ancora supportati) :
Close(msgok, delfile)
-- Argomenti versione corrente :
Close{msgok = "Terminato", delfile = 1, showmode = nil}
```

# show



QS Informatica - Manuale LUA » Utility » LOG »

## show

### <LOG>:show([verb])

Apre il file in tutti i casi, sia che ci siano state delle registrazioni di errore o meno, il file di log. 'verb' (opzionale) che di default è 'open' può essere anche 'print' o una opzione valida presente nella shell.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **delete**



QS Informatica - Manuale LUA » Utility » LOG »

## **delete**

### **<LOG>:delete()**

Elimina fisicamente il file di log.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## show



QS Informatica - Manuale LUA » Finestra di Dialogo .NET (PalDialog.dll) » Gestione dialogo »

## show

**<dialogo>:show()**

Mostra il dialogo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## hide



QS Informatica - Manuale LUA » Finestra di Dialogo .NET (PalDialog.dll) » Gestione dialogo »

### hide

**<dialogo>:hide()**

Nasconde il dialogo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## next

### <GridDlg>:next()

In fase di creazione dei controlli, si posiziona nella cella a destra di quella corrente.

**Cr****<GridDlg>:cr()**

In fase di creazione dei controlli, si posiziona nella prima colonna della prossima riga.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# enable



QS Informatica - Manuale LUA » LuaGridDlg » Funzioni »

## enable

**<GridDlg>:enable(zona [, zonan...])**

Abilita una zona.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## disable



QS Informatica - Manuale LUA » LuaGridDlg » Funzioni »

## disable

**<GridDlg>:disable(zona [, zonan...])**

Disabilita una zona.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## zone



QS Informatica - Manuale LUA » LuaGridDlg » Funzioni »

### zone

**<GridDlg>:zone(nome) -> <GridZone>**

Ritorna l'oggetto zona definito con nome oppure nil se non trovato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **seldata**



QS Informatica - Manuale LUA » LuaGridDlg » Funzioni per la definizione dei Controlli »

## **seldata**

### **<GridZone>:seldata()**

In una cella di tipo combo, ritorna il valore xdata dell'elemento attualmente selezionato (cella unica).

Copyright © <Dates> by <Authors>. All Rights Reserved.

# enable



QS Informatica - Manuale LUA » LuaGridDlg » Funzioni per la definizione dei Controlli »

## enable

**<GridZone>:enable(flagAbilita)**

Abilita o disabilita una zona.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **inv\_curdoc**



QS Informatica - Manuale LUA » Inventor (LualInvNet.dll) » Oggetto Document »

## **inv\_curdoc**

### **inv\_curdoc() -> document**

Ritorna il documento attivo.

#### **Esempio 118. Esempio inv\_curdoc()**

```
top_ptop():setAttr("codice", "2") local top = top_ptop()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_ptop**



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Document »

## **top\_ptop**

**top\_ptop()** -> <document>

Restituisce il documento attivo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## inv\_docopen



QS Informatica - Manuale LUA » Inventor (LualInvNet.dll) » Oggetto Document »

## inv\_docopen

### **inv\_docopen(filename[, visible])-> document**

Apre un documento in Inventor e lo restituisce. Se visible è presente ed è diverso da 1, il documento viene aperto non visibile. Di default il documento viene aperto visibile.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **foreach1**

```
<document>:foreach1(func<occ>)
<occurrence>:foreach1(func<occ>)
```

Scorre tutte le occorrenze figlie al primo livello referenziate dall'assieme doc/occ, e chiama la funzione <func> per ognuna, passandola come parametro. Fate molta attenzione a richiamare correttamente l'oggetto dell'iterazione. Negli assiemi multilivello cambia il contesto dell'occorrenza a seconda che sia richiamato da un doc oppure da un occ del doc ad un livello superiore. Se si sta facendo riferimento ad un doc inserito come sotto-assieme occorre iterarlo come occorrenza.

# load



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Document »

## load

### <document>:load(fullfilename[, sincrono])

Carica nell'assieme corrente il documento indicato da fullfilename (deve essere uno .IAM o un .IPT). Il caricamento avviene in modo asincrono, se invece si specifica il secondo parametro il caricamento avviene in modo sincrono. In questo caso viene restituita l'ultima occorrenza inserita dall'utente ed il numero di occorrenze inserite.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **save**



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Document »

## **save**

### **<document>:save()**

Salva il documento. Il documento deve avere già un nome. Restituisce nil se non è riuscito a salvare.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# update



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Document >>

## update

### <document>:update()

Aggiorna la grafica del documento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## preview



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Document »

## preview

### <document>:preview()

Visualizza una finestra non modale con l'anteprima del documento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## close



QS Informatica - Manuale LUA » Inventor (LualInvNet.dll) » Oggetto Document »

## close

### <document>:close(skipSave)

Chiude il documento; se skipSave è vero non chiede di salvare il documento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## new

### <occurrence>:new(filename [, templFileName])

Se filename rappresenta un nome di file (es: c:\temp\pippo.ipt) verrà creato un nuovo file su disco e aggiunto come parte figlia dell'occorenza (se l'occorenza riferenzia un documento di tipo assieme). Se l'estensione del nome di file è .iam verrà creato un sotto assieme. Se "name" non rappresenta un nome di file verrà creata una parte virtuale figlia dell'occorenza (se l'occorenza riferenzia un documento di tipo assieme). Il parametro templFileName permette di indicare il template di Inventor da utilizzare per creare una nuova parte.

## activate



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

### activate

**<occurrence>:activate()**

Rende attiva (selezionata) l'occorrenza.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## foreach



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

## foreach

### <occurrence>:foreach(func)

Scorre tutti le occorrenze partendo dalle occorrenze figlie di occ (occ inclusa) e chiama la funzione <func> per ogni occorrenza passandola come parametro.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## foreach1



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

### foreach1

#### <occurrence>:foreach1(func)

Scorre tutte le occorrenze figlie al primo livello referenziate dall'occurrence passata, e chiama la funzione <func> per ognuna, passandola come parametro.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# load



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

## load

### <occurrence>:load(fullfilename)

Carica nell'assieme corrente il documento indicato da fullfilename (deve essere uno .IAM o un .IPT).

Copyright © <Dates> by <Authors>. All Rights Reserved.

