Text and Word Functions

Objectives

Text and Word Functions

In this module, you will examine text and word functions and their syntax in a REXX program.

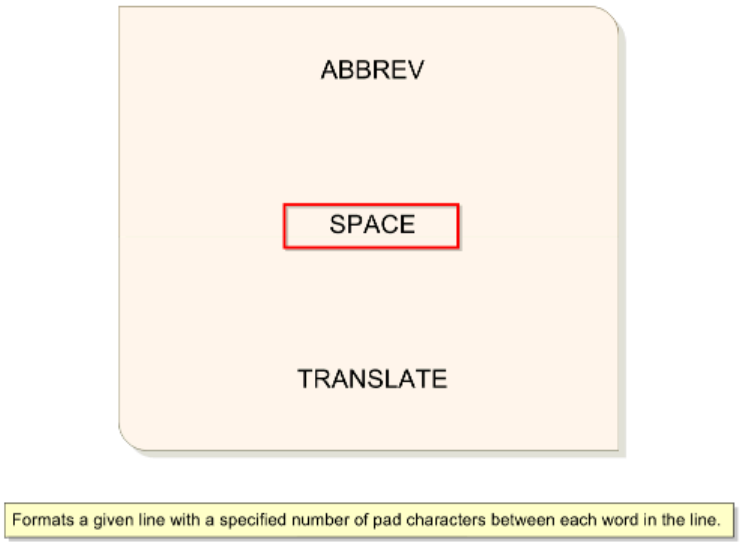
After completing this module, you will be able to:

- Identify Text Functions
- Identify Word Functions

String functions	Perform various comparison, interrogation, and manipulation actions on data strings.
Text and word functions	Interrogate and manipulate words and specific data within a string.
Justification functions	Justify and format text and data strings.
Numeric functions	Interrogate and format numeric values.
Character conversion functions	Convert and manipulate binary, hex, and character values.
Environment functions	Interrogate the environment that the REXX program is running under, and return settings and definitions.
Stream I/O functions	Used for file processing on many platforms.

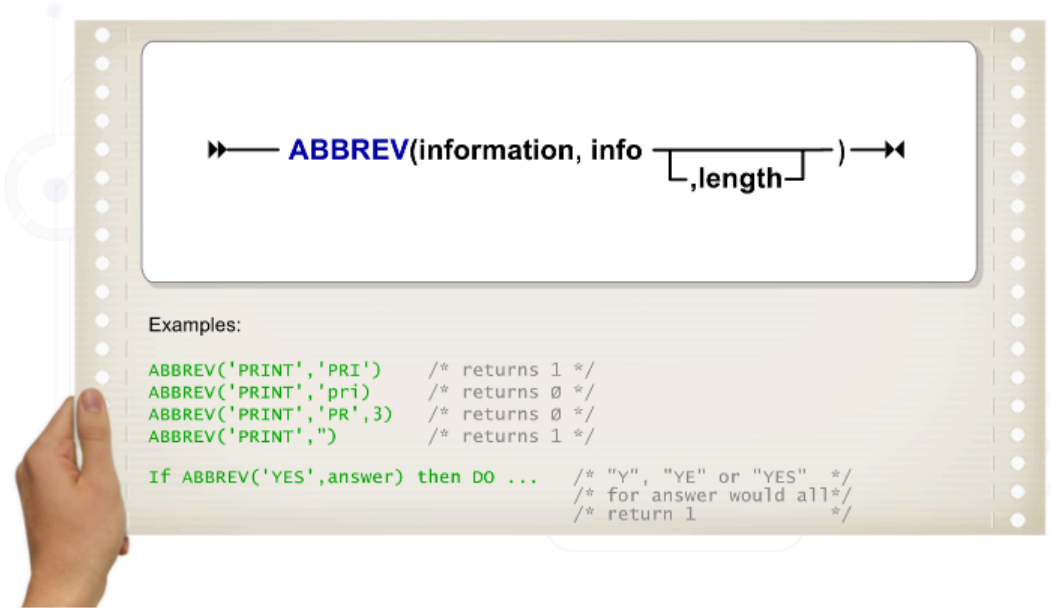
For the purposes of this course, we have divided the built-in functions into the groups listed above.

