* sys Module PREDICATION

  A sys module is a set of the library module of the
standard included in the system. When calling,
it describes after ::sys.

::sys <args VAR>

        ARG when starting the Descartes language is
        assigned to VAR.

::sys <avg VAR LIST>
::sys <avgf VAR LIST>

        The average value of LIST element is set as a VAR variable.
        LIST element is calculated after being evaluated.
        avg is calculated integrally.
        avgf is calculated by a floating point number.

::sys <basename VAR PATH>

        A file name is extracted from a path and
        it is set as a variable.

::sys <dirname VAR PATH>

        A directory name is extracted from a path and
        it is set as a variable.

::sys <suffix VAR PATH SUFFIX>

        The suffix of a path is changed into the SUFFIX of an argument.

::sys <clear>

        A screen is cleared.

::sys <DLIBPATH VAR>

        The path DLIBPATH which the Descartes language
        uses is displayed on VAR.

::sys <mkpred ARG>

        ARG is changed into PRED.

::sys <writenl LIST>

        After outputting LIST, a new line is started.

::sys <write LIST>
::sys <w LIST>

Outputting LIST.

::sys <isNil ARG>
::sys <isAtom ARG>
::sys <isList ARG>
::sys <isPred ARG>
::sys <isVar ARG>
::sys <isUndefVar ARG>
::sys <isFloat ARG>
::sys <isInteger ARG>
::sys <isInf ARG>
::sys <isNan ARG>

true will be returned, if ARG is judged and it corresponds. unknown is returned if it does not correspond.

::sys <isTrue PRED>
::sys <isFalse PRED>
::sys <isUnknown PRED>

The result of PRED of ARG is judged, and true will be returned if it corresponds. unknown is returned if it does not correspond.

::sys <max VAR LIST>

The greatest integer value of the element of a list is set as a variable.

::sys <min VAR LIST>

The minimum integer value of the element of a list is set as a variable.

::sys <maxf VAR LIST>

The greatest floating point number value of the element of a list is set as a variable.

::sys <minf VAR LIST>

The minimum floating point number value of the element of a list is set as a variable.

::sys <regex PATTERN STRINGS BEFORE-STRINGS MATCH-STRINGS AFTER-STRINGS>

The result of having applied the regular expression pattern to STRINGS is set as Match STRINGS, and STRINGS of order is set as front STRINGS and back STRINGS. (It does not operate on Windows.)

::sys <sub PATTERN STRINGS CORRESDED-STRINGS AFTER-STRINGS>

The result of having transposed the portion which corresponded to STRINGS with the application of the regular expression pattern to Substitution STRINGS is set as Output STRINGS. Replacement

is performed only once.
(It does not operate on Windows.)

::sys <gsub PATTERN STRINGS CORRESDED-STRINGS AFTER-STRINGS>

The result of having transposed the portion which corresponded to STRINGS with the application of the regular expression pattern to Substitution STRINGS is set as Output STRINGS. Replacement is performed into all the applicable portions of STRINGS. (It does not operate on Windows.)

::sys <leftstr VAR STR LENGTH>

The character string of length is cut out from the left of a character string, and it is set as a variable.

::sys <rightstr VAR STR LENGTH>

The character string of length is cut out from the right of a character string, and it is set as a variable.

::sys <substr VAR STR POS LENGTH>

The character string of length is cut out from the position of a character string, and it is set as a variable.

::sys <insertstr VAR STR1 POS STR2>

The character string 2 is inserted in the position of the character string 1.

::sys <real VAR COMPLEX>

The real part of a complex number is set as a variable.

::sys <image VAR COMPLEX>

The imaginary part of a complex number is set as a variable.

::sys <arg VAR COMPLEX>

angle of a complex number is set as a variable.

::sys <norm VAR COMPLEX>

The square of a real part and an imaginary number part sum total is set as a variable.

