functions.dkh

Biblioteca de funciones de propósito general para Devkron

Versión 1

Contenido

[Variables globales 3](#_Toc78197737)

[Constantes 3](#_Toc78197738)

[Plantillas de variables 3](#_Toc78197739)

[Variables de configuración 3](#_Toc78197740)

[Funciones 3](#_Toc78197741)

[Registros (estructuras) 3](#_Toc78197742)

[Listas 6](#_Toc78197743)

[Tablas 7](#_Toc78197744)

[Cadenas y textos 13](#_Toc78197745)

[Fecha y hora 18](#_Toc78197746)

[Codificación y encriptación 18](#_Toc78197747)

[Archivos y carpetas 20](#_Toc78197748)

[Utilidades 24](#_Toc78197749)

# Variables globales

Estas variables globales mejoran la seguridad de tipos y facilitan la lectura de los programas.

## Constantes

|  |  |
| --- | --- |
| @true | Use estas variables globales como constantes para comparaciones en expresiones como: valor==@true, valor==@false |
| @false |

## Plantillas de variables

|  |  |
| --- | --- |
| @bool | Use estas variables para establecer atributos al momento de declarar variables en la forma (ejemplos):  mivariable[@bool] // Declara una variable que solo admite 1 o 0  minumero [@int] =10 /\* Declara una variable que solo admite números enteros y la inicializa con el valor 10\*/ |
| @int |
| @number |
| @positive |
| @negative |
| @string |

Vea cómo se han declarado estas variables en el archivo functions.dkh

## Variables de configuración

|  |  |
| --- | --- |
| @exec\_timeout | Contiene el tiempo en milisegundos que espera el intérprete por la respuesta de un proceso invocado con la función exec definida más adelante. |

# Funciones

## Registros (estructuras)

|  |  |
| --- | --- |
| field.remove  Elimina el campo indicado del registro y devuelve @true si tuvo éxito.  field.remove(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a eliminar, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.exist  Devuelve @true si el campo indicado existe en el registro.  field.exist(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a buscar, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.typeof  Devuelve un valor numérico que indica el tipo del campo especificado. Los posibles valores de retorno son:  1 – El campo es de tipo cadena,  2- El campo es de tipo numérico,  3- El campo es una referencia o  0 – El campo no existe.  field.typeof(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a buscar, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.num  Devuelve el valor numérico de un campo, si no existe se produce un error, si el campo no es de tipo numérico se intenta realizar la conversión automáticamente.  field.num(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.str  Devuelve el valor de cadena de un campo, si no existe se produce un error, si el campo no es de tipo cadena se intenta realizar la conversión automáticamente.  field.str(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.obj  Devuelve una referencia al valor del campo indicado, si no existe se produce un error.  field.obj(registro,campo) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros. |
| field.dnum  Devuelve el valor numérico de un campo, si el campo no es de tipo numérico se intenta realizar la conversión automáticamente, si no existe devuelve el valor predeterminado por defaul.  field.dnum(registro,campo,default) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros.  default – Número que será devuelto si el campo no existe. |
| field.dstr  Devuelve el valor de cadena de un campo, si el campo no es de tipo cadena se intenta realizar la conversión automáticamente, si no existe devuelve el valor predeterminado por defaul.  field.dstr(registro,campo,default) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros.  default – Cadena que será devuelta si el campo no existe. |
| field.dobj  Devuelve una referencia al valor de un campo, si no existe el campo devuelve la referencia predeterminada por defaul.  field.dstr(registro,campo,default) | registro – Referencia a un registro (estructura)  campo – Cadena que identifica el nombre del campo a devolver, se admite la notación de rutas “campo/campo/campo” para acceder a miembros dentro de otros miembros.  default – Referencia que será devuelta si el campo no existe. |
| record.create  Devuelve una referencia a un nuevo registro.  record.create() | Sin argumentos |
| record.clear  Elimina todos los campos de una estructura.  record.clear(registro) | registro – Referencia a un registro (estructura) |
| record.copy  Devuelve un nuevo registro con una copia de los campos indicados.  record.copy(origen,campos) | origen – Referencia a un registro  campos – Cadena con la lista de nombres de campos a copiar al nuevo registro delimitados por comas, si se establece únicamente el carácter \* se copian todos los campos. |
| record.copyex  Devuelve un nuevo registro con una copia de los campos indicados renombrados.  record.copyex(origen, campos, delimitador, alias, delimitadoralias) | origen – Referencia a un registro  campos – Cadena con la lista de nombres de campos a copiar al nuevo registro delimitados por el carácter indicado por delimitador.  delimitador – Cadena con el carácter que usa el parámetro anterior para delimitar los campos a copiar  alias – Cadena con la lista de nombres de campos que tendrán los campos enumerados por campos en el registro copiado.  delimitadoralias - Cadena con el carácter que usa el parámetro anterior para delimitar los alias  La lista de campos de campos y alias deben tener el mismo número de elementos. |
| record.caseinsensitive  Devuelve una referencia a una copia del registro pero con insensibilidad a mayúsculas o minúsculas para los nombres de campos.  record.caseinsensitive(registro) | registro – Referencia a un registro  En DKL los registros son de manera predeterminada insensibles a mayúsculas/minúsculas en sus campos, esta función no es usada generalmente. |