You will now focus on text and word functions.



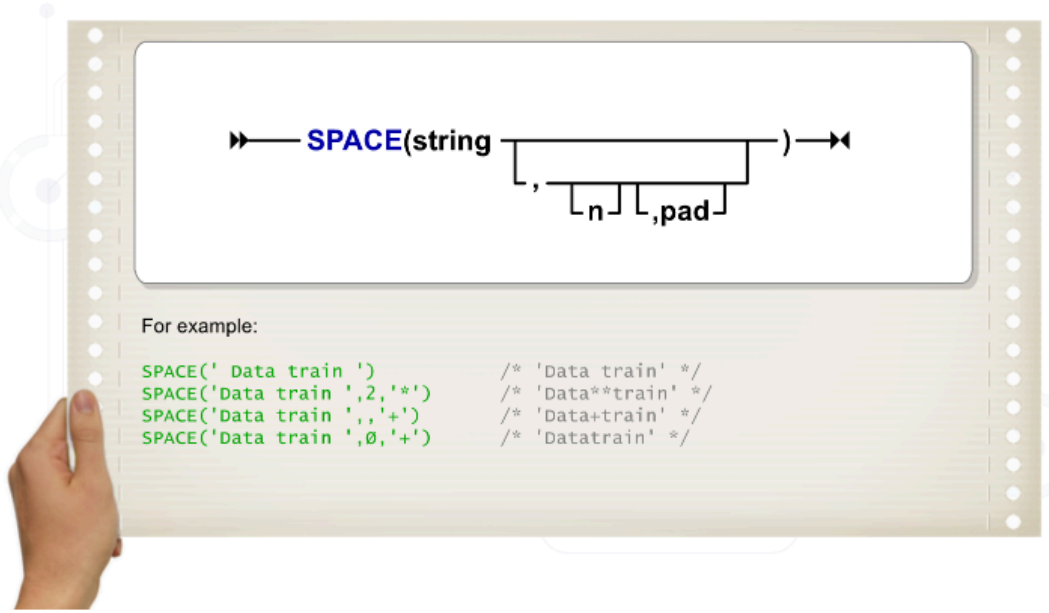
Text functions interrogate or manipulate specific text within a character string. Listed above are the standard built-in functions in this loosely defined category.

Mouse-over each function for a brief description.



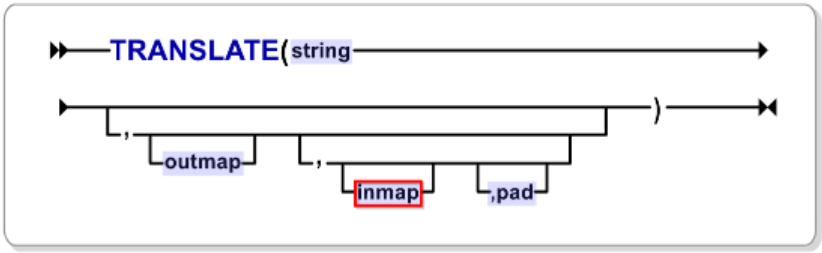
The ABBREV function compares the first `length` characters of `information` to `info` and returns 1 (true) if they match exactly, including case; otherwise, 0 (false) is returned.

The default for `length` is the number of characters in `info`.



The SPACE function replaces all spaces between each word in `string` with the number of `pad` characters specified by `n`. Leading and trailing blanks are always removed.

The default for `n` is 1 and the default for `pad` is a space.

