●　sys モジュール述語

　sysモジュールはシステムに組み込まれた標準のライブラリ
モジュールの集合です。
呼び出すときには、″::sys″に続けて記述します。

::sys <append 変数 リスト1 リスト2>

　　リスト1とリスト2を連結して変数に設定します。
　　listモジュールライブラリの同名の述語の高速版です。

::sys <reverse #out #in>

　　リスト#inを逆の順に変換して#outに設定します。
　　listモジュールライブラリの同名の述語の高速版です。

::sys <member #mem #list>

　　#listの中に#memが含まれているか判定します。
　　listモジュールライブラリの同名の述語の高速版です。

::sys <arg 変数 複素数>

　　複素数の偏角を変数に設定します。

::sys <args 変数>

　　変数に、デカルト言語を起動したときの引数を設定
　　します。

::sys <avg 変数 リスト>
::sys <avgf 変数 リスト>

　　リストの要素の平均値を変数に設定する。
　　リストの要素は評価されてから、計算される。
　　avgは整数で計算する。
　　avgfは浮動小数点数で計算する。

::sys <basename 変数 パス>

　　パスからファイル名の部分を抜き出して変数に設定します。

::sys <checkObj 名前>

　　名前がオブジェクトか判定します。

::sys <chomp 変数 文字列>

文字列の後に改行が付いていたら削除する。

::sys <chop 変数 文字列>

文字列の最後の文字を削除する。

::sys <clear>

画面をクリアします。

::sys <conj 変数 複素数>

複素数の虚数部の符号を反転させて共役複素数を変数に設定します。

::sys <dirname 変数 パス>

パスからディレクトリ名の部分を抜き出して変数に設定します。

::sys <DLIBPATH 変数>
デカルト言語の使用するパスDLIBPATHを変数に表示します。

::sys <EqOR 比較値 値1 値2 ...>
比較値と比べて、その他の値1, 値2の一つでも一致した時にはtrueを返します。一つも一致しなければunknownを返します。

::sys <findlist 変数 キー リスト>

リストは、((キー1 値 ...)(キー2 値 ...) ...)のような形式です。
このリストの中から、指定されたキーと合致するデータを抽出します。
結果は、変数に設定されます。

::sys <flush>

IOキャッシュをフラッシュする。

::sys <gsub 正規表現パターン 文字列 置換文字列 出力文字列>

文字列に正規表現パターンを適用して該当した部分を、
置換文字列に置換えた結果を出力文字列に設定します。
置換えは文字列のすべての該当部分に行われます。
(Windows上では動作しません。)

::sys <insert 変数 文字列１ 位置 文字列2>

文字列1の位置に文字列2を挿入します。

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

引数を判定して該当すればtrueを返します。該当
しなければunknownを返します。

::sys <isObj 引数>

引数がオブジェクトならばtrueを返します。
そうでなければunknownを返します。

::sys <isTrue 述語>
::sys <isFalse 述語>
::sys <isUnknown 述語>

引数の述語の結果を判定し、該当すればtrueを返します。
該当しなければunknownを返します。

::sys <isRegistered VAL LIST>

VALがLISTの要素であれば、trueを返します。

そうでなければunknownを返します。

::sys <isUnregistered VAL LIST>

VALがLISTの要素でなければ、trueを返します。
そうでなければunknownを返します。

::sys <leftstr 変数 文字列 長さ>

文字列の左から長さの文字列を切り取り変数に設定します。

::sys <lefttrim 変数 文字列>

文字列の前の空白を外して変数に設定します。

::sys 〈max 変数 リスト〉

リストの中の要素の最大の整数値を変数に設定する。

::sys 〈maxf 変数 リスト〉

リストの中の要素の最大の浮動小数点値を変数に設定する。

::sys 〈min 変数 リスト〉

リストの中の要素の最小の整数値を変数に設定する。

::sys 〈minf 変数 リスト〉

リストの中の要素の最小の浮動小数点値を変数に設定する。

::sys 〈mkpred 引数〉

引数を述語に変換します。

::sys 〈nth 変数 リスト インデックス〉
リストのインデックス番目の値を変数に設定する。

::sys 〈setnth 変数 リスト インデックス 値〉
リストのインデックス番目に値を入れた
リストを変数に設定する。

::sys 〈real 変数 複素数〉

複素数の実部を変数に設定します。

::sys 〈image 変数 複素数〉

複素数の虚部を変数に設定します。

::sys 〈norm 変数 複素数〉

実数部と虚数部の平方の合計を変数に設定します。

::sys 〈polar 変数 絶対値 偏角〉

絶対値と偏角の極座標系による複素数を変数に設定します。

::sys 〈regex 正規表現パターン 文字列 前文字列 マッチ文字列 後文字列〉

文字列に正規表現パターンを適用した結果をマッチ文字列
に設定し、前後の文字列を前文字列と後文字列に設定します。
(Windows上では動作しません。)