## Listas

Las listas no son una estructura nativa del lenguaje Devkron, por lo que se implementan a través de objetos que implementan System.Collections.Generic.IList con estas funciones.

Las listas tienen índice base cero.

|  |  |
| --- | --- |
| list.create  Devuelve una referencia a una nueva instancia de un objeto de lista.  list.create () | Si argumentos |
| list.create\_from\_fields  Devuelve una referencia a una nueva lista formada por los nombres de todos los campos de un registro.  list.create\_from\_fields(registro) | registro – Referencia a un registro (estructura) |
| list.count  Devuelve la cantidad de elementos de una lista.  list.count(lista) | lista – Referencia a una lista |
| list.add  Agrega un elemento a la lista  list.add(lista, elemento) | lista – Referencia a una lista  elemento – Cadena, número o referencia a agregar a la lista |
| list.ins  Inserta un elemento en la posición especificada de la lista  list.ins(lista, posición, elemento) | lista – Referencia a una lista  posición – Número que indica la posición en la que se va a insertar el elemento  elemento – Cadena, número o referencia a insertar a la lista |
| list.set  Establece un elemento en la posición indicada  list.set(lista, posición, elemento) | lista – Referencia a una lista  posición – Número que indica la posición en la que se va a reemplazar el elemento  elemento – Cadena, número o referencia a colocar en la lista |
| list.obj  Devuelve una referencia al valor en la posición indicada de la lista  list.obj(lista, posición) | lista – Referencia a una lista  posición – Índice base cero del elemento cuya referencia va a devolverse |
| list.str  Devuelve como cadena el valor de la posición indicada de la lista  list.str(lista, posición) | lista – Referencia a una lista  posición – Índice base cero del elemento que se va a devolver como cadena |
| list.num  Devuelve como número el valor de la posición indicada de la lista  list.num(lista.posición) | lista – Referencia a una lista  posición – Índice base cero del elemento que se va a devolver como número |
| list.del  Elimina el elemento de la lista en la posición indicada  list.del(lista, posición) | lista – Referencia a una lista  posición – Índice base cero del elemento que se va a eliminar |
| list.clear  Elimina todos los elementos de la lista  list.clear(lista) | lista – Referencia a una lista |

## Tablas

Las tablas (en memoria) se implementan a través de las clases de System.Data.DataTables con estas funciones para crear y manipular estructura y datos.