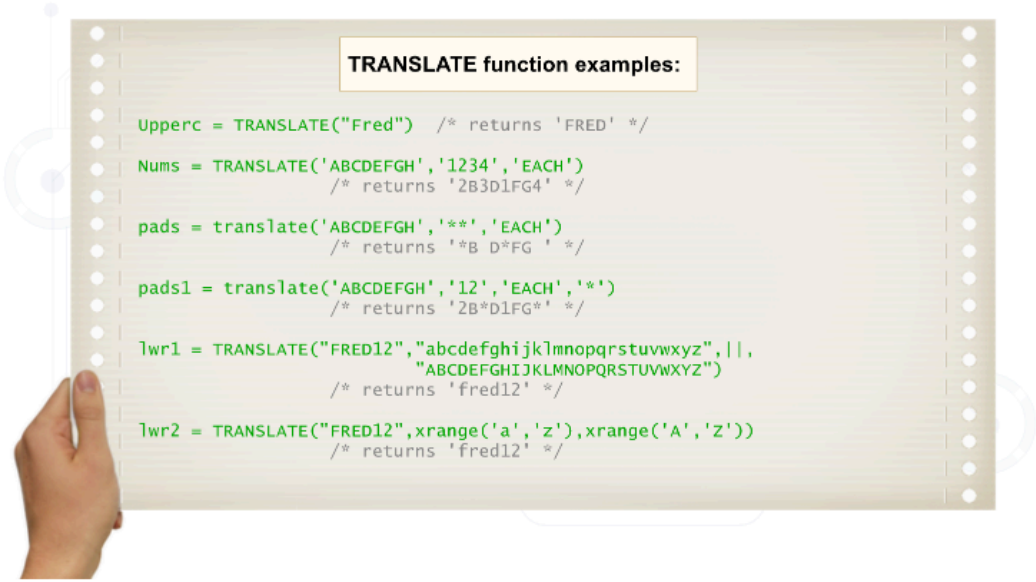


A list or "map" of characters that are to be translated; the translation of characters from inmap to outmap is normally performed on a one-to-one basis, that is, each character in inmap is changed to the character in the corresponding position of outmap.

The TRANSLATE function can be used to translate one specific character in a string to another specific character as defined by a character map.

Character maps are always a one-to-one relationship and never work as complete strings. If `string` is the only argument defined, the result will be 'string' in uppercase.

Mouse-over the syntax for a description of each argument.



If only the string is passed to the TRANSLATE function, it is returned with all alpha characters translated to uppercase.

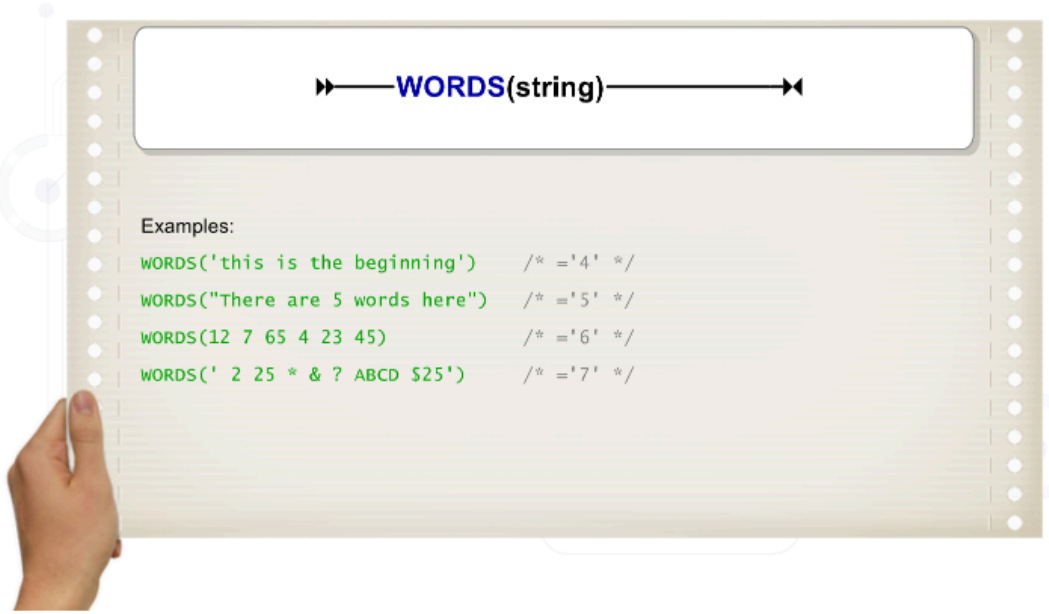
While there is an UPPER keyword instruction in z/OS/VM only, and the default TRANSLATE function to convert a value to uppercase, there is no equivalent for lowercase. The last two examples above show how alpha characters in a value or string can be translated to lowercase by using the TRANSLATE function.