# update



QS Informatica - Manuale LUA » Inventor (LualInvNet.dll) » Oggetto Occurance »

## update

### <occurrence>:update()

Aggiorna la grafica del documento associato all'occorrenza.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# replace



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

## replace

### <occurrence>:replace(filename, replaceAll)

Sostituisce l'occorrenza <occ> con un documento (.iam/.ipt) specificato da filename, replaceAll indica se sostituire tutte le occorrenze della parte all'interno dell'assieme dove l'occorrenza si trova (non in eventuali sottoassiemi).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **delete**



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

### **delete**

#### **<occurrence>:delete()**

Rimuove l'occorrenza dall'assieme.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_pcur**

**top\_pcur()** -> <occurrence>

Ritorna l'occorrenza attiva (selezionata). Se fallisce ritorna nil ed una stringa contenente la motivazione del fallimento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **top\_pedit**



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto Occurance »

## **top\_pedit**

### **top\_pedit() -> <occurrence>**

Ritorna l'occorrenza in editazione. Se fallisce ritorna nil ed una stringa contenente la motivazione del fallimento.

Copyright © <Dates> by <Authors>. All Rights Reserved.

**name**



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Sheet >>

## **name**

**<Sheet>:name()**

Restituisce/imposta il nome del foglio.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## size



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Sheet >>

## size

<Sheet>.size()

Restituisce le dimensioni del foglio.

**Tabella 59. Size() Tabella Dimensioni Foglio**

tipo	valori
EnumDrawingSheetSizeEnum	kDefaultDrawingSheetSize;
	kCustomDrawingSheetSize;
	kADrawingSheetSize;
	kBDrawingSheetSize;
	kCDrawingSheetSize;
	kDDrawingSheetSize;
	kEDrawingSheetSize;
	kFDrawingSheetSize;
	kA0DrawingSheetSize;
	kA1DrawingSheetSize;
	kA2DrawingSheetSize;
	kA3DrawingSheetSize;
	kA4DrawingSheetSize;

Copyright © <Dates> by <Authors>. All Rights Reserved.

## width



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Sheet >>

## width

### <Sheet>.width()

Restituisce la larghezza dello sheet.

Copyright © <Dates> by <Authors>. All Rights Reserved.

# **height**



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Sheet >>

## **height**

**<Sheet>:height()**

Restituisce l'altezza dello sheet.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## activate



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto Sheet >>

## activate

**<Sheet>:activate()**

Attiva lo sheet.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## orientation



QS Informatica - Manuale LUA >> Inventor (LualInvNet.dll) >> Oggetto Sheet >>

## orientation

### <Sheet>:orientation()

Restituisce l'orientamento del foglio.

**Tabella 60. Orientation() Tabella Orientamento Foglio**

tipo	valori
EnumPageOrientationTypeEnum	kDefaultPageOrientation;
	kLandscapePageOrientation
	kPortraitPageOrientation;

Copyright © <Dates> by <Authors>. All Rights Reserved.

## name



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto SketchedSymbolDefinition >>

### name

**<SketchedSymbolDefinition>:name()**

Restituisce il nome dell'oggetto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## delete



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto SketchedSymbolDefinition >>

## delete

### <SketchedSymbolDefinition>:delete()

Elimina il riferimento dal documento solo se non è referenziato in qualche Sheet.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## name



QS Informatica - Manuale LUA » Inventor (LuaInvNet.dll) » Oggetto SketchedSymbol »

### name

**<SketchedSymbol>:name()**

Restituisce il nome dell'oggetto.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **delete**



QS Informatica - Manuale LUA >> Inventor (LuaInvNet.dll) >> Oggetto SketchedSymbol >>

### **delete**

#### **<SketchedSymbol>:delete()**

Elimina dallo Sheet il simbolo di schizzo.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## importassembly



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## importassembly

### importassembly(assemblypath)

Permette di caricare un assembly .NET. A differenza delle altre funzioni questa non ritorna l'assembly ma lo mantiene nella cache interna. Va usato in combinazione con importtype. Accetta in ingresso una stringa o una tabella di stringhe.

#### Esempio 121. Esempio importassembly

```
-- single assembly loadingimportassembly "mscorlib"  
importassembly "System"  
-- multiple assemblies loading  
importassembly {"mscorlib", "System", "System.Windows.Forms"}  
local String = importtype("System.String")
```

Questa chiamata consente di utilizzare gli **Extension Methods**.

Questa funzione richiede che LuaAsmLdr.dll sia caricato.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## loadassembly

### loadassembly(assemblypath)

Permette di caricare un assembly .NET passando il percorso alla dll

#### Esempio 122. Esempio loadassembly

```
local asm, err = loadassembly("vbluanet.dll")
```

questa funzione di caricamento si basa sul percorso del file.

il valore di ritorno **err** contiene la stringa di errore in caso di mancato caricamento.

Tutte le funzioni loadassembly sono in grado di riconoscere se all'interno dell'assembly esistono dei wrapper e, se trovati viene chiamato il metodo **Initialize** (se implementato)

## loadassembly2



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## loadassembly2

### loadassembly2(assemblyname)

Permette di caricare un assembly .NET dal nome parziale del file. Occorre fare attenzione perché non è possibile specificare il framework da utilizzare.

#### Esempio 123. Esempio loadassembly2

```
local asm, err = loadassembly("System.Windows.Forms")
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## loadassembly3



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## loadassembly3

### **loadassembly3(assemblyname)**

Permette di caricare un assembly .NET dal nome esatto del file. In questo caso è possibile utilizzare la firma completa dell'assembly

#### **Esempio 124. Esempio loadassembly3**

```
local asm, err = loadassembly3("")
```

Una volta caricato l'assembly è possibile chiamare dei metodi su di esso.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getassemblies



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## getassemblies

### getassemblies()

Ritorna la lista di oggetti Assembly attualmente caricati nella cache vbluanet.

#### Esempio 125. Esempio getassemblies

```
local assemblies = getassemblies()
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## importtype



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## importtype

### importtype(typename)

Permette di caricare un tipo (typename) cercandolo nella cache degli assemblies caricati. Offre il vantaggio di non dover dichiarare l'assembly da cui caricare il tipo.

#### Esempio 126. Esempio importtype

```
local String = importtype("System.String")
```



#### Nota

Questa funzione è più lenta del caricamento diretto dall'assembly perché deve comunque fare una scansione di tutti i tipi definiti in tutti gli assemblies caricati

Questa funzione permette l'importazione "lazy" dei tipi. In pratica è possibile non dichiarare tutto il namespace: vbluanet è in grado di identificare il tipo e caricarlo automaticamente. In caso di conflitti un messaggio di errore viene generato.

#### Esempio 127. Esempio importtype lazy

```
local String = importtype "String"
```



#### Nota

Questa funzione è più lenta del caricamento passando il namespace lazy. In questo caso devono essere controllati tutti i tpi di tutti gli assembly caricati.