|  |  |
| --- | --- |
| table.rows  Devuelve la cantidad de filas de una tabla  table.rows(tabla) | tabla – Referencia a una tabla |
| table.cols  Devuelve la cantidad de columnas de una tabla  table.cols(tabla) | tabla – Referencia a una tabla |
| cell.num  Devuelve el valor numérico de una celda ubicada en la posición de fila y columna indicada  Si la celda es DBNull se produce una excepción  cell.num(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla |
| cell.numc  Devuelve el valor numérico de una celda ubicada en la posición de fila y columna indicada por nombre  Si la celda es DBNull se produce una excepción  cell.numc(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla |
| cell.dnum  Devuelve el valor numérico de una celda ubicada en la posición de fila y columna indicada  Si la celda es DBNull devuelve el valor indicado como predeterminado  cell.dnum(tabla, fila, columna, default) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla  default – Valor que será devuelto si la celda indicada es DBNull |
| cell.dnumc  Devuelve el valor numérico de una celda ubicada en la posición de fila y columna indicada por nombre  Si la celda es DBNull devuelve el valor indicado como predeterminado  cell.dnumc(tabla, fila, columna, default) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla  default – Valor que será devuelto si la celda indicada es DBNull |
| cell.str  Devuelve el valor de cadena de una celda ubicada en la posición de fila y columna indicada  Si la celda es DBNull se produce una excepción  cell.str(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla |
| cell.strc  Devuelve el valor de cadena de una celda ubicada en la posición de fila y columna indicada por nombre  Si la celda es DBNull se produce una excepción  cell.strc(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla |
| cell.dstr  Devuelve el valor de cadena de una celda ubicada en la posición de fila y columna indicada  Si la celda es DBNull devuelve el valor indicado como predeterminado  cell.dstr(tabla, fila, columna,default) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla  default – Valor que será devuelto si la celda indicada es DBNull |
| cell.dstrc  Devuelve el valor de cadena de una celda ubicada en la posición de fila y columna indicada por nombre  Si la celda es DBNull devuelve el valor indicado como predeterminado  cell.dstrc(tabla, fila, columna,default) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla  default – Valor que será devuelto si la celda indicada es DBNull |
| cell.obj  Devuelve una referencia al valor de una celda ubicada en la posición de fila y columna indicada  cell.obj(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla |
| cell.objc  Devuelve una referencia al valor de una celda ubicada en la posición de fila y columna indicada por nombre.  cell.objc(tabla, fila, columna) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla |
| col.caption  Establece el título de la columna indicada por su índice  col.caption(tabla, columna, titulo) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla  titulo – Cadena con el título que se asignará a la columna |
| col.captionc  Establece el título de la columna indicada por su nombre  col.captionc(tabla, columna, titulo) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla  titulo – Cadena con el título que se asignará a la columna |
| col.get  Devuelve una referencia al objeto subyacente a la columna indicada por su índice  col.get(tabla, columna) | tabla – Referencia a una tabla  columna – Número que representa el índice base cero de una columna de la tabla |
| col.getc  Devuelve una referencia al objeto subyacente a la columna indicada por su nombre  col.get(tabla, columna) | tabla – Referencia a una tabla  columna – Cadena con el nombre de una columna de la tabla |
| row.get  Devuelve una referencia a un registro que contiene todos los datos de la fila indicada.  row.get(tabla, fila) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla |
| cell.set  Establece el valor de una celda de la tabla por el índice de su fila y columna  cell.set(tabla, fila, columna, valor) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Número que representa el índice base cero de una columna de la tabla  valor – Cadena, número o referencia que será asignada a la celda indicada |
| cell.setc  Establece el valor de una celda de la tabla en la fila y columna indicada por nombre  cell.setc(tabla, fila, columna, valor) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  columna – Cadena con el nombre de una columna de la tabla  valor – Cadena, número o referencia que será asignada a la celda indicada |
| table.create  Devuelve una referencia hacia una nueva tabla con las columnas correspondientes a los campos del registro indicado  table.create(registro) | registro – Referencia a un registro cuyos nombres y tipo de campos serán usados para definir las columnas de una tabla nueva |
| table.new  Devuelve una referencia hacia una tabla vacía | Sin argumentos |
| row.del  Elimina la fila indicada  row.del(tabla,fila) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla |
| row.add  Agrega los datos del registro indicado como una nueva fila al final de la tabla, los nombres de los campos del registro deben coincidir con los nombres de las columnas.  row.add(tabla, registro) | tabla – Referencia a una tabla  registro – Referencia a un registro cuyos datos de campos se agregarán a la tabla como una nueva fila |
| row.ins  Inserta los datos del registro indicado como una nueva fila en la posición señalada de la tabla, los nombres de los campos del registro deben coincidir con los nombres de las columnas.  row.ins(tabla,fila, registro) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  registro – Referencia a un registro cuyos datos de campos se insertarán a la tabla como una nueva fila |
| row.upd  Actualiza los datos del registro indicado en la fila señalada de la tabla, los nombres de los campos del registro deben coincidir con los nombres de las columnas.  row.upd(tabla, fila, registro) | tabla – Referencia a una tabla  fila – Número que representa el índice base cero de una fila en la tabla  registro – Referencia a un registro cuyos datos de campos actualizarán los de la tabla en la fila indicada |
| table.pivot  Devuelve una referencia a una nueva tabla que contiene la representación transversal de otra.  table.pivot(tabla, x, y, z, valor\_nulo, sumar) | tabla – Referencia a una tabla  x,y,z – Cadenas con los nombres de campos que serán usados como columnas de consolidación al “girar” la tabla, se requiere indicar al menos x, las demás pueden ser cadena vacía (“”)  valor\_nulo – Cadena con el valor que será colocado en las celdas nulas, puede indicarse como cadena vacía(“”)  sumar – Numérico (booleano) si es @true se incluirá una fila adicional con las sumatorias de las columnas. |
| table.from  Deserializa una tabla a partir de una cadena y devuelve una referencia  table.from(cadena) | cadena – Cadena con código DKL que representa la tabla (generalmente obtenida con la función table.to) |
| table.to  Serializa una tabla y devuelve una cadena que la representa.  table.to(tabla) | tabla – Referencia a una tabla de la que se obtendrá su representación como cadena (código DKL) |