When outmap has fewer characters than inmap, the later characters of inmap will be translated to blanks or, if specified, the pad character.

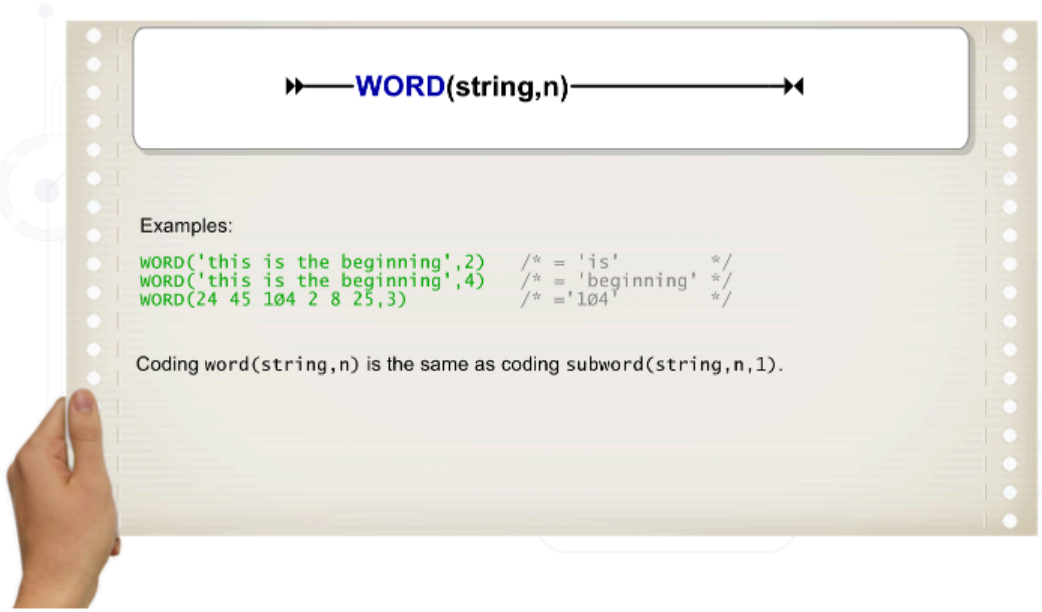
SUBWORD	Returns a substring of a given string from a specified word for a specified number of words.
WORD	Returns the nth specified word in a string.
WORDINDEX	Returns the position of the first character in a specified word in a string.
DELWORD	Deletes one or more words, starting at a specified word.
WORDLENGTH	Returns the length of a specified word.
WORDPOS	Returns the position of a specified word or phrase in a given string.
WORDS	Returns the number of words in a given string.
FIND	Returns the word number of the first occurrence of a given string in a phrase. WORDPOS is usually preferred.

In REXX, a word is defined as a blank delimited set of characters within a string. The characters in a word can be alphanumeric or special characters, or hex or binary characters as the REXX interpreter does not differentiate between them within a string.

