

RAT()

Retorna a posição da última ocorrência de um substring

Sintaxe

RAT(<cPesquisa>, <cDestino>) --> nPosicao

Argumentos

<cPesquisa> é a cadeia de caracteres a ser localizada.

<cDestino> é a cadeia de caracteres a ser procurada.

Retorno

RAT() retorna a posição de <cPesquisa> dentro de <cDestino> na forma de um valor numérico inteiro. Se <cPesquisa> não for encontrada, RAT() retorna zero.

Descrição

RAT() é uma função de tratamento de caracteres que retorna a posição da última ocorrência de um substring em uma cadeia de caracteres. Isto é feito pesquisando-se a cadeia de caracteres destino a partir da direita. RAT() é semelhante à função AT(), a qual retorna a posição da primeira ocorrência de um substring em uma cadeia de caracteres. RAT() também é semelhante ao operador \$, que determina se um substring está contido em uma cadeia de caracteres.

As funções RAT() e AT() são usadas juntamente com SUBSTR(), LEFT(), e RIGHT() para extrair substrings.

Exemplos

Este exemplo utiliza RAT() para criar uma função definida pelo usuário, FilePath(), que extrai o path de uma especificação de arquivo. Caso não haja especificação de path, FilePath() retorna uma cadeia de caracteres nula (""):

```
? FilePath("C:\DBF\Sales.dbf")           // Resulta: C:\DBF\
```

```
FUNCTION FilePath( cFile )
  LOCAL nPos, cFilePath
  IF (nPos := RAT("\", cFile)) != 0
    cFilePath = SUBSTR(cFile, 1, nPos)
  ELSE
    cFilePath = ""
  ENDIF
  RETURN cFilePath
```

Arquivos: Biblioteca EXTEND.LIB.

LEFT()

Extrai um substring que começa com o primeiro caractere em um string

Sintaxe

LEFT(<cString>, <nCont>) --> cSubString

Argumentos

<cString> é uma cadeia de caracteres da qual serao extraídos caracteres. O tamanho máximo de <cString> é 65.535 (64K) bytes.

<nCont> é a quantidade de caracteres a serem extraídos.

Retorno

LEFT() retorna os caracteres <nCont> mais à esquerda de <cString> na forma de uma cadeia de caracteres. Caso <nCont> seja negativo ou zero, LEFT() retorna uma cadeia de caracteres nula (""). Se <nCont> for maior do que o tamanho da cadeia de caracteres, LEFT() retorna toda a cadeia de caracteres.

Descrição

LEFT() é uma função de tratamento de caracteres que retorna um substring de uma cadeia de caracteres especificada. É o mesmo que SUBSTR(<cString>, 1, <nCont>). LEFT() também é semelhante a RIGHT(), função que retorna um substring que começa com o último caractere em um string.

LEFT(), RIGHT(), e SUBSTR() são frequentemente utilizadas juntamente com ambas as funções AT() e RAT() para localizar a primeira e/ou a última posição de um substring antes que o mesmo seja extraído.

Exemplos

- _ Este exemplo extrai os três primeiros caracteres da esquerda de um string:

```
? LEFT("ABCDEF", 3)           // Resulta: ABC
```

- _ Este exemplo extrai um substring a partir do início de um string até a primeira ocorrência de uma vírgula:

```
LOCAL cName := "James, William"  
? LEFT(cName, AT(",", cName) - 1) // Resulta: James
```

Arquivos: Biblioteca EXTEND.LIB.

ALLTRIM()

Remove espaços em branco à direita e à esquerda de uma cadeia de caracteres

Sintaxe

ALLTRIM(<cString>) --> cTrimString

Argumentos

<cString> é a expressão caractere cujos espaços em branco serão eliminados.

Retorno

ALLTRIM() retorna uma cadeia de caracteres cujos espaços em branco à direita e à esquerda foram removidos.

Descrição

ALLTRIM() é uma função de tratamento de dados do tipo caractere que

remove os espaços em branco à direita e à esquerda de uma cadeia de caracteres. É relacionada a LTRIM() e RTRIM(), que removem espaços em branco à esquerda e à direita de uma cadeia de caracteres, respectivamente. O inverso de ALLTRIM(), LTRIM(), e RTRIM() são as funções PADC(), PADR(), e PADL(), as quais centralizam, alinham à direita, ou alinham à esquerda cadeias de caracteres através da inserção de caracteres de preenchimento.