## Cadenas y textos

|  |  |
| --- | --- |
| trim  Devuelve una cadena sin los caracteres vacíos (espacio y tabulador) del inicio y final de la cadena dada  trim(cad) | cad – Cadena de texto |
| ltrim  Devuelve una cadena sin los caracteres vacíos (espacio y tabulador) del inicio de la cadena dada  ltrim(cad) | cad – Cadena de texto |
| rtrim  Devuelve una cadena sin los caracteres vacíos (espacio y tabulador) del final de la cadena dada  rtrim(cad) | cad – Cadena de texto |
| left  Devuelve una subcadena que omite los caracteres al inicio de la cadena original indicados.  left(cad,omitir) | cad – Cadena de texto  omitir – Número de caracteres |
| right  Devuelve una subcadena que inicia en la cantidad de caracteres indicados a partir del final de la cadena original.  right(cad, antesfinal) | cad – Cadena de texto  antesfinal – Número de caracteres |
| substring  Devuelve una subcadena que inicia en el carácter indicado y tiene la longitud indicada.  substring(cad, inicio, longitud) | cad – Cadena de texto  inicio – Número (índice base cero)  longitud – Número (cantidad de caracteres) |
| lremove  Devuelve una subcadena sin los caracteres después de la posición indicada a partir del inicio de la cadena dada  lremove(cad, cant) | cad – Cadena de texto  cant – Número (cantidad) de caracteres |
| remove  Devuelve una subcadena sin los caracteres indicados del inicio y longitud en la cadena dada  remove(cad, inicio, longitud) | cad – Cadena de texto  inicio – Número (índice base cero)  longitud – Número (cantidad de caracteres) |
| rremove  Devuelve una subcadena sin los caracteres indicados contados a partir del final de la cadena dada  rremove(cad, cant) | cad – Cadena de texto  cant – Número (cantidad de caracteres) |
| length  Devuelve la longitud de una cadena  length(cad) | cad – Cadena de texto |
| toupper  Devuelve la cadena dada en mayúsculas  toupper(cad) | cad – Cadena de texto |
| tolower  Devuelve la cadena dada en minúsculas  tolower(cad) | cad – Cadena de texto |
| split  Devuelve una referencia a una lista de subcadenas delimitadas por el carácter indicado en la cadena dada.  split(cad, carácter) | cad – Cadena de texto  carácter – Cadena con el carácter separador |
| replace  Devuelve una cadena en donde se ha reemplazado una subcadena por otra en una cadena dada  replace(cad, buscar, reemplazo) | cad – Cadena de texto  buscar- Cadena (subcadena) a buscar  reemplazo – Cadena (subcadena) a usar como reemplazo de la subcadena buscar |
| padright  Devuelve una cadena de la longitud indicada que incluye a la cadena dada alineada a la derecha con las posiciones vacías llenas con el carácter indicado.  padright(cad, tam, car) | cad – Cadena de texto  tam – Tamaño al que se ajustará la cadena resultante  car – Carácter con el que se rellenarán los espacios vacíos |
| padleft  Devuelve una cadena de la longitud indicada que incluye a la cadena dada alineada a la izquierda con las posiciones vacías llenas con el carácter indicado.  padleft(cad, tam, car) | cad – Cadena de texto  tam – Tamaño al que se ajustará la cadena resultante  car – Carácter con el que se rellenarán los espacios vacíos |
| contains  Devuelve cierto si una cadena contiene a la subcadena indicada  contains(cad, subcad) | cad – Cadena de texto  subcad – Cadena (subcadena) a buscar |
| findstrex  Devuelve la posición de inicio (base cero) de la subcadena buscada en la cadena dada. Si la subcadena no forma parte de la cadena retorna -1.  findstrex(cadena, subcadena, sensible) | cad – Cadena de texto  subcadena – Cadena (subcadena) a buscar  sensible – Número (booleano) que si es @false la búsqueda no es sensible a mayúsculas/minúsculas |
| findstr  Devuelve la posición de inicio (base cero) de la subcadena buscada en la cadena dada sin distinción de mayúsculas/minúsculas. Si la subcadena no forma parte de la cadena retorna -1.  findstr(cadena, subcadena) | cad – Cadena de texto  subcadena – Cadena (subcadena) a buscar |