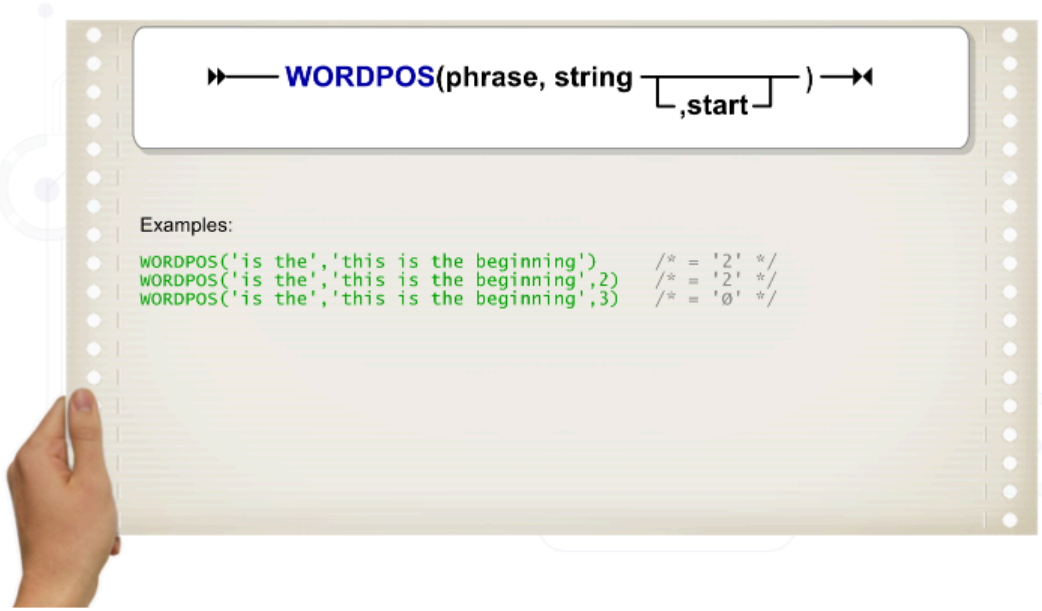
Word functions are designed to process words within a string. Listed above are some word functions and their descriptions.



The WORDS function returns the number of words that are contained in a string.



The WORD function returns the nth word from string. Words are considered to be groups of characters separated by blanks.

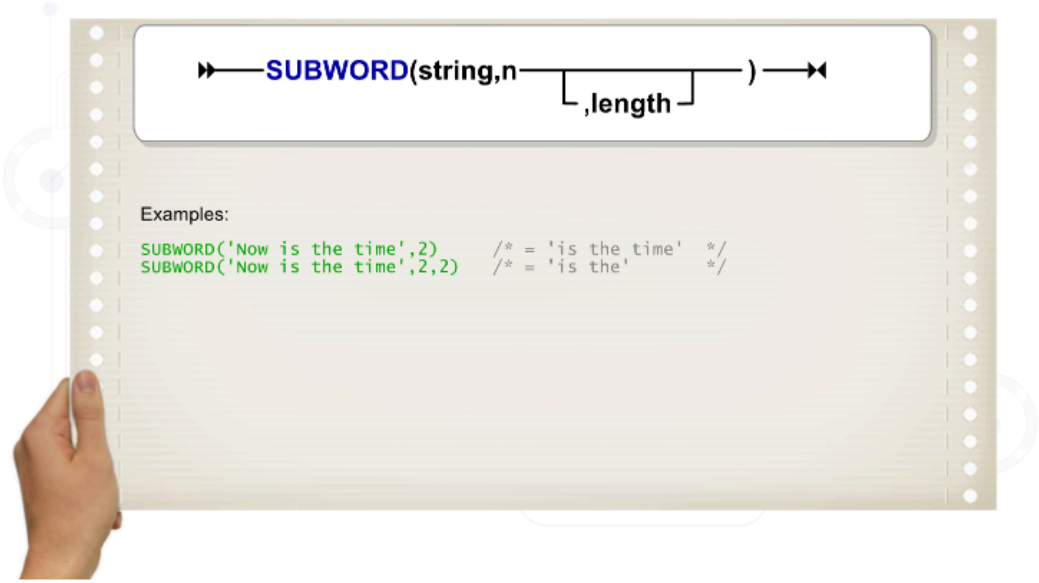


The WORDPOS function returns the word number of the first word of phrase from `string`. If `start` is specified, it identifies the word in `string` at which to begin the search. If `phrase` is not found, WORDPOS returns 0.