## box



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## box

### **box(luatype, typedescriptor)**

Permette di convertire un tipo valore lua (stringa/number/...) nel corrispondente tipo .NET. Il TypeDescriptor è rappresentato da una stringa che indica il formato di destinazione. I tipi accettati sono Int16, Int32, Int64, Double, Decimal, Single, String, Boolean.

#### **Esempio 138. Esempio box()**

```
local ns1 = box("stringa lua", "String") --> ns1 è una stringa .NET
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

# unbox



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## unbox

### unbox(nettype)

Permette di convertire un tipo valore .NET nel corrispondente tipo Lua. Il tipo viene gestito in modo automatico e i tipi gestiti sono quelli della funzione globale box

#### Esempio 139. Esempio box()

```
local s1 = unbox(ns1) --> s1 è una stringa lua, ns1 è una stringa .NET
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## toclr (obsoleto)



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## toclr (obsoleto)

### toclr(luatype)

Permette di convertire un tipo nativo lua nel corrispondente wrapper NET. Per esempio converte una tabella LUA in una QsLua.LuaTable. Questo comando è obsoleto. Utilizzare la funzione "luatable"

#### Esempio 140. Esempio toclr()

```
local luanettbl = toclr({'a', 'b', 'c'}) --> luanettbl è una  
QsLua.LuaTable.local arr = luanettbl:AsArray() --> ritorna un array di  
stringhe
```

# luatable



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## luatable

### luatable(tbl)

Permette di convertire una tabella LUA in una QsLua.LuaTable. Si può ottenere lo stesso risultato utilizzando il costruttore della LuaTable.

#### Esempio 141. Esempio toclr()

```
local luanettbl = luatable({'a', 'b', 'c'}) --> luanettbl è una  
QsLua.LuaTable.local arr = luanettbl:AsArray() --> ritorna un array di  
stringhe
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## isnettype



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## isnettype

### isnettype(myobj, type, isAssignable)

Verifica se un oggetto è del tipo type. Equivale all' is del Csharp. Il terzo parametro rende stringente l'assegnazione sui tipi derivati

#### Esempio 142. Esempio isnettype()

```
local String = importtype "System.String"
local s = box("ciao", "String")
local isString = isnettype(s, String)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## isqswrapper

### isqswrapper(myobj)

Verifica se un oggetto è un tipo nativo del .NET o un wrapper QS.

#### Esempio 143. Esempio isqswrapper()

```
local String = importtype "System.String"local s = String("ciao")  
local isString = isqswrapper(s) -- <<--- nil
```

## iscomwrapper



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## iscomwrapper

### iscomwrapper(myobj)

Verifica se un oggetto è un wrapper .NET su un oggetto COM.

#### Esempio 144. Esempio iscomwrapper()

```
local String = importtype "System.String"local s = String("ciao")  
local isString = iscomwrapper(s) -- <<--- nil
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getnettype



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## getnettype

### getnettype(myobj)

Ritorna il tipo (come stringa) di un oggetto .net

#### Esempio 145. Esempio getnettype()

```
local s = box("ciao", "String")local tp = getnettype(s)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getluatype



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## getluatype

### getluatype(myobj)

Ritorna il tipo (come stringa) di un oggetto .net oppure lua. La funzione è in grado di elaborare sia i tipi base del lua (nil, string, number, ...) sia gli userdata registrati da QS ritornano il nome del TAG dello userdata, sia il tipo .NET dell'oggetto nel caso si tratti di un oggetto .NET

#### Esempio 146. Esempio getluatype()

```
local s1 = "ciao"local s2 = box(s1, "String")
local tp = getnettype(s1)
local tp = getnettype(s2)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## bitandnet

### **bitandnet(val1, val2, ...)**

Versione NET della funzione bitand. Funziona sia con valori numerici Lua che con Enum (flaggati) .NET. Esegue l'and logico tra i valori passati in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64.

#### **Esempio 147. Esempio bitandnet()**

```
local b = AnchorStyles.Bottomlocal r = AnchorStyles.Right
local t = AnchorStyles.Top
local l = AnchorStyles.Left
local res = bitandnet(b, r, t) --> AnchorStyles.None
```

## bitornet

### bitornet(val1, val2, ...)

Versione NET della funzione bitor. Funziona sia con valori numerici Lua che con Enum (flaggati) .NET. Esegue l'or logico tra i valori passati in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64.

#### Esempio 148. Esempio bitornet()

```
local b = AnchorStyles.Bottomlocal r = AnchorStyles.Right
local t = AnchorStyles.Top
local l = AnchorStyles.Left
local res = bitornet(b, r, t) --> Bottom,Right,Top
```

## **bitxornet**

### **bitxornet(val1, val2, ...)**

Versione NET della funzione bitxor. Funziona sia con valori numerici Lua che con Enum (flaggati) .NET. Esegue l'xor logico tra i valori passati in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **bitnotnet**

### **bitnotnet(val1)**

Funziona sia con valori numerici Lua che con Enum (flaggati) .NET. Esegue il not logico tra i valori passati in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64. Il valore ritornato è Int64.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## bitshlnet

### **bitshlnet(val, count)**

Versione LUA/.NET della funzione <<. Funziona sia con valori numerici Lua che con tipi .NET. Esegue llo shift a sinistra di count posizioni del valore passato in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64.

#### **Esempio 149. Esempio bitshlnet()**

```
local res = bitshlnet(4, 1) --> 8
```

## bitshrnet

### bitshrnet(val, count)

Versione LUA/.NET della funzione >>. Funziona sia con valori numerici Lua che con tipi .NET. Esegue llo shift a destra di count posizioni del valore passato in ingresso. Ritorna lo stesso tipo fornito in ingresso. Le operazioni interne vengono eseguite su UInt64.

#### Esempio 150. Esempio bitshrnet()

```
local res = bitshrnet(4, 1) --> 2
```

## lightcast

### lightcast(val, type)

Questa funzione si utilizza per assicurarsi di avere l'oggetto NET-COM corretto. Il problema si manifesta su oggetti COM che implementano interfacce multiple. Qualche volta il sistema ritorna l'interfaccia non corretta. Per esempio in Inventor "Document" e "PartDocument". Questa funzione assicura di avere in mano l'oggetto corretto.

#### Esempio 151. Esempio lightcast()