| ranychars  Devuelve una cadena donde se han reemplazado cualquiera de los caracteres indicados por el carácter señalado.  ranychasr(cadena, caracteres, car\_reemplazo) | cad – Cadena de texto  caracteres – Cadena que contiene los caracteres a reemplazar  car\_reemplazo – Cadena (un único carácter) que será usado para reemplazar cualquiera de los de la cadena ‘caracteres’ |
| ranydiffchars  Devuelve una cadena donde se han reemplazado cualquiera de los caracteres diferentes a los indicados por el carácter señalado.  ranydiffchars (cadena, caracteres, car\_reemplazo) | cad – Cadena de texto  caracteres – Cadena que contiene los caracteres que no serán reemplazados  car\_reemplazo – Cadena (un único carácter) que será usado para reemplazar cualquiera de los de la cadena ‘caracteres’ |
| cut  Devuelve una cadena recortada a la cantidad de caracteres especificada, si la cadena dada es de longitud menor a lo solicitado la devuelve completa.  cut(cad, longitud) | cad – Cadena de texto  longitud – Número, longitud máxima a la que se truncará la cadena |
| format  Devuelve la representación de un número con el formato indicado.  format(num, formato) | num – Número  formato – Cadena que describe el formato para la representación (los formatos de C#) |
| isany  Devuelve @true si el carácter indicado es alguno de los que forman la cadena dada  isany(carácter, cadena) | carácter – Cadena (un único carácter) que será evaluado  cadena – Cadena de caracteres |
| containschar  Devuelve @true si alguno de los caracteres de la cadena de caracteres está en la cadena dada  containschar(cadena, cadena\_caracteres) | cadena – Cadena cuyos caracteres se evaluaran  cadena\_caracteres – Cadena con la lista de caracteres para verificar coincidencia |
| containsword  Devuelve @true si alguna de las palabras delimitadas por comas de la cadena de palabras está en la cadena dada.  containsword(cadena, lista\_palabras) | cadena – Cadena de texto  lista\_palabras – Cadena con las palabras a buscar delimitadas por comas |
| isanyword  Devuelve @true si alguna de las palabras delimitadas por comas de la cadena de palabras es la cadena  isanyword(cadena, lista\_palabras) | cadena – Cadena con una palabra  lista\_palabras – Lista de palabras separadas por comas contra la que se evaluará la cadena |
| from.args  Devuelve un registro con los pares clave valor delimitados por el carácter separado indicado de la lista de argumentos dada.  from.args(lista, separador) | lista – Referencia a una lista de cadenas  reparador – Cadena (un único carácter) que se usa para delimitar los pares clave/valor de cada elemento de la lista de cadenas |
| from.argsn  Devuelve un registro con los pares clave valor delimitados por el carácter separado indicado de la lista de argumentos dada. Además, aquellos listados como numéricos se asignan como tales al registro de resultados  from.args(lista, separador,numéricos) | lista – Referencia a una lista de cadenas  reparador – Cadena (un único carácter) que se usa para delimitar los pares clave/valor de cada elemento de la lista de cadenas  numéricos – Cadena de claves separadas por comas que indica cuáles claves deben reconocerse como números |
| text.create  Devuelve una referencia a un nuevo objeto de buffer de texto (basado en StringBuilder)  text.create() | Sin argumentos |
| text.echo  Agrega una cadena a un buffer de texto  text.echo(buffer, cadena) | buffer – Referencia a un buffer de texto  cadena – Cadena de texto |
| text.say  Agrega una línea de texto (adiciona la secuencia de retorno de carro y avance de línea) a un buffer de texto  text.say(buffer, cadena) | buffer – Referencia a un buffer de texto  cadena – Cadena de texto |
| text.clear  Limpia un buffer de texto  text.clear(buffer) | buffer – Referencia a un buffer de texto |
| text.get  Devuelve una cadena a partir de un buffer de texto  text.get(buffer) | buffer – Referencia a un buffer de texto |