The FIND function could be used instead of WORDPOS, but the syntax is slightly different, for example:

```
FIND(string,phrase[,start])
```

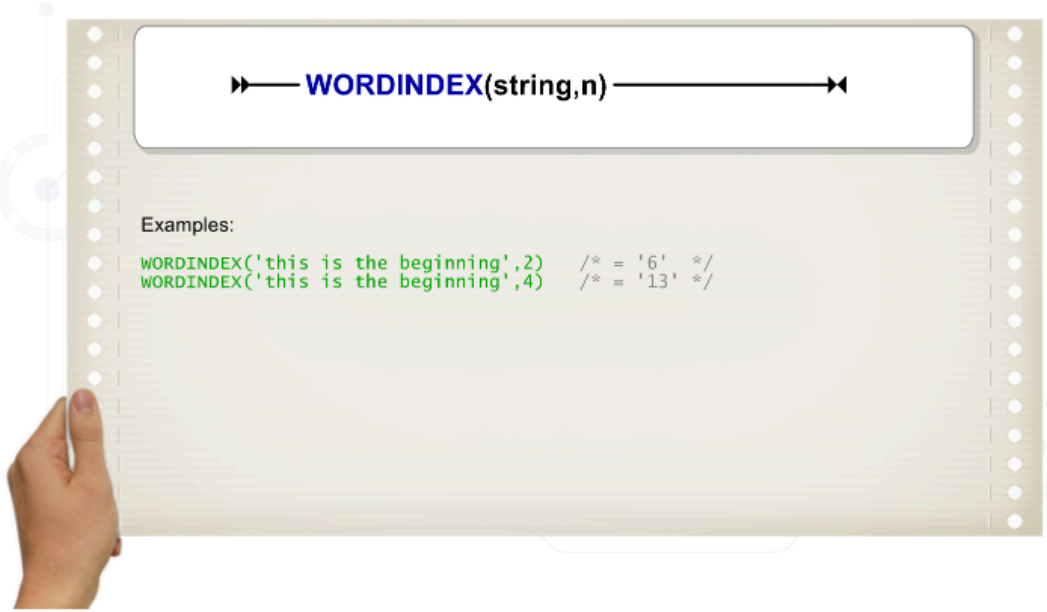
WORDPOS is preferred for standardization reasons.



The SUBWORD function is similar to the SUBSTR function, except that `n` specifies the `n`th word of the string, and `length` specifies the number of words in the returned string. If `length` is not specified, it defaults to the rest of the words in the string.

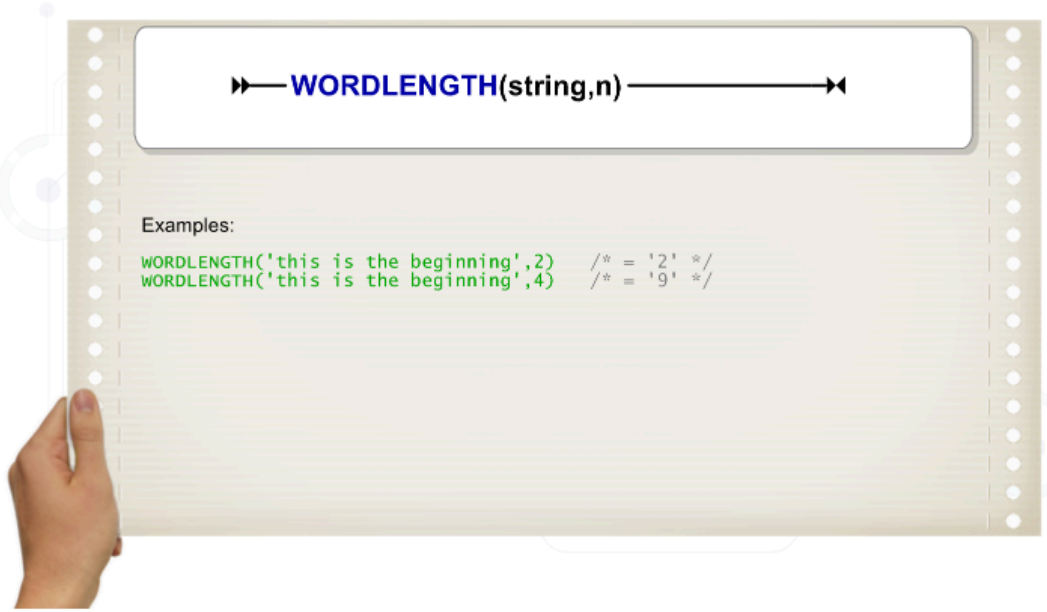
Specifying `SUBWORD(string,3,1)` is the same as specifying the following:

```
WORD(string,3)
```