::sys 〈rightstr 変数 文字列 長さ〉

文字列の右から長さの文字列を切り取り変数に設定します。

::sys <righttrim 変数 文字列>

文字列の後の空白を外して変数に設定します。

::sys <padding 変数 数 ITEM>

ITEMを数だけ含むリストを変数に設定する。

::sys <seq 変数 初期値 終了値>

変数に、初期値から終了値までの数のリストを設定します。
初期値が終了値より小さければ、昇順のリストを生成します。
初期値が終了値より大きければ、降順のリストを生成します。

::sys <sort 変数 リスト>
::sys <sortascend 変数 リスト>

リストを昇順でソートして、変数に設定する。

::sys <sortdescend 変数 リスト>

リストを降順でソートして、変数に設定する。

::sys <split 変数 文字列 [区切り文字]>

文字列を区切り文字で分けてリストにしたものを
変数に設定します。区切り文字が指定されていない
場合は、空白とタブで区切られます。

::sys <strbyte 変数 文字列>

文字列のbyte数を変数に設定する。

::sys <strdelcntl 変数 文字列>

文字列の中の制御文字を削除して、変数に設定する。

::sys <strfry 変数 文字列>

文字列の文字の順番をランダムに変更して、変数に設定する。

::sys <strlen 変数 文字列>

文字列の長さを変数に設定する。

::sys <strrepeat 変数 文字列 繰り返し数>

文字列を繰り返した値を変数に設定する。

::sys <strrotateleft 変数 文字列 [シフト数]>
::sys <strrotateright 変数 文字列 [シフト数]>

文字列をシフト数ずらした値を変数に設定する。
溢れた文字は反対側に移される。
シフト数が省略された場合は、１文字だけシフトする。

::sys <strshiftleft 変数 文字列 [シフト数]>
::sys <strshiftright 変数 文字列 [シフト数]>

文字列をシフト数ずらした値を変数に設定する。
シフト数が省略された場合は、１文字だけシフトする。

::sys <strsort 変数 文字列>
::sys <strsortreverse 変数 文字列>

文字列の中の文字をソートして変数に設定する。

::sys <sub 正規表現パターン 文字列 置換文字列 出力文字列>

文字列に正規表現パターンを適用して該当した部分を、
置換文字列に置換えた結果を出力文字列に設定します。
置換えは最初の１回だけ行われます。
(Windows上では動作しません。)

::sys <substr 変数 文字列 位置 長さ>

文字列の位置から長さの文字列を切り取り変数に設定します。

::sys <suffix 変数 パス サフィックス>

パスのサフィックスを引数のサフィックスに変換する。

::sys <sum 変数 リスト>
::sys <sumf 変数 リスト>

リストの要素を合計して変数に設定します。
リストの要素には、関数述語も記述できます。
sumは整数として合計します。
sumfは浮動小数点数として合計します。

::sys <switch 比較値 値1 述語1 値2 述語2 ...>
比較値と比べて、値と一致した時には対になる述語を実行します。
複数の述語を実行死体場合は、括弧()でくくります。

::sys <toupper 変数 文字列>

文字列を大文字にします。

::sys <tolower 変数 文字列>

文字列を小文字にします。

::sys <trim 変数 文字列>

文字列の前後の空白を外して変数に設定します。

::sys <writenl リスト>

リストを出力した後、改行します。

::sys <write リスト>
::sys <w リスト>

リストを出力します。

::sys <length 変数 リスト>

リストの長さを変数に設定します。

::sys <random 変数>

変数に乱数を設定します。

::sys <PI 変数>

円周率の値を変数に設定します。

::sys <sin 変数 ラジアン>
::sys <cos 変数 ラジアン>
::sys <tan 変数 ラジアン>

三角関数

::sys <asin 変数 値>
::sys <acos 変数 値>
::sys <atan 変数 値>
::sys <atan2 変数 値1 値2>

逆三角関数

::sys <sinh 変数 ラジアン>
::sys <cosh 変数 ラジアン>
::sys <tanh 変数 ラジアン>

ハイパボリック三角関数

::sys <asinh 変数 値>

::sys <acosh 変数 値>
::sys <atanh 変数 値>

ハイパボリック逆三角関数

::sys <e 変数>

自然対数の底eを変数に設定します。

::sys <log 変数 値>
::sys <log10 変数 値>
::sys <exp 変数 値>
::sys <exp2 変数 値>
::sys <exp10 変数 値>
::sys <pow 変数 値1 値2>

対数関数

::sys <sqrt 変数 値>

平方根

::sys <abs 変数 値>

絶対値

::sys <int 変数 値>

整数値

::sys <ceil 変数 値>

引き数より小さくない最小の整数値

::sys <floor 変数 値>

引き数を越えない最大の整数値

::sys <trunc 変数 値>

0 に近い方の整数値に丸める

::sys <car 変数 値>
::sys <cdr 変数 値>

リストのcar, cdr

::sys <caar 変数 値>

car(car(値))

::sys <cadr 変数 値>

car(cdr(値))