```
local PartDocument = asminv:importType("Inventor.PartDocument")...
local doc = occurrence.Definition.Document
local partdoc = lightcast(doc, PartDocument) -- partdoc è sicuramente
del tipo corretto
```

# sizeof



QS Informatica - Manuale LUA » VbLuaNet (VbLuaNet.dll - LuaAsmLdr.dll) » API LUA »

## sizeof

### sizeof(obj)

Ritorna la lunghezza in byte dell'oggetto .NET. Si può passare sia l'oggetto sia il tipo. Non è affidabile al 100%.

#### Esempio 152. Esempio sizeof()

```
local asmcorlib, err = loadassembly2("mscorlib")  
local b =  
    sizeof(mscorlib.Double) -- b ---> 8
```

## convert

### convert(luaval, typecode)

Permette la conversione di un valore lua in un tipo .NET bypassando il sistema di conversione automatico del LUA. Per esempio non è possibile ottenere automaticamente la conversione del valore 3 del lua in un tipo UInt16 perché il vbluanet lo trasformerà sempre in un Int32. E' necessario passare il valore LUA e la stringa del TypeCode .NET. Possibili valori sono: "Boolean", "Byte", "Char", "DateTime", "DBNull", "Decimal", "Double", "Empty", "Int16", "Int32", "Int64", "Object", "SByte", "Single", "String", "UInt16", "UInt32", "UInt64". Per i booleani si può passare nil/<qualsiasi cosa>, oppure "true"/"false"

#### Esempio 153. Esempio convert()

```
local v = 3
local b = convert(v, "UInt16")
local x = convert(1, "Boolean")    -- <<---- True
local x = convert(nil, "Boolean") -- <<---- False
```

## astable



QS Informatica - Manuale LUA » XTable (Extended Tables) » Service »

## astable

### <lst>:astable()

Ritorna la table "normale" contenuta nella XTable. Normalmente viene chiamata alla fine di ogni query.

#### Esempio 204. Esempio astable

```
local a = XTable(Orders) :where(function(x, i) return x.ShipCountry ==  
"USA" and x.CustomerID == "OLDW0" end)  
:astable()  
alert(a)
```

## saveastable



QS Informatica - Manuale LUA » XTable (Extended Tables) » Service »

## saveastable

**<lst>:saveastable(fname)**

Esegue la serializzazione della tabella su disco. Richiede il nome del file.

### Esempio 205. Esempio saveastable

```
local orders = { {id=2, product="Computer", price=1000, year=2010},  
  {id=2, product="Book",      price=50,    year=2012},  
  {id=2, product="TV",       price=800,   year=2011},  
}  
XTable(orders)  
:saveastable("C:\\temp\\orders.txt")
```

# help



QS Informatica - Manuale LUA » XTable (Extended Tables) » Service »

## help

<lst>:**help()**

Ritorna l'elenco degli operatori disponibili e la loro descrizione

### Esempio 206. Esempio astable

local a = XTable(Orders)	: <b>help()</b>
--------------------------	-----------------

Copyright © <Dates> by <Authors>. All Rights Reserved.

## where



QS Informatica - Manuale LUA » XTable (Extended Tables) » Restriction »

## where

**<lst>:where(predicate(item, pos)**

Itera il contenuto della lista ritornando gli elementi che soddisfano il predicato in ingresso.

### Esempio 207. Esempio where

```
local a = XTable(Orders) :where(function(x, i) return x.ShipCountry ==  
"USA" and x.CustomerID == "OLDWO" end)  
:astable()  
alert(a)
```

## where2



QS Informatica - Manuale LUA » XTable (Extended Tables) » Restriction »

## where2

**<map>:where2(predicate(key, val))**

Itera il contenuto della tabella ritornando gli elementi che soddisfano il predicato in ingresso.

### Esempio 208. Esempio where2

```
local codici = {[ 'C001' ] = 'C001', [ 'A002' ] = 'A002', [ 'D003' ] = 'D003',
[ 'A004' ] = 'A004', [ 'F005' ] = 'F005', [ 'B006' ] = 'B006', [ 'C007' ] =
'C007', }

local a = XTable(codici)
:where2(function(k, v) return substr(k, 0, 1) == "A" end)
:astable()
alert(a)
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## select



QS Informatica - Manuale LUA » XTable (Extended Tables) » Projection »

### select

```
<lst>:select(field)
<lst>:select({field1, field2})
<lst>:select(foo(item, pos))
```

Itera il contenuto della lista ritornando nuovi elementi costruiti dal selettore. Nel primo caso i membri stessi della lista che hanno chiave "field", nel secondo caso i membri della lista che hanno chiave uguale ad uno dei fields nella lista in ingresso. Nell'ultimo caso il valore di ritorno dalla funzione foo che passa in ingresso l'elemento corrente e la posizione corrente nella lista.

#### Esempio 209. Esempio select

```
-- ritorna una nuova lista con nome e anno
a = XTable({{nome="massimo",
cognome="celino", anno=1971},
            {nome="daniele", cognome="piazza", anno=1971},
            {nome="oreste", cognome="montanari", anno=1991}})

:select({"nome", "anno"})
:astable()
-->{{nome="massimo", anno=1971},
    {nome="daniele", anno=1971},
    {nome="oreste", anno=1991}}
```

#### Esempio 210. Esempio select

```
-- ritorna una nuova lista con codice e prezzo totale (qta * pru) a
partire da un bom di TOP
b = XTable(bom)
-- codice->[1], qta->[3], pru->[7]
:select(function(record, i) return {codice=record[1],
totprice=record[3]*record[7]}; end)
:astable()
```

## select2



QS Informatica - Manuale LUA » XTable (Extended Tables) » Projection »

## select2

**<map>:select2(foo(key, val))**

Itera il contenuto della tabella per chiave e valore ritornando nuovi elementi costruiti dal selettore.

