

string — String manipulation functions

[Contents](#)[Description](#)[Remarks and examples](#)[Also see](#)

Contents

[M-5] Manual entry	Function	Purpose
Parsing		
tokens()	<code>tokens()</code>	obtain tokens (words) from string
invtokens()	<code>invtokens()</code>	concatenate string vector into string scalar
strmatch()	<code>strmatch()</code>	pattern matching
tokenget()	<code>...</code>	advanced parsing
Length & position		
strlen()	<code>strlen()</code>	length of string
fmtwidth()	<code>fmtwidth()</code>	width of <i>%fmt</i>
strpos()	<code>strpos()</code>	find substring within string
indexnot()	<code>indexnot()</code>	find character not in list
Editing		
substr()	<code>substr()</code>	extract substring
strupper()	<code>strupper()</code>	convert to uppercase
	<code>strlower()</code>	convert to lowercase
	<code>strproper()</code>	convert to proper case
strtrim()	<code>stritrim()</code>	replace multiple, consecutive internal blanks with one blank
	<code>strltrim()</code>	remove leading blanks
	<code>strrtrim()</code>	remove trailing blanks
	<code>strtrim()</code>	remove leading and trailing blanks
subinstr()	<code>subinstr()</code>	substitute text
	<code>subinword()</code>	substitute word
_substr()	<code>_substr()</code>	substitute into string

Editing, *continued*

strdup()	*	duplicate string
strreverse()	strreverse()	reverse string
soundex()	soundex() soundex_nara()	convert to soundex code convert to U. S. Census soundex code

Stata

abbrev()	abbrev()	abbreviate strings
strtoname()	strtoname()	translate strings to Stata names

ASCII translation

stroofreal()	stroofreal()	convert real to string
strtoreal()	strtoreal()	convert string to real
ascii()	ascii() char()	obtain ASCII codes of string make string from ASCII codes

Description

The above functions are for manipulating strings. Strings in Mata are strings of ASCII characters, usually the printable characters, but Mata enforces no such restriction. In particular, strings may contain binary 0.

Remarks and examples

[stata.com](http://www.stata.com)

In addition to the above functions, two operators are especially useful for dealing with strings.

The first is +. Addition is how you concatenate strings:

```
: "abc" + "def"  
abcdef  
:  
: command = "list"  
:  
: args = "mpg weight"  
:  
: result = command + " " + args  
:  
: result  
list mpg weight
```

The second is `*`. Multiplication is how you duplicate strings:

```
: 5*"a"  
aaaaa  
: "b"*3  
bbb  
: indent = 20  
: title = indent*" " + "My Title"  
: title  
My Title
```

Also see

[M-4] [intro](#) — Index and guide to functions