Exemplos

O exemplo a seguir cria uma cadeia de caracteres com espaços em branco à direita e à esquerda, e depois remove os espaços com ALLTRIM():

```
cString := SPACE(10) + "string" + SPACE(10)
? LEN(cString)           // Resulta: 26
? LEN(ALLTRIM(cString))  // Resulta: 6
```

Arquivos: Biblioteca EXTEND.LIB.

RIGHT()

Retorna um substring que começa com o caractere mais à direita

Sintaxe

RIGHT(<cString>, <nCont>) --> cSubString

Argumentos

<cString> é a cadeia de caracteres da qual serão extraídos caracteres.

<nCont> é a quantidade de caracteres a serem extraídos.

Retorno

RIGHT() retorna os caracteres <nCont> mais à direita de <cString>. Caso <nCont> seja negativo ou zero, RIGHT() retorna uma cadeia de caracteres nula (""). Se <nCont> for maior que o tamanho da cadeia de caracteres, RIGHT() retorna <cString>. O tamanho máximo é de 65.535 (64K) bytes.

Descrição

RIGHT() é uma função de tratamento de caracteres a qual extrai um substring que começa com o caractere mais à direita em <cString>. É o mesmo que a expressão caractere, SUBSTR(<cString>, -<nCont>). Por exemplo, RIGHT("ABC", 1) é o mesmo que SUBSTR("ABC", -1). RIGHT() é relacionado a LEFT(), a qual extrai um substring que começa com o caractere mais à esquerda em <cString>.

As funções RIGHT(), LEFT(), e SUBSTR() geralmente são utilizadas juntamente com as funções AT() e RAT() para localizar a primeira e/ou a última posição de um substring antes de extrai-lo.

Exemplos

_ Este exemplo ilustra a relação entre RIGHT() e SUBSTR():

```
? RIGHT("ABCDEF", 3)           // Resulta: DEF
? SUBSTR("ABCDEF", -3)          // Resulta: DEF
```

- _ Este exemplo extrai um substring a partir do final de uma cadeia de caracteres até a última ocorrência de uma vírgula:

```
LOCAL cName := "James,William"
? RIGHT(cName,
  LEN(cName) - RAT(",", cName) - 1)    // Resulta: William
```

Arquivos: Biblioteca CLIPPER.LIB.

VAL()

Converte um número presente em uma cadeia de caracteres para um valor numérico

Sintaxe

VAL(<cNúmero>) --> nNúmero

Argumentos

<cNúmero> é a expressão caractere a ser convertida.

Retorno

VAL() retorna <cNúmero> convertido para um valor numérico, incluindo dígitos decimais.

Descrição

VAL() é uma função de conversão de caracteres, a qual converte uma cadeia de caracteres que contém dígitos numéricos para um valor numérico. Quando VAL() é executada, ela avalia <cNúmero> até encontrar o segundo ponto decimal, o primeiro caractere não numérico, ou o final da expressão. Os espaços à direita são ignorados. Com SET FIXED em ON, VAL() retorna a quantidade de casas decimais especificadas por SET DECIMALS, arredondando <cNúmero> se ele for especificado com uma quantidade de dígitos maior do que o valor corrente de DECIMALS. Da mesma forma que todas as outras funções que arredondam, os dígitos entre zero e quatro são arredondados para baixo, e os dígitos entre cinco e nove são arredondados para cima. Com SET FIXED em OFF, VAL() retorna a quantidade de casas decimais especificadas em <cNúmero>.

VAL() é o oposto de STR() e TRANSFORM(), as quais convertem valores numéricos para cadeias de caracteres.