### Esempio 211. Esempio select2

```
local orders = { {orderid=1, product="Computer", price=1000,
year=2010},
{orderid=1, product="Book",      price=50,     year=2012},
{orderid=3, product="TV",       price=800,    year=2011},
...
}
-- ritorna i tre anni più remunerativi
local res = XTable(orders):groupby("year")
:select2(function(k, v) return {year=k,
totalprice=XTable(v):sum('price')} end)
:orderbyascending('totalprice')
:take(3)
:astable()
```

## selectmany



QS Informatica - Manuale LUA » XTable (Extended Tables) » Projection »

## selectmany

<map>:selectmany(foo(item, pos))

Itera il contenuto della lista ritornando nuovi elementi costruiti dal selettore. Gli elementi di ogni iterazione vengono accorpati nella lista globale

### Esempio 212. Esempio selectmany

```
local a = {"Daniele Piazza", "Massimo Celino", "Oreste Montanari"}local  
b = XTable(a)  
  :selectmany(function(fullname) return strSplitDelim(fullname, " ", "-"  
trim"); end)  
  :astable()  
--> {"Daniele", "Piazza", "Massimo", "Celino", "Oreste", "Montanari"}
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## skip



QS Informatica - Manuale LUA » XTable (Extended Tables) » Partitioning »

### skip

<lst>:skip(n)

Scarta un numero dato di elementi dalla lista

#### Esempio 213. Esempio skip

```
-- ritorna una nuova lista con nome e anno
a = XTable({{nome="massimo",
cognome="celino", anno=1971},
            {{nome="daniele", cognome="piazza", anno=1971},
            {{nome="oreste", cognome="montanari", anno=1991}}})
:skip(2)
:astable()
--> {{nome="oreste", cognome="montanari", anno=1991}}
```

## skipwhile



QS Informatica - Manuale LUA » XTable (Extended Tables) » Partitioning »

## skipwhile

<lst>:skipwhile(predicate(item, pos))

Scarta gli elementi della lista finché il predicato è vero.

### Esempio 214. Esempio skipwhile

```
local orders = { {orderid=1, product="Computer", price=1000,
year=2010},
{orderid=3, product="TV",           price=800,   year=2011},
{orderid=1, product="Book",        price=50,    year=2012},
}
local res = XTable(orders)
    :skipwhile(function(item) return item.year<2011; end)
    :astable()
--> {{orderid=3, product="TV", price=800,   year=2011},
{orderid=1, product="Book", price=50,    year=2012}}
```

## take



QS Informatica - Manuale LUA » XTable (Extended Tables) » Partitioning »

### take

**<lst>:take(n)**

Ritorna i primi n elementi della lista

#### Esempio 215. Esempio take

```
local a = {"Daniele Piazza", "Massimo Celino", "Oreste Montanari"}local  
b = XTable(a):take(2):astable()  
--> {"Daniele", "Piazza", "Massimo", "Celino"}
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## takewhile



QS Informatica - Manuale LUA » XTable (Extended Tables) » Partitioning »

## takewhile

**<lst>:takewhile(predicate(item, pos))**

Prende gli elementi della lista finché il predicato è vero.

### Esempio 216. Esempio takewhile

```
local orders = { {orderid=1, product="Computer", price=1000,
year=2010},
{orderid=3, product="TV",           price=800,   year=2011},
{orderid=1, product="Book",        price=50,    year=2012},
}
local res = XTable(orders)
:takewhile(function(item) return item.year<2011; end)
:astable()
--> {{orderid=1, product="Computer", price=1000, year=2010}
```

## join



QS Informatica - Manuale LUA » XTable (Extended Tables) » Join »

## join

<seq1>:join(seq2, predicate(seq1.item, seq2.item), alias="table2")

Esegue l'inner join delle due sequenze basato sul predicato di matching. Se le tabelle sono chiave valore e ci sono delle collisioni i campi della prima tabella vengono preservati mentre quelli della seconda prendono l'alias (es. "ID" --> "Ordini.ID"). Se non passato l'alias vale "table2".

Nel caso di tabelle sequenziali i campi della seconda tabella vengono accodati alla prima. Se le tabelle non sono rettangolari non sono facilmente gestibili.

### Esempio 217. Esempio join

```
local a = XTable(Orders) :where(function(x, i) return x.ShipCountry ==  
"USA" and x.CustomerID == "OLDWO" end)  
  :join(Customers, function(order, customer) return order.CustomerID ==  
order.CustomerID end)  
  :select({ "ContactName", "CompanyName", "OrderID", "ShipCountry"})  
  :astable()  
alert(a)
```

## concat



QS Informatica - Manuale LUA » XTable (Extended Tables) » Concatenation »

## concat

<lst>:concat(lst2)

Concatena gli elementi delle lista passata a quelli della lista corrente.

### Esempio 218. Esempio concat

```
local a = {"a", "b", "c"}  
local b = XTable(a):concat({"d", "e", "f"},  
{"g", "h", "i"}):astable()  
--> {"a", "b", "c", "d", "e", "f", "g", "h", "i"}
```

Copyright © <Dates> by <Authors>. All Rights Reserved.

## orderby



QS Informatica - Manuale LUA » XTable (Extended Tables) » Ordering »

## orderby

```
<lst>:orderby(field)
<lst>:orderby({field1, field2, ...})
<lst>:orderby(foo(item1, item))
```

Ordina gli elementi della lista. Nel primo caso esegue un sort ascendente sul valore della chiave passata. Nel secondo caso esegue un sort ascendente sulla prima chiave della lista, in caso di uguaglianza passa alla seconda e così via. Nell'ultimo caso è possibile passare la funzione di comparazione (comparer)

### Esempio 219. Esempio orderby

```
local orders = { {id=1, product="Computer", price=1000, year=2010},
  {id=1, product="Book",      price=50,    year=2012},
  {id=3, product="TV",       price=800,   year=2011},
  ...
}
XTable(orders):orderby("year"):astable()
XTable(orders):orderby({"id", "price"}):astable()
XTable(orders):orderby(function(a, b) return a.year<b.year
end):astable()
```

## orderbydescending



QS Informatica - Manuale LUA » XTable (Extended Tables) » Ordering »

### orderbydescending

```
<lst>:orderbydescending(field)
<lst>:orderbydescending({field1, field2, ...})
<lst>:orderbydescending(foo(item1, item))
```

Come orderby discendente

Copyright © <Dates> by <Authors>. All Rights Reserved.

## orderbyascending



QS Informatica - Manuale LUA » XTable (Extended Tables) » Ordering »

### orderbyascending

```
<lst>:orderbyascending(field)
<lst>:orderbyascending({field1, field2, ...})
<lst>:orderbyascending(foo(item1, item))
```

Come orderby come default (ascendente).

Copyright © <Dates> by <Authors>. All Rights Reserved.

## reverse



QS Informatica - Manuale LUA » XTable (Extended Tables) » Ordering »

### reverse

#### <lst>:reverse()

Inverte l'ordine degli elementi della lista.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## groupby

<lst>:groupby(key)

<lst>:groupby(foo(item, pos))

Raggruppa gli elementi della lista. Ritorna un dizionario che ha come chiave i valori della key e come valore i risultati del raggruppamento. Nel secondo caso la funzione 'foo' deve ritornare la chiave di raggruppamento.