## Fecha y hora

|  |  |
| --- | --- |
| now  Devuelve una referencia a una estructura DateTime de .Net que corresponde a la fecha y hora actuales  now() | Sin argumentos |
| str2dt  Devuelve una referencia a una estructura DateTime de .Net que corresponde a la cadena indicada si es posible hacer la conversión.  str2dt(cadena) | cadena – Cadena que representa una fecha-hora |
| date\_str  Devuelve una cadena que corresponde con la fecha (DateTime) indicada con el formato especificado.  date\_str(fechahora, formato) | fechahora – Referencia a una estructura de fecha-hora (DateTime)  formato – Cadena con el formato (C#) con el que se representará la fecha |

## Codificación y encriptación

|  |  |
| --- | --- |
| url\_encode  Devuelve una cadena codificada que puede usarse como parte de una url  url\_encode(cad) | cad – Cadena a codificar |
| url\_dcode  Devuelve una cadena decodificada que previamente fue codificada por url\_encode  url\_dcode(cad) | cad – Cadena a decodificar |
| to.base64  Devuelve la cadena codificada en base64 utf8  to.base64(cadena) | cadena – Cadena a codificar en base64 |
| from.base64  Devuelve la cadena decodificada de base64 utf8  from.base64(cadena) | cadena – Cadena a decodificar de base64 |
| to.b64  Devuelve la cadena codificada en base64 con la codificación de caracteres indicada  to.b64(cadena,codificación) | cadena – Cadena a codificar en base64  codificación – Cadena (constante) que indica la codificación a usar: utf8,utf7, utf32, Unicode o ascii |
| from.b64  Devuelve la cadena decodificada de base64 con la codificación de caracteres indicada  from.b64(cadena, codificación) | cadena – Cadena a decodificar de base64  codificación – Cadena (constante) que indica la codificación a usar: utf8,utf7, utf32, Unicode o ascii |
| encoding  Devuelve una referencia a un objeto Encoding de .Net con base en la cadena dada  encoding(codificación) | codificación – Cadena (constante) que indica la codificación a usar: utf8,utf7, utf32, Unicode o ascii |
| encryp  Devuelve una cadena encriptada con la clave dada  encryp(cadena, clave) | cadena – Cadena a encriptar  clave – Cadena que contiene la clave de encriptación |
| decryp  Devuelve una cadena desencriptada con la clave dada  decryp(cadena,clave) | cadena – Cadena a desencriptar  clave – Cadena que contiene la clave de encriptación |
| md5  Devuelve la firma md5 de la cadena  md5(cadena) | cadena –Cadena de texto |
| uuid  Devuelve una cadena que representa un Identificado Global Único de 128bits.  uuid() | Sin argumentos |
| hmacsha256  Devuelve una cadena que representa el HMAC de un mensaje con cifrado SHA 256  hmacsha256(mensaje, clave) | mensaje – Cadena de texto  clave – Clave de cifrado |