Exemplos

Os exemplos a seguir ilustram VAL() com SET FIXED ON, e SET DECIMALS TO 2:

```
SET DECIMALS TO 2
SET FIXED ON
//
? VAL("12.1234")    // Resulta: 12.12
? VAL("12.1256")    // Resulta: 12.13
? VAL("12A12")      // Resulta: 12.00
? VAL("A1212")      // Resulta: 0.00
? VAL(SPACE(0))     // Resulta: 0.00
? VAL(SPACE(1))     // Resulta: 0.00
? VAL(" 12.12")     // Resulta: 12.12
? VAL("12 .12")     // Resulta: 12.00
```

Arquivos: Biblioteca CLIPPER.LIB.

STR()

Converte uma expressão numérica para uma cadeia de caracteres

Sintaxe

STR(<nNúmero>, [<nTamanho>], [<nDecimais>]) --> cNúmero

Argumentos

<nNúmero> é a expressão numérica a ser convertida para uma cadeia de caracteres.

<nTamanho> é o tamanho da cadeia de caracteres a ser retornada incluindo dígitos decimais, ponto decimal, e sinal.

<nDecimais> é a quantidade de casas decimais a serem retornadas.

Retorno

STR() retorna <nNúmero> formatado como uma cadeia de caracteres. Se os argumentos opcionais de tamanho e decimais não foram especificados, STR() retorna a cadeia de caracteres conforme as seguintes regras:

Tabela 5-32: Resultados de STR() sem Argumentos

Expressão	Tamanho do Valor de Retorno
Variável Campo	Tamanho do campo mais decimais
Expressões/constantes	Mínimo de 10 dígitos mais decimais
VAL()	Mínimo de 3 dígitos
MONTH()/DAY()	3 dígitos
YEAR()	5 dígitos
RECNO()	7 dígitos

Descrição

STR() é uma função que converte valores numéricos para cadeias de caracteres. Ela é comumente utilizada para concatenar valores numéricos a cadeias de caracteres. STR() é aplicada na exibição de números, criação de códigos tais como números de produtos, e criação de chaves de indexação que combinam dados do tipo numérico e caractere.

STR() é semelhante a TRANSFORM(), que formata valores numéricos como cadeias de caracteres utilizando uma máscara ao invés de especificações de tamanho e decimais. Como TRANSFORM() usa uma máscara, ela pode inserir caracteres de formatação tais como vírgulas, cifrões, e parênteses.

O inverso de STR() é VAL(), a qual converte números presentes em cadeias de caracteres em valores numéricos.

Notas

- Se <nTamanho> for menor do que a quantidade de dígitos numéricos inteiros em <nNúmero>, STR() retorna asteriscos ao invés do número.

- _ Caso <nTamanho> seja menor do que a quantidade de dígitos decimais necessários à parte decimal da cadeia de caracteres retornada, o Clipper arredonda o número para a quantidade de casas decimais disponíveis.
- _ Se <nTamanho> for especificado, mas <nDecimais> for omitido (sem casas decimais), o valor de retorno é arredondado para um inteiro.

Exemplos

- _ Estes exemplos demonstram a faixa de valores retornados por STR(), de acordo com os argumentos especificados:

```
nNumber = 123.45
? STR(nNumber)           // Resulta: 123.45
? STR(nNumber, 4)        // Resulta: 123
? STR(nNumber, 2)        // Resulta: **
? STR(nNumber * 10, 7, 2) // Resulta: 1234.50
? STR(nNumber * 10, 12, 4) // Resulta: 1234.5000
? STR(nNumber, 10, 1)    // Resulta: 1234.5
```

- _ Este exemplo utiliza STR() para criar um índice com uma chave composta de números de ordem e nomes de clientes:

```
USE Customer NEW
INDEX ON STR(NumOrders, 9) + CustName TO CustOrd
```

Arquivos: Biblioteca CLIPPER.LIB.

SUBSTR()

Extraí um substring de uma cadeia de caracteres

Sintaxe

SUBSTR(<cString>, <nInício>, [<nCont>]) --> cSubstring

Argumentos

<cString> é a cadeia de caracteres da qual será extraído um substring, podendo ter até 65.535 (64K) bytes, que é o tamanho máximo de uma cadeia de caracteres em Clipper.

<nInício> é a posição inicial em <cString>. Se <nInício> for positivo, ele é relativo ao caractere mais à esquerda em <cString>. Se <nInício> for negativo, ele é relativo ao caractere mais à direita em <cString>.