### Esempio 220. Esempio groupby

```
local orders = { {id=1, product="Computer", price=1000, year=2010},  
    {id=1, product="Book",      price=50,     year=2012},  
    {id=3, product="TV",       price=800,    year=2011},  
    {id=4, product="Manga",    price=5,      year=2010},  
    ...  
}  
-- ritorna i tre anni più remunerativi  
local a = XTable(orders)  
    :groupby("year")      -- [*]  
    :select2(function(k, v) return {year=k,  
totalprice=XTable(v):sum('price')} end)  
    :orderbyascending('totalprice')  
    :take(3)  
    :astable()  
--[*]  
--> {[ '2010' ] = {{id=1, product="Computer", price=1000, year=2010},  
                      {id=4, product="Manga",    price=5,     year=2010}},  
          [ '2011' ] = {{id=3, product="TV",       price=800,    year=2011}},  
          [ '2012' ] = {{id=1, product="Book",      price=50,     year=2012}}},  
    }
```

## distinct



QS Informatica - Manuale LUA » XTable (Extended Tables) » Set »

## distinct

<lst>:distinct()

<lst>:distinct(foo(item, pos))

Rimuove gli elementi duplicati. Utilizza la funzione di comparazione del lua (==) nella prima versione. Attenzione: le tabelle non vengono confrontate per contenuto ma per riferimento. Quindi due tabelle identiche per contenuto risultano diverse per il comparatore. Nel secondo caso è possibile passare la funzione che genera una key univoca per il confronto degli elementi.

### Esempio 221. Esempio distinct

```
alert(XTable({"a", "a", "b", "c", "d", "d", "d",  
"e"}):distinct():astable())--> {"a", "b", "c", "d", "e"}  
local orders = {  
    {id=1, product="Computer", price=1000, year=2010},  
    {id=1, product="Book",     price=50,    year=2012},  
    {id=3, product="TV",      price=800,   year=2011},  
    {id=5, product="Computer", price=1000, year=2010},  
    ...  
}  
  
local a = XTable(orders)  
    -- rimuove gli elementi duplicati ignorando l'id  
    :distinct(function(item, pos) return format("%s-%d-%d",  
item.product, item.price, item.year) end)  
    :astable()  
--> {  
    {id=1, product="Book",     price=50,    year=2012},  
    {id=3, product="TV",      price=800,   year=2011},  
    {id=5, product="Computer", price=1000,  year=2010},  
}
```

## intersect



QS Informatica - Manuale LUA » XTable (Extended Tables) » Set »

## intersect

**<lst>:intersect(lst1, lst2, ..., lstn, [foo(item, pos)])**

Esegue l'intersezione di tutte le liste. Mantiene solo gli elementi presenti in tutte le liste. Attenzione perché utilizza la comparazione del lua e quindi non funziona per le tabelle che vengono passate per riferimento.

Nel secondo caso è possibile passare la funzione che genera una key univoca per il confronto degli elementi.

### Esempio 222. Esempio intersect

```
local a = {"a", "a", "b", "e"}a = XTable(a):intersect({"a", "b", "e"},  
{"a", "e"}):astable()  
--> {"a", "e"}  
local orders = {  
    {id=2, product="Computer", price=1000, year=2010},  
    {id=2, product="Book",     price=50,   year=2012},  
    {id=2, product="TV",      price=800,  year=2011},  
}  
local orders1 = {  
    {id=4, product="Computer", price=1000, year=2010},  
    {id=4, product="Book",     price=50,   year=2012},  
}  
local x = (XTable(orders)  
            :intersect(orders1,  
                        function(item, pos) return format("%s-%d-%d",  
item.product, item.price, item.year) end)  
            :astable()  
--> {{id=4, product="Computer", price=1000, year=2010},  
     {id=4, product="Book",     price=50,   year=2012}}}
```

## union



QS Informatica - Manuale LUA » XTable (Extended Tables) » Set »

## union

**<lst>:union(lst1, lst2, ..., lstn, [foo(item, pos)])**

Esegue l'unione di tutte le liste. Sovrascrive gli elementi identici. Attenzione perché utilizza la comparazione del lua e quindi non funziona per le tabelle che vengono passate per riferimento.

Nel secondo caso è possibile passare la funzione che genera una key univoca per il confronto degli elementi.

### Esempio 223. Esempio union

```
local a = {"e"}a = XTable(a):union({"a", "b", "e"}, {"a",  
"e"}):astable()  
--> {"e", "a", "b"}  
local orders = {  
    {id=2, product="Computer", price=1000, year=2010},  
    {id=2, product="Book",     price=50,   year=2012},  
}  
local orders1 = {  
    {id=4, product="Computer", price=1000, year=2010},  
    {id=4, product="Book",     price=50,   year=2012},  
    {id=4, product="TV",       price=800,  year=2011},  
}  
local x = (XTable(orders)  
            :union(orders1,  
                  function(item, pos) return format("%s-%d-%d",  
item.product, item.price, item.year) end)  
            :astable()  
--> {{id=2, product="Computer", price=1000, year=2010},  
     {id=2, product="Book",     price=50,   year=2012},  
     {id=4, product="TV",       price=800,  year=2011}}}
```

## count



QS Informatica - Manuale LUA » XTable (Extended Tables) » Aggregate »

### count

<seq>:count([predicate(item, pos)])

Conta gli elementi della lista (come il getn del lua). Se viene passato il predicato, conta solo gli elementi che soddisfano il predicato.

#### Esempio 224. Esempio count

```
local a = {"a", "a", "b", "e"}local res = XTable(a):count(function(item,  
i) return item=="a"; end)  
--> 2  
local orders = {  
{id=1, product="Computer", price=1000, year=2010},  
{id=1, product="Book", price=50, year=2012},  
{id=3, product="TV", price=800, year=2011},  
{id=5, product="Computer", price=1000, year=2010},  
...  
}  
local res = XTable(a):count(function(item, i) return  
item.product=="Computer"; end)  
--> 2
```

## sum



QS Informatica - Manuale LUA » XTable (Extended Tables) » Aggregate »

### sum

<seq>:sum()  
<seq>:sum(field)  
<seq>:sum(function(item, pos))

Esegue la somma degli elementi della lista. Se non viene passato nulla somma solo gli elementi di tipo numerico. Se viene passata una stringa allora si intende il nome del campo dell'item della lista. Se viene passata una funzione allora il valore di ritorno dalla funzione viene sommato.

