

# FoxPro Functions

## **ALLTRIM**

ALLTRIM(expC)

ALLTRIM() returns the character expression expC minus any leading and trailing blanks. ALLTRIM() is functionally equivalent to LTRIM(RTRIM(expC)).

---

## **AT**

AT(expC1, expC2[,expN])

The AT() function finds expC1 in expC2. (Note that expC2 may be a memo field.) The function returns as an integer the starting position of expC1. If expC1 is not found, the function returns a zero. If the optional expN is used, the expression Nth occurrence of expC1 is searched for.

---

## **ATC**

ATC(expC1, expC2[,expN])

The ATC() function searches a character string expC1 for another character string expC2. If expC1 is not found, the function returns a zero. If the optional expN is used, the expNth occurrence of expC1 is search for. The ATC() function operates just like the AT() function, but the ATC function is not case-sensitive.

---

## **BETWEEN**

BETWEEN(expr1, expr2, expr3)

The BETWEEN() function returns a logical true (.T.) if expr1 is greater than or equal to expr2 and less than or equal to expr3; otherwise, the function returns a logical false(.F.) The expressions used must be of the same type.

---

## **CMONTH**

CMONTH(expD)

The CMONTH() function returns the name of the moth that corresponds to the date expression.

## **CTOD**

CTOD(expC)

The CTOD() function returns the date value which corresponds to expC, in the default date format (generally MM/DD/YY). Use the SET DATE and SET CENTURY commands to change the default format. Also, note that curly braces surrounding a date perform the equivalent of the CTOD() function; hence, CTOD("12/12/93") and {12/21/93} produce the same date value in FoxPro.

---

## **DATE**

DATE()

The DATE() function returns the current system date, as measured by the system clock.

---

## **DAY**

Day(expD)

The DAY() function returns the numeric day of the month that correspond to the date expression.

---

## **DOW**

DOW(expD)

The DOW() function returns the numeric day of the week corresponding to the date expression. The value returned ranges from 1 (Sunday) to 7 (Saturday).

---

## **DTOC**

DTOC(expD[,1])

The DTOC() function returns a character string containing the date that corresponds to the date expression. Use the SET DATE and the SET CENTURY commands to change the format of the string. The optional ,1 argument causes DTOC() to return the string in the YYYYMMDD format, similar to the DTOS()function.

## **DTOS**

DTOS(*expD*)

The DTOS() function returns a character string in the format YYYYMMDD for the given date expression. This function is useful when indexing on a date field.

---

## **EMPTY**

EMPTY(*expression*)

The EMPTY() function returns a logical true (.T.) if the expression is blank. The function will also return a value of true if the expression is a numeric expression with a value of zero or a logical expression with a value of false. Not that FoxPro will also return a value of true if the expression is a memo field, and the memo field contains no text.

---

## **GOMONTH**

GOMONTH(*expD*, *expN*)

The GOMONTH() function returns a date that is *expN* months before or after *expD*. If *expN* is positive, the date returned is *expN* months after *expD*. If *expN* is negative, the date returned is *expN* months before *expD*.

---

## **IFF**

IFF(*expL*, *expr1*, *expr2*)

The IFF() function (Immediate IF) returns the value of *expr1* if the logical expression is true and returns the value of *expr2* if the logical expression is false. *Expr1* and *expr2* must be of the same data type.

---

## **ISALPHA**

ISALPHA(*expC*)

The ISALPHA() function returns a logical true (.T.) if the first character of *expC* is “a”-“z” or “A”-“Z.” A logical (.F.) is returned if *expC* begins with a nonalphabetic or a numeric character.

## **LEFT**

LEFT(expC, expN)

The LEFT() function returns the leftmost number of characters specified in expN from the character expression expC, starting with the first or leftmost character.

---

## **LEN**

LEN(expC)

The LEN() function returns the length of a character string expression specified in expC. ExpC can be a memo field name, in which case the length of the text stored within the memo field is returned. Note that in the case of character fields, LEN() returns the length of the field, not the length of the text within the field. With character fields, you must add a TRIM() function to get the length of the text stored in the field.

---

## **LIKE**

LIKE(expC1, expC2)

The LIKE() function compares two character expressions and returns a logical true (.T.) if the character string in expC2 contains the characters in expC1. The pattern can include the wildcard characters \*(representing any sequence of characters) and ? (representing any single character).

---

## **LTRIM**

LTRIM(expC)

The LTRIM() function trims all leading blanks from the character expression defined as expC.

---

## **MAX**

MAX(expr1, expr2[ , expr3...])

The MAX() function returns the maximum value from the list of expressions. The expression must all be of the same data type. The expressions can be numeric or date expressions; if date expressions are used, MAX() returns the later of the dates.

## **MIN**

MIN(expr1, expr2[, expr3...])

The MIN() function returns the minimum value expression from the list of expressions. The expressions must all be of the same data type. The expressions can be numeric or date expressions; if date expressions are used MIN() returns the earlier of the dates.

---

## **MONTH**

MONTH(expD)

The MONTH() function returns the numeric month (1 to 12) that corresponds to the date expression. The numbers 1 through 12 correspond to January through December.

---

## **RIGHT**

RIGHT(expC or memvar, expN)

The RIGHT() function returns the rightmost part of the character string expC or memory variable memvar. Use the numeric expression expN to specify the number of characters that will be returned.

---

## **ROUND**

ROUND(expN1, expN2)

The ROUND() function rounds off the number supplies in expN1. Use expN2 to specify the number of decimal places to round off to. If expN2 is negative, the rounded number returned is a whole number.

---

## **SIGN**

SIGN(expN)

SIGN() returns a numeric value that represents the sign of the numeric expression. If expN is positive, SIGN() returns a value of 1. If expN is negative, SIGN() returns a value of -1. If expN is zero, SIGN() returns a zero.

## **STR**

STR(expN1[ , expN2[ , expN3]])

The STR() function converts a numeric expression to a character expression, where expN1 is the numeric expression to be converted to a character string. Use the optional expN2 to specify a length (including the decimal point and decimal places), and use the optional expN3 to specify a number of decimal places.

---

## **STRTRAN**

STRTRAN(expC1, expC2[ ,expC3][ ,expN1][ , expN2])

The STRTRAN() function performs a search-and-replace operation on a character string. The function returns the given expression expC1 with occurrences of expC2 replaced with expC3. If expC3 is omitted, the found text is replaced with a null string (“”). Replacements start at the expN1th occurrence and continue for a total of expN2 replacements. If expN1 is omitted, replacements start at the beginning of the string. If expN2 is omitted, all needed replacement are made.

---

## **STUFF**

STUFF(expC1, expN1, expN2, expC2)

The STUFF() function inserts or removes characters from any part of a character string. ExpC1 is the existing character string, expN1 is the starting position in the string, expN2 is the number of characters to remove, and expC2 is the character to insert.

---

## **SUBSTR**

SUBSTR(expC, expN1[ , expN2])

The SUBSTR() function extracts a portion of a string from a character expression. ExpC is the character expression to extract the string from expN1 is the starting position in the expression, and expN2 is the number of characters to extract from the expression.

## **TRIM**

TRIM(expC)

The TRIM() function trims trailing spaces from a character string. If the character string is composed entirely of spaces, TRIM() returns a null string. The TRIM() function is identical to the RTRIM() function.

---

## **YEAR**

YEAR(expD)

The YEAR() function returns the numeric year corresponding to the date expression.