<nCont> é a quantidade de caracteres a serem extraídos. Se omitido, o substring começa em <nInício> e continua até o fim da cadeia de caracteres. Se <nCont> for maior do que a quantidade de caracteres existentes a partir de <nInício> até o final de <cString>, o excedente é ignorado.

Retorno

SUBSTR() retorna uma cadeia de caracteres.

Descrição

SUBSTR() é uma função de tratamento de caracteres que extrai um

substring de qualquer outra cadeia ou campo memo. SUBSTR() está relacionada às funções LEFT() e RIGHT(), que extraem substrings que começam com os caracteres mais à esquerda e mais à direita em <cString>, respectivamente.

As funções SUBSTR(), RIGHT(), e LEFT() são utilizadas juntamente com as funções AT() e RAT() para localizar a primeira e/ou última posição de um substring antes de extrair-lo. Elas também são utilizadas para exibir ou imprimir apenas uma parte de uma cadeia de caracteres.

Exemplos

_ Os exemplos a seguir extraem o primeiro e o último nome de uma variável:

```
cName := "Biff Styvesent"
? SUBSTR(cName, 1, 4)           // Resulta: Biff
? SUBSTR(cName, 6)             // Resulta: Styvesent
? SUBSTR(cName, LEN(cName) + 2) // Resulta: null string
? SUBSTR(cName, -9)            // Resulta: Styvesent
? SUBSTR(cName, -9, 3)         // Resulta: Sty
```

_ Este exemplo utiliza SUBSTR() juntamente com AT() e RAT() para criar uma função definida pelo usuário com a finalidade de extrair um nome de arquivo de uma especificação dada:

```
? FileBase("C:\PRG\MYFILE.OBJ") // Resulta: MYFILE.OBJ
```

```
FUNCTION FileBase( cFile )
  LOCAL nPos
  IF (nPos := RAT("\", cFile)) != 0
    RETURN SUBSTR(cFile, nPos + 1)
  ELSEIF (nPos := AT(":", cFile)) != 0
    RETURN SUBSTR(cFile, nPos + 1)
  ELSE
    RETURN cFile
  ENDIF
```

Arquivos: Biblioteca CLIPPER.LIB.

STRTRAN()

Pesquisa e substitui caracteres de uma cadeia de caracteres ou campo memo

Sintaxe

```
STRTRAN(<cString>, <cProcura>,
  [<cSubstitui>], [<nInício>], [<nCont>]) --> cNovoString
```

Argumentos

<cString> é a cadeia de caracteres ou campo memo onde será efetuada a busca.

<cProcura> é a sequência de caracteres a ser pesquisada.

<cSubstitui> é a sequência de caracteres que substituirá <cProcura>. Se este argumento não for especificado, as ocorrências especificadas do argumento de pesquisa são substituídas por uma cadeia de caracteres nula ("").

<nInício> é a primeira ocorrência a ser substituída. Caso este

argumento seja omitido, o padrao é um.

<nCont> é a quantidade de ocorrências a serem substituídas. Se este argumento não for especificado, o padrao é todos.

Retorno

STRTRAN() retorna uma nova cadeia de caracteres, sendo que as ocorrências especificadas de <cProcura> foram substituídas por <cSubstitui>.

Descrição

STRTRAN() é uma função de tratamento de caracteres que faz a pesquisa de um substring padrao em uma cadeia de caracteres. Ao ser encontrada uma correspondência, ela substitui a cadeia de caracteres encontrada pela cadeia de caracteres de substituição. Todas as ocorrências de <cProcura> são substituídas, a não ser que seja especificado <nInício> ou <nCont>. Note que STRTRAN() substitui substrings e, portanto, não considera palavras inteiras.

Exemplos

O exemplo a seguir utiliza STRTRAN() para estabelecer uma analogia pós-moderna com uma frase famosa do período pré-Iluminismo:

```
cString = "programar ou não programar?"  
? STRTRAN(cString, "programar", "ser")    // Resulta: ser ou não ser?
```

Arquivos: Biblioteca CLIPPER.LIB.