#### Esempio 225. Esempio sum

```
local a = {"a", 1, "b", 2}local res = XTable(a):sum()
--> 3
local orders = {
    {id=1, product="Computer", price=1000, year=2010},
    {id=1, product="Book",     price=50,   year=2012},
    {id=3, product="TV",      price=800,  year=2011},
    {id=5, product="Computer", price=1000, year=2010},
    ...
}
local res = XTable(orders):sum('price')
--> 2850
local res = XTable(orders)
:sum(function(item, pos)
    return item.product=="Computer" and item.price or 0;
end)
--> 2000
```

## max

<seq>:max()  
<seq>:max(field)  
<seq>:max(function(item, pos))

Estrae il max degli elementi della lista. Se non viene passato nulla confronta solo gli elementi di tipo numerico. Se viene passata una stringa allora si intende il nome del campo dell'item della lista. Se viene passata una funzione allora il valore di ritorno dalla funzione viene utilizzato per il calcolo del max.

### Esempio 226. Esempio max

```
local a = {"a",1,"b",2}local res = XTable(a):max()
--> 2
local orders = {
  {id=1, product="Computer", price=1100, year=2010},
  {id=1, product="Book",      price=50,   year=2012},
  {id=3, product="TV",       price=800,  year=2011},
  {id=5, product="Computer", price=1000, year=2010},
  ...
}
local res = XTable(orders):max('price')
--> 1000
local res = XTable(orders)
  :max(function(item, pos) return item.product=="Computer" and
item.price; end)
--> 1100
```

## min



QS Informatica - Manuale LUA » XTable (Extended Tables) » Aggregate »

### min

**<seq>:min()**

**<seq>:min(field)**

**<seq>:min(function(item, pos))**

Come max... più un po' di immaginazione.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## fold



QS Informatica - Manuale LUA » XTable (Extended Tables) » Altre »

## fold

**<lst>:fold(foo(x, item, pos), initval)**

Applica foo su tutti gli elementi della lista passando l'accumulatore x (valore iniziale "initval") e l'elemento corrente.

### Esempio 227. Esempio fold

```
local a = {"The", "quick", "brown", "fox", "jumps", "over", "the",
"lazy", "dog"}local b = XTable(a)
  :fold(function(sentence, word) return sentence .. " " .. word; end,
"")
--> "The quick brown fox jumps over the lazy dog"
local orders = {
  {id=2, product="Computer", price=1000, year=2010},
  {id=2, product="Book",     price=50,    year=2012},
  {id=2, product="TV",      price=800,   year=2011},
}
local b = XTable(orders)
  :fold(function(x, item, i) return x + item.price; end, 1000))
--> 2850
```

## unfold

**<lst>:unfold(foo(x, item, pos), initval)**

Riempie lst con il risultato della computazione di foo. foo viene eseguita finché ritorna un valore diverso da nil. foo riceve in ingresso il valore della computazione precedente o initval.

### Esempio 228. Esempio unfold

```
local a = XTable({})  :unfold(function(x) return (x <= 10) and x+1;  
end, 0)  
:astable()  
-->{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  
-- calcola la sequenza di Fibonacci (valori minori di 1000)  
local a = XTable({})  
:unfold(function(x)  
    if x[2] > 1000 then return end  
    return {x[1]+ x[2], x[2] + x[1] + x[2]}  
  end, {1, 1})  
:selectmany(function(tuple) return tuple end)  
:astable()  
--> {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987}
```

# map



QS Informatica - Manuale LUA » XTable (Extended Tables) » Altre »

## map

**<lst>:map(foo(item, pos))**

Ritorna una lista con il risultato dell'applicazione di foo a tutti gli elementi della lista lst.

### Esempio 229. Esempio map

```
local a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}local b = XTable(a)
:map(function(item, i) return (mod(i, 2) == 0) and item end) -- map
keeps nil!!
:where(function(item, i) return item end) -- remove nils
:astable()
{2, 4, 6, 8, 10}
```

## map2

**<tbl>:map2(foo(k, v))**

Ritorna una lista con il risultato dell'applicazione di foo a tutti gli elementi della lista tabella

Copyright © <Dates> by <Authors>. All Rights Reserved.

## iter

<lst>:**iter(foo(item, pos))**

Applica la funzione passata a tutti gli elementi della lista senza accumulare nulla.  
Ritorna se stessa inalterata.

### Esempio 230. Esempio iter

```
local a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}local b = XTable(a)
:iter(function(item, i) alert(item, i) end) -- alert for debug
:where(function(item, i) return item < 5 end)
:astable()
{1, 2, 3, 4}
```

## iter2

**<map>:iter2(foo(k, v))**

Applica la funzione passata a tutti gli elementi della lista senza accumulare nulla.  
Ritorna se stessa inalterata.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## find



QS Informatica - Manuale LUA » XTable (Extended Tables) » Altre »

## find

**<lst>:find(foo(item, pos))**

Ritorna il primo elemento per cui la funzione passata vale true. Gli elementi successivi non vengono valutati.

### Esempio 231. Esempio find

```
local a = {"paperino", "pippo", "pluto", "minnie"}local b = XTable(a)
:find(function(item, i) return strlen(item) == 5 end)
--> "pippo"
```

## fromlol

### <--->:**fromlol(lol)**

Costruisce e restituisce una xtable basata sulla lol. Da notare che modifica la struttura della tabella interna trasformando ogni singola riga della lol da una lista in una mappa nome-valore. In questo modo è possibile eseguire delle query in modo più semplice.

#### Esempio 232. Esempio fromlol

```
local lol = { {"nome", "cognome", "anno"},  
  {"donald", "duck", 1934},  
  {"mickey", "mouse", 1928},  
}  
local b = XTable()  
  :fromlol(lol)  
--> {  
  {nome="donald", cognome="duck", anno=1934},  
  {nome="mickey", cognome="mouse", anno=1928},  
}
```

## aslol

### <lol>:aslol(headers)

Costruisce e restituisce una lol a partire dal contenuto interno. Il contenuto deve essere stato generato da una lol. Riceve in ingresso la lista delle intestazioni per poter restituire la lol con le colonne ordinate correttamente. Se gli headers non vengono passati la struttura della tabella viene composta automaticamente.