::sys <cdar 変数 値>

cdr(car(値))

::sys <cddr 変数 値>

cdr(cdr(値))

::sys <caaar 変数 値>

car(car(car(値)))

::sys <caadr 変数 値>

car(car(cdr(値)))

::sys <cadar 変数 値>

car(cdr(car(値)))

::sys <caddr 変数 値>

car(cdr(cdr(値)))

::sys <cdaar 変数 値>

cdr(car(car(値)))

::sys <cdadr 変数 値>

cdr(car(cdr(値)))

::sys <cddar 変数 値>

cdr(cdr(car(値)))

::sys <cdddr 変数 値>

cdr(cdr(cdr(値)))

::sys <caaaar 変数 値>

car(car(car(car(値))))

::sys <caaadr 変数 値>

car(car(car(cdr(値))))

::sys <caadar 変数 値>

car(car(cdr(car(値))))

::sys <caaddr 変数 値>

car(car(cdr(cdr(値))))

::sys <cadaar 変数 値>

car(cdr(car(car(値))))

::sys <cadadr 変数 値>

car(cdr(car(cdr(値))))

::sys <caddar 変数 値>

car(cdr(cdr(car(値))))

::sys <cadddr 変数 値>

car(cdr(cdr(cdr(値))))

::sys <cdaaar 変数 値>

cdr(car(car(car(値))))

::sys <cdaadr 変数 値>

cdr(car(car(cdr(値))))

::sys <cdadar 変数 値>

cdr(car(cdr(car(値))))

::sys <cdaddr 変数 値>

cdr(car(cdr(cdr(値))))

::sys <cddaar 変数 値>

cdr(cdr(car(car(値))))

::sys <cddadr 変数 値>

cdr(cdr(car(cdr(値))))

::sys <cdddar 変数 値>

cdr(cdr(cdr(car(値))))

::sys <cddddr 変数 値>

cdr(cdr(cdr(cdr(値))))

::sys <cons 変数 リスト1 リスト2>

リストの連結

::sys <code コード>

文字コードの設定。UTF8, EUCJP, SJISが指定できます。

::sys <char 変数 文字列>

文字列を文字ごとに分解してリストにして、変数に設定します。漢字のような多バイト文字も正しく一文字ごとに分解します。

::sys <byte 変数 文字列>

文字列をバイトごとに分解してリストにして変数に設定します。漢字のような多バイト文字もバイト単位に分解されます。

::sys <asciichar 変数 文字列>
::sys <utf8char 変数 文字列>
::sys <eucchar 変数 文字列>
::sys <sjischar 変数 文字列>

文字列を文字コードごとに分解してリストにして変数に設定します。

::sys <concat 変数 リスト>

文字のリストを合体させ文字列を合成して変数に設定します。

::sys <concatcode 変数 リスト>

文字コードのリストを合体させ文字列を合成して変数に設定します。

::sys <bitand 変数 数値1 数値2>
::sys <bitor 変数 数値1 数値2>
::sys <bitxor 変数 数値1 数値2>
::sys <bitnot 変数 数値1>

bit演算

::sys <shiftl 変数 数値 シフト数>
::sys <shiftr 変数 数値 シフト数>

整数値のビットシフト。
shiftlは左へ、shifutrは右へシフトさせます。

::sys <eq 引数1 引数2>
::sys <noteq 引数1 引数2>
::sys <is 引数1 引数2>

引数1と引数2の比較

::sys <getc 変数>

変数に１char入力します。

::sys <putc 文字>

1charを出力します。

::sys <getline 変数 [述語...]>

1行入力して変数に設定します。述語が設定されている場合は
変数に設定された文字列を入力として述語が実行されます。

::sys <syntax 文字列 述語...>

文字列を入力ファイルとして述語が実行されます。

::sys <tmpfile 変数>

テンポラリファイル名を変数に設定します。

::sys <openr ファイル名 述語...>

ファイル名のファイルを読み取り用にオープンして、
述語を実行します。

::sys <openw ファイル名 述語...>

ファイル名のファイルを書き込み用にオープンして、
述語を実行します。

::sys <openwp ファイル名 述語...>

ファイル名のファイルを追記書き込み用にオープン
して、述語を実行します。

::sys <gettime 変数>
現在の時刻をマイクロ秒単位で変数に設定します。

::sys <time 変数>
実行中の述語の(user時間, sys時間, elapsed時間)のリストを
変数に設定する。

::sys <date 変数>
日時を変数に設定します。

::sys <sleep 秒数>
秒数の間スリープします。

::sys <usleep マイクロ秒数>
マイクロ秒数の間スリープします。

::sys <pause>
Enterキーが押されるまで待ちます。

::sys <uname VAR>

システムの情報をVARに設定する。

::sys <countnode 変数>

使用しているノード数を変数に設定します。

::sys <PrintResultOn>

実行結果resultを表示する。

::sys <PrintResultOff>

実行結果resultを表示しない。

::sys <gc>

ガーベージコレクタを起動します。