::sys <conj VAR COMPLEX>

The mark of the imaginary number part of a complex number is reversed, and a conjugate complex number is set as a variable.

::sys <polar VAR ABSOLUTE-VALUE ANGLE>

The complex number by a polar coordinate system is set as a variable.

::sys <split VAR STRINGS [DELIMITER]>

What divided STRINGS by the DELIMITER and was
set to LIST is set as VAR. When the delimiter
is not specified, it is divided with a blank
and a tab.

::sys <sum VAR LIST>
::sys <sumf VAR LIST>

The element of a list is totaled and it is set as a variable.
A function predicate can also be described to the element of a list.
sum is totaled as an integer.
sumf is totaled as a floating point number.

::sys <toupper VAR STRINGS>

A character string is made into a capital letter.

::sys <tolower VAR STRINGS>

A character string is made into a small letter.

::sys <length VAR LIST>

The length of LIST is set as VAR.

::sys <random VAR>

A random number is assigned to VAR.

::sys <PI VAR>

The value of a circular constant PI is set as a variable.

::sys <sin VAR RADIAN>
::sys <cos VAR RADIAN>
::sys <tan VAR RADIAN>

Trigonometric functions

::sys <asin VAR VAL>
::sys <acos VAR VAL>
::sys <atan VAR VAL>
::sys <atan2 VAR VAL1 VAL2>

Inverse trigonometric function

::sys <sinh VAR RADIAN>
::sys <cosh VAR RADIAN>
::sys <tanh VAR RADIAN>

Hyperbolic trigonometric functions

::sys <asinh VAR VAL>
::sys <acosh VAR VAL>
::sys <atanh VAR VAL>

Hyperbolic inverse trigonometric function

::sys <e VAR>

The bottom e of a natural logarithm e is set as a variable.

::sys <log VAR VAL>
::sys <log10 VAR VAL>
::sys <exp VAR VAL>
::sys <exp2 VAR VAL>
::sys <exp10 VAR VAL>
::sys <pow VAR VAL1 VAL2>

Logarithmic function

::sys <sqrt VAR VAL>

Square root

::sys <abs VAR VAL>

Absolute value

::sys <int VAR VAL>

Integral value

::sys <ceil VAR VAL>

smallest integral value not less than argument

::sys <floor VAR VAL>

largest integral value not greater than argument

::sys <trunc VAR VAL>

round to interger, towards zero

::sys <car VAR LIST>
::sys <cdr VAR LIST>

car, cdr of LIST

::sys <caar VAR LIST>

car(car(LIST))

::sys <cadr VAR LIST>

car(cdr(LIST))

::sys <cdar VAR LIST>

```
        cdr(car(LIST))
::sys <cddr VAR LIST>
        cdr(cdr(LIST))

::sys <caaar VAR LIST>
        car(car(car(LIST)))
::sys <caadr VAR LIST>
        car(car(cdr(LIST)))
::sys <cadar VAR LIST>
        car(cdr(car(LIST)))
::sys <caddr VAR LIST>
        car(cdr(cdr(LIST)))
::sys <cdaar VAR LIST>
        cdr(car(car(LIST)))
::sys <cdadr VAR LIST>
        cdr(car(cdr(LIST)))
::sys <cddar VAR LIST>
        cdr(cdr(car(LIST)))
::sys <cdddr VAR LIST>
        cdr(cdr(cdr(LIST)))

::sys <caaaar VAR LIST>
        car(car(car(car(LIST))))
::sys <caaadr VAR LIST>
        car(car(car(cdr(LIST))))
::sys <caadar VAR LIST>
        car(car(cdr(car(LIST))))
::sys <caaddr VAR LIST>
        car(car(cdr(cdr(LIST))))
```

::sys <cadaar VAR LIST>

car(cdr(car(car(LIST))))

::sys <cadadr VAR LIST>

car(cdr(car(cdr(LIST))))