#### Esempio 233. Esempio aslol

```
local lol = { {"nome", "cognome", "anno"},  
  {"donald", "duck", 1934},  
  {"mickey", "mouse", 1928},  
}  
local b = XTable()  
:fromlol(lol)  
:where(function(x, i) return x.anno < 1930 end)  
:aslol(lol[1])  
--> {"nome", "cognome", "anno"},  
  {"mickey", "mouse", 1928},
```

## saveaslol

**<lol>:saveaslol(fname, headers)**

Esegue la serializzazione della lol su disco. Richiede il nome del file e la lista delle intestazioni. Se le intestazioni non sono passate vengono salvate in modo automatico (casuale)

### Esempio 234. Esempio saveaslol

```
local lol = { {"nome", "cognome", "anno"},  
  {"donald", "duck", 1934},  
  {"mickey", "mouse", 1928},  
}  
XTable()  
:fromlol(lol)  
:saveaslol("C:\\temp\\lol.txt", lol[1])
```

## getinfo

### **getinfo (FUNCTION, [WHAT])**

Returns a table with information about a function. FUNCTION can also be a level number, relative to getinfo (level 0). If the number is invalid, a nil is returned. The returned table contains all the fields returned by lua\_getinfo. The default for WHAT is to get all information available.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## getlocal

### getlocal (LEVEL, LOCAL)

Returns the name and the value of the local variable with index LOCAL of the function at level level of the stack. Similar to `lua_getlocal`. Returns nil if LOCAL is out of range, and raises an error when called with a LEVEL that is out of range.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **setlocal**



QS Informatica - Manuale LUA » Lua 4 » Debug - The Reflexive Debug Interface »

## **setlocal**

### **setlocal (LEVEL, LOCAL, VALUE)**

Assigns VALUE to the local variable with index LOCAL of the function at level LEVEL of the stack. Returns nil if LOCAL is out of range, and raises an error when called with a LEVEL out of range.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setcallhook

### setcallhook (HOOK)

Sets the function HOOK as the call hook, and returns the old hook. The only argument to the call hook is the event name (“call” or “return”).) Use getinfo with level 2 to get more information about the function being called or returning (level 0 is getinfo, level 1 is the hook.)Without arguments, call hooks are turned off.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## setlinehook

### setlinehook (HOOK)

Sets the function HOOK as the line hook, and returns the old hook. The only argument to the line hook is the line number the interpreter is about to execute. Without arguments, line hooks are turned off.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## lua\_open

**lua\_State \*lua\_open (int stacksize);**

If stacksize is zero, then a default size of 1024 is used. Each function call needs one stack position for each argument, local variable, and temporary value, plus one position for book-keeping. The stack must also have some 20 extra positions available.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **lua\_close**



QS Informatica - Manuale LUA » The Lua C API » Lua States »

### **lua\_close**

**void lua\_close (lua\_State \*L);**

Destroys all objects in the given Lua environment (calls garbage-collection tag methods, if any) and frees all dynamic memory used by that state. Usage is not compulsory, because all resources are released when your program ends. Useful for long-running applications such as daemons or servers.

Copyright © <Dates> by <Authors>. All Rights Reserved.

## printf - format

QS Informatica - Manuale LUA » Format Placeholders »



## printf - format

Formatting takes place via placeholders within the format string. The syntax for a format placeholder is:

```
"%[parameter][flags][width][.precision][length]type"
```

See the syntax tokens

Copyright © <Dates> by <Authors>. All Rights Reserved.

## **date - format**

QS Informatica - Manuale LUA » Format Placeholders »



## date - format

Returns the date

```
date ([FORMAT])
```

The syntax for a format placeholder is:

%a	The abbreviated weekday name according to the current locale.	Wed
%A	The full weekday name according to the current locale.	Wednesday
%b	The abbreviated month name according to the current locale.	Jan
%B	The full month name according to the current locale.	January
%c	The preferred date and time representation for the current locale.	
%C	The century number (year/100) as a 2-digit integer.	19
%d	The day of the month as a decimal number (range 01 to 31).	7
%D	Equivalent to %m/%d/%y. (This is the USA date format. In many countries %d/%m/%y is the standard date format. Thus, in an international context, both of these	12/25/04

	formats are ambiguous and should be avoided.)	
%e	Like %d, the day of the month as a decimal number, but a leading zero is replaced by a space.	7
%G	The ISO 8601 year with century as a decimal number. The 4-digit year corresponding to the ISO week number (see %V). This has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, that year is used instead.	2004
%g	Like %G, but without century, i.e., with a 2-digit year (00-99).	4
%h	Equivalent to %b.	Jan
%H	The hour as a decimal number using a 24-hour clock (range 00 to 23).	22
%I	The hour as a decimal number using a 12-hour clock (range 01 to 12).	7
%j	The day of the year as a decimal number (range 001 to 366).	8
%k	The hour (24-hour clock) as a decimal number (range 0 to 23); single digits are preceded by a blank.	7

	(See also %H.)	
%l	The hour (12-hour clock) as a decimal number (range 1 to 12); single digits are preceded by a blank. (See also %I.)	7
%m	The month as a decimal number (range 01 to 12).	4
%M	The minute as a decimal number (range 00 to 59).	58
%n	A newline character.	
%p	Either `AM' or `PM' according to the given time value, or the corresponding strings for the current locale. Noon is treated as `pm' and midnight as `am'.	AM
%P	Like %p but in lowercase: `am' or `pm' or a corresponding string for the current locale.	am
%r	The time in a.m. or p.m. notation. In the POSIX locale this is equivalent to `%I:%M:%S %p'.	7.58.40 AM
%R	The time in 24-hour notation (%H:%M). For a version including the seconds, see %T below.	7.58
%s	The number of seconds since the Epoch, i.e.,	1099928130

	since 1970-01-01 00:00:00 UTC.	
%S	The second as a decimal number (range 00 to 61). the upper level of the range 61 rather than 59 to allow for the occasional leap second and even more occasional double leap second.	7
%t	A tab character.	
%T	The time in 24-hour notation (%H:%M:%S).	17.58.40
%u	The day of the week as a decimal, range 1 to 7, Monday being 1. See also %w.	3
%U	The week number of the current year as a decimal number, range 00 to 53, starting with the first Sunday as the first day of week 01. See also %V and %W.	26
%V	The ISO 8601:1988 week number of the current year as a decimal number, range 01 to 53, where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. See also %U and %W.	26

%w	The day of the week as a decimal, range 0 to 6, Sunday being 0. See also %u.	5
%W	The week number of the current year as a decimal number, range 00 to 53, starting with the first Monday as the first day of week 01.	34
%x	The preferred date representation for the current locale without the time.	
%X	The preferred time representation for the current locale without the date.	4
%y	The year as a decimal number without a century (range 00 to 99).	2004
%z	The time-zone as hour offset from GMT. Required to emit RFC822-conformant dates (using "%a, %d %b %Y %H:%M:%S %z").	-2
%Z	The time zone or name or abbreviation.	GMT
%%	A literal '%' character.	%