## Archivos y carpetas

|  |  |
| --- | --- |
| commandline  Devuelve una referencia a un registro que contiene los pares clave/valor pasados por línea de comando  commandline(sep) | sep – Cadena (un único carácter) que indica el separador clave/valor |
| commandlinen  Devuelve una referencia a un registro que contiene los pares clave/valor pasados por línea de comando y además identifica aquellos que deben considerarse de tipo numérico  commandlinen(sep, numéricos) | sep – Cadena (un único carácter) que indica el separador clave/valor  numéricos – Cadena que identifica las claves delimitadas por comas que deben ser tratadas como números |
| exec  Ejecuta un proceso (programa ejecutable) del sistema y devuelve su salida de consola como una cadena, desencadena una excepción si ocurre el ejecutable termina con un error.  exec(exe,args,wait) | exe – Cadena con el nombre y ruta del ejecutable  args – Cadena de argumentos de línea de comando  wait – Numérico (booleano), si es @true detendrá la ejecución del programa DKL y esperará hasta por @exec\_timeout milisegundos a que termine la ejecución del ejecutable. |
| \_exec  Ejecuta un proceso (programa ejecutable) del sistema y devuelve su salida de consola como una cadena.  \_exec(exe,args,wait, ml, riseerror) | exe – Cadena con el nombre y ruta del ejecutable  args – Cadena de argumentos de línea de comando  wait – Numérico (booleano), si es @true detendrá la ejecución del programa DKL y esperará hasta por ml milisegundos a que termine la ejecución del ejecutable.  ml – Numérico, cantidad de milisegundos a esperar  riseerror – Numérico (booleano) si es @false no se desencadena una excepción aunque el ejecutable termine por error |
| file.newer  Devuelve un número que indica cuál de dos archivos es más nuevo (con base en su fecha de modificación). Retorna -1 si f1 no existe, -2 si f2 no existe, 1 si f1 es más reciente que f2, 2 si f2 es más reciente que f1 y 0 si ambos tienen la misma fecha y hora de modificación.  file.newer(f1,f2) | f1 – Cadena con la ruta y nombre de un archivo  f2 – Cadena con la ruta y nombre de un archivo |
| file.exists  Deuelve @true si el archivo indicado existe  file(archivo) | archivo – Cadena con la ruta y nombre de un archivo |
| dir.exists  Devuelve @true si el directorio indicado existe  dir.exists(ruta) | ruta – Cadena con la ruta de un directorio |
| dir.delete  Elimina el directorio indicador  dir.delete(directorio) | directorio – Cadena con el nombre y ruta de un directorio |
| dir.create  Crea el directorio indicado  dir.create(directorio) | directorio – Cadena con el nombre y ruta de un directorio |
| file.rbytes  Devuelve una referencia a un array de Bytes que es el contenido del archivo indicado.  file.rbytes(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| file.wbytes  Escribe un array de bytes en el archivo indicado sobrescribiendo su contenido (si existe y si no lo crea).  file.wbytes(archivo, bytes) | archivo – Cadena con el nombre y ruta de un archivo  bytes – Referencia a un array de bytes |
| file.delete  Elimina el archivo indicado  file.delete(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| file.\_copy  Copia el archivo indicado, si el archivo de destino existe y se indica, se sobrescribe  file.\_copy(origen, destino, sobrescribir) | origen – Cadena con el nombre y ruta de un archivo  destino – Cadena con el nombre y ruta de un archivo  sobrescribir – Número (booleano) que indica si es @true, en caso de existir el archivo de destino será sobrescrito |
| file.extension  Devuelve una cadena con la extensión del archivo indicado  file.extension(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| file.name  Devuelve el nombre con extensión del archivo indicado  file.name(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| dir.name  Devuelve el nombre del directorio indicado  dir.name(directorio) | directorio – Cadena con el nombre y ruta de un directorio |
| file.justname  Devuelve solo el nombre (sin extensión) del archivo indicado  file.justname(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| list.dirs  Devuelve una referencia a una lista de cadenas que contiene todos los nombres de directorios de la ubicación indicada.  list.dirs(ruta) | ruta – Cadena con el nombre y ruta de un directorio |
| list.files  Devuelve una referencia a una lista de cadenas que contiene todos los nombres de archivos de la ubicación indicada.  list.files(ruta) | ruta – Cadena con el nombre y ruta de un directorio |
| dir.ensure  Verifica si un directorio existe y si no lo crea.  dir.ensure(directorio) | ruta – Cadena con el nombre y ruta de un directorio |
| file.copy  Copia el archivo indicado y sobrescribe el destino en caso de que exista.  file.copy(origen, destino) | origen – Cadena con el nombre y ruta de un archivo  destino – Cadena con el nombre y ruta de un archivo |
| file.wtext  Escribe la cadena de texto indicada en el archivo especificado con codificación utf8 sin BOM, si el archivo ya existe lo sobrescribe.  file.wtext(archivo, texto) | archivo – Cadena con el nombre y ruta de un archivo  texto – Cadena de texto |
| file.wtexte  Escribe la cadena de texto indicada en el archivo especificado con la codificación especificada, si el archivo ya existe lo sobrescribe.  file.wtexte(archivo, texto, codificación) | archivo – Cadena con el nombre y ruta de un archivo  texto – Cadena de texto  codificación – Cadena de texto que indica la codificación |
| file.rtext  Devuelve una cadena con el contenido del archivo especificado asumiendo que está codificado utf8  file.rtext(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| file.rtexte  Devuelve una cadena con el contenido del archivo especificado con la codificación especificada  file.rtexte(archivo, codificación) | archivo – Cadena con el nombre y ruta de un archivo  codificación – Cadena de texto que indica la codificación |
| file.len  Devuelve un número que representa el tamaño en bytes del archivo indicado  file.len(archivo) | archivo – Cadena con el nombre y ruta de un archivo |
| path.concat  Devuelve una cadena de ruta que es la concatenación de las dos rutas dadas.  path.concat(r1,r2) | r1 – Cadena con una ruta  r2- Cadena con otra parte de la ruta |
| dir.struct  Devuelve una referencia a una lista de cadenas que representa todas las carpetas de la estructura de directorios de la ruta dada  dir.struct(ruta) | ruta – Cadena con el nombre y ruta de un archivo o directorio |