::sys <caddar VAR LIST>

car(cdr(cdr(car(LIST))))

::sys <cadddr VAR LIST>

car(cdr(cdr(cdr(LIST))))

::sys <cdaaar VAR LIST>

cdr(car(car(car(LIST))))

::sys <cdaadr VAR LIST>

cdr(car(car(cdr(LIST))))

::sys <cdadar VAR LIST>

cdr(car(cdr(car(LIST))))

::sys <cdaddr VAR LIST>

cdr(car(cdr(cdr(LIST))))

::sys <cddaar VAR LIST>

cdr(cdr(car(car(LIST))))

::sys <cddadr VAR LIST>

cdr(cdr(car(cdr(LIST))))

::sys <cdddar VAR LIST>

cdr(cdr(cdr(car(LIST))))

::sys <cddddr VAR LIST>

cdr(cdr(cdr(cdr(LIST))))

::sys <cons VAR LIST1 LIST2>

Connection of LIST

::sys <code CODE-NAME>

A setup of a character code.
UTF8, EUCJP, and SJIS can be specified.

::sys <char VAR STRINGS>

STRINGS is decomposed for every be character, and it
is made LIST, and is set as VAR. A multibyte character
like a Japanese character is also right, and it is every
character.

::sys <byte VAR STRINGS>

STRINGS is decomposed for every character code,
and it is made LIST, and is set as VAR.

::sys <asciichar VAR STRINGS>
::sys <utf8char VAR STRINGS>
::sys <eucchar VAR STRINGS>
::sys <sjischar VAR STRINGS>

STRINGS is decomposed for every character code,
and it is made LIST, and is set as VAR.

::sys <concat VAR LIST>

Strings LIST is made to unite, STRINGS is compounded and
it is set as VAR.

::sys <concatcode VAR LIST>

Character code LIST is made to unite, STRINGS is compounded and
it is set as VAR.

::sys <bitand VAR NUMBER1 NUMBER2>
::sys <bitor VAR NUMBER1 NUMBER2>
::sys <bitxor VAR NUMBER1 NUMBER2>
::sys <bitnot VAR NUMBER1>

bit operation

::sys <shiftl VAR NUMBER SHIFT>
::sys <shiftr VAR NUMBER SHIFT>

The bit shift of an integral value. shiftl is shifted
to the left and shifutr is shifted to the right.

::sys <eq ARG1 ARG2>
::sys <noteq ARG1 ARG2>
::sys <is ARG1 ARG2>

Comparison of ARG1 and ARG2

::sys <getc VAR>

One character is inputted into VAR.

::sys <putc CHAR>

One character is outputted.

::sys <getline VAR [PRED...]>

One line is inputted and it is set as VAR. When
PRED is set up, PRED is performed by considering
a tmp file as an input.

::sys <syntax STRINGS PRED...>

A predicate is performed by making a character string
into an input file.

::sys <tmpfile VAR>

Temporary FILE NAME is set as VAR

::sys <openr FILE-NAME PRED...>

The file of FILE-NAME is opened in postscript
reading, and PRED is performed.

::sys <openw FILE-NAME PRED...>

The file of FILE-NAME is opened in postscript
writing, and PRED is performed.

::sys <openwp FILE-NAME PRED...>

The file of FILE-NAME is opened in postscript
writing, and PRED is performed.

::sys <gettime VAR>

The present time is set as a variable by a micro second bit.

::sys <time VAR>

List of (the user time, sys time, and elapsed time)
of the predicate under execution is set as a variable.

::sys <date VAR>

Time is set as a variable.

::sys <sleep SEC>

Sleep is done during SEC second.

::sys <usleep SEC>

Sleep is done during micro SEC second.

::sys <pause>

It waits until Enter is pushed.

::sys <uname VAR>

The information on a system is set as VAR.

::sys <countnode VAR>

The number of nodes currently used is set as VAR.

::sys <gc>

A garbage collector is started.