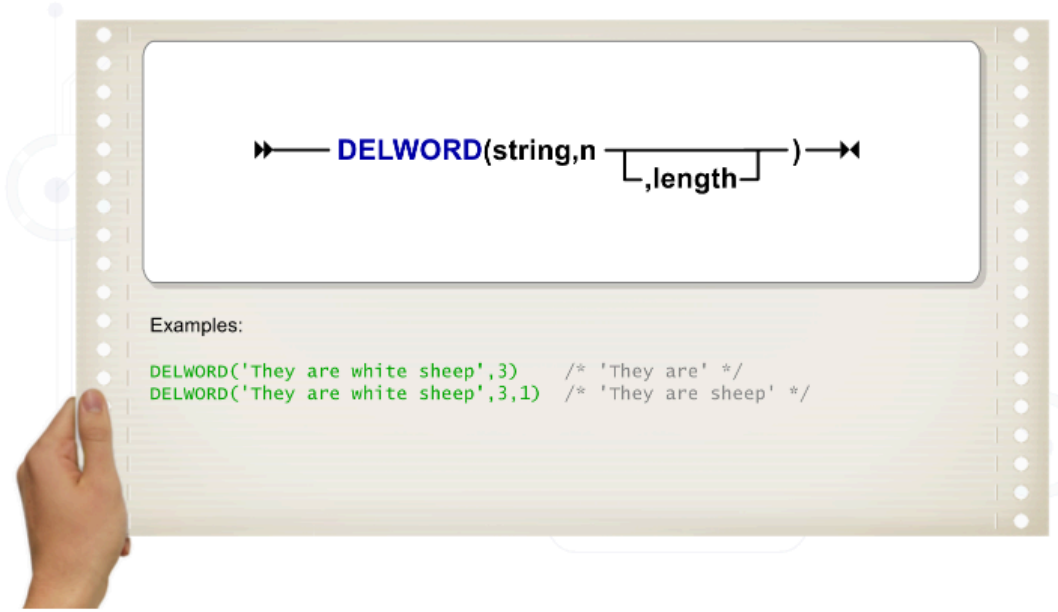
The WORDINDEX function returns the character position of the first character of word `n` from `string`.

If there are fewer than `n` words in `string`, WORDINDEX returns 0.



The WORDLENGTH function returns the length of word `n` from `string`.

If there are fewer than `n` words in `string`, WORDLENGTH returns 0.



The DELWORD function returns `length` words deleted from `string`, starting at word `n`.

If `length` is not specified, all words from `n` are deleted.

```
/* REXX TEXT and WORD Function examples */
rec1 = 'frederick frog'
rec2 = 'addr: 65 best St. state: Vic postcode: 1234'

fname = word(rec1,1)                  /* fred          */
sname = word(rec1,2)                  /* frog          */

fname1 = translate(fname,'F','f')     /* Fred          */

if abbrev('frederick',fname) then     /* 1 - True      */
 do
  adrstrt = wordpos('addr:',rec2)     /* 1 (1st word) */
  statestrt = wordpos('state:',rec2)  /* 5 (5th word) */
  addr = subword(rec2,adrstrt+1,statestrt-adrstrt-1)
                                      /*65 best St.    */
  state = word(rec2,statestrt+1)      /* Vic           */
  state = translate(state)            /* VIC           */
  fulladdr = delword(rec2,statestrt,1)
                /*addr: 65 best St. Vic postcode: 1234*/
 end
nowords = words(rec2)                 /* 8             */
```

Here are some examples of text and word functions in use.