## Utilidades

|  |  |
| --- | --- |
| parse.bool  Devuelve un valor numérico (booleado) tras el análisis de una cadena dada  parse.bool(texto) | texto- Cadena que puede evaluarse como booleano |
| parse.num  Devuelve un valor numérico tras el análisis de una cadena dada  parse.num(texto) | texto – Cadena que puede evaluarse como número |
| field.parse.dbool  Devuelve un valor numérico (booleado) tras el análisis de un campo de un registro  field.parse.dbool(registro, campo, default) | registro – Referencia a un registro  campo – Cadena con el nombre o ruta de un campo  default – Valor numérico a usar como predeterminado en caso de que el campo no existe |
| field.parse.dnum  Devuelve un valor numérico tras el análisis de un campo de un registro  field.parse.dnum(registro, campo, default) | registro – Referencia a un registro  campo – Cadena con el nombre o ruta de un campo  default – Valor numérico a usar como predeterminado en caso de que el campo no existe |
| foreach.fstr  Itera los campos de un registro y permite usar sus valores como cadenas con el apoyo de la sentencia go | Variables expuestas: index, key y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |
| foreach.fnum  Itera los campos de un registro y permite usar sus valores como números con el apoyo de la sentencia go | Variables expuestas: index, key y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |
| foreach.fobj  Itera los campos de un registro y permite usar sus valores como referencias a objetos con el apoyo de la sentencia go | Variables expuestas: index, key y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |
| foreach.lstr  Itera una lista y permite usar sus valores como cadenas con el apoyo de la sentencia go | Variables expuestas: index y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |
| foreach.lnum  Itera una lista y permite usar sus números como cadenas con el apoyo de la sentencia go | Variables expuestas: index y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |
| foreach.lobj  Itera una lista y permite usar sus valores como referencias a objetos con el apoyo de la sentencia go | Variables expuestas: index y value  Vea el código en functions.dkh y el ejemplo al final de esta tabla |

#include "dkli.dkh"

#!

program "ejemplo"

{

    #include "functions.dkh"

    new registro

    {

        @"nombre":"Juan"

        @"apellido1":"López"

        @"apellido2":"García"

        @"Identidad de género":"Hombre"

    }

    go foreach.fstr(index:i,key:k, value:v) with(registro)

    {

        do print("Índice:"+str(i)+" clave:"+k + " valor:"+v)

    }

    ref lista=list.create()

    do list.add(lista,"Uno")

    do list.add(lista,"Dos")

    do list.add(lista,"Tres")

    go foreach.lstr(index:i, value:v) with(lista)

    {

        do print("Índice:"+str(i)+ " valor:"+v)

    }

}