Title

txt —— Prints string and values of scalar expressions or macros in dynamic document. By default, the command writes a text paragraph. The primary purpose of the command is writing dynamic text to interpret analysis results in the dynamic document. This command belongs to Weaver package, but it also supports the MarkDoc package. The syntax for both packages is similar, but the **txt** command behaves differently based on which package is in use. If Weaver log is on, **txt** functions for Weaver package only. This document only describes the **txt** command in Weaver package. For using the command in MarkDoc package see the MarkDoc documentation on GitHub wiki

Syntax

Prints dynamic text on the Weaver log or smcl log. The **code** subcommand prints the output "as is" in the dynamic document.

txt [**code**] [*display_directive* [*display_directive* [...]]]

where the *display_directive* is

Weaver Markup
"*double-quoted string* "
`"*compound double-quoted string* "'
[**%***fmt*] [**=**]*exp*
_skip(#)
_column(#)
_newline[**(#)**]
_dup(#)
,
,,

Description

txt prints dynamic text i.e. strings and values of scalar expressions or macros in the Weaver or the smcl log-file. **txt** also prints output from the user-written Stata programs. Any of the supported markup languages can be used to alter the string and scalar expressions. This command is to some extent similar to **display** command in Stata. For example, it can be used to carry out a mathematical calculation by typing **txt 1+1**. It also supports many of the display directives as well.

Note that in contrast to the display command that prints the scalar unformatted, the **txt** command uses the default **%10.2f** format for displaying the scalar. This feature helps the users avoid specifying the format for every scalar, due to popularity of this format. However, specifying the format expression can overrule the default format. For example:

```
. scalar num = 10.123
. txt "The value of the scalar is " %5.1f num
```

The example above will print the scalar with only 1 decimal number. This feature only supports scalar interpretation and does not affect the macro contents.

Display directives

The supported *display_directive* s are used in do-files and programs to produce formatted output. The directives are

Weaver Markup	A simplified markup language for annotating the content of the HTML log
"*double-quoted string* "	displays the string without the quotes
`"*compound double-quoted string* "'	display the string without the outer quotes; allows embedded quotes
[**%***fmt*] [**=**] **exp**	allows results to be formatted; see **[U] 12.5 Formats: Controlling how data are displayed**.
_skip(#)	skips # columns
_column(#)	skips to the #th column
_newline	goes to a new line
_newline(#)	skips # lines
_dup(#)	repeats the next directive # times
,	displays one blank between two directives

,, places no blanks between two directives

Mathematical notations

The **txt** command can be used for writing mathematical notations in Weaver package, both in HTML and LaTeX log files. Writing mathematical notations in the HTML log is made possible by including **MathJax** engine, a JavaScript-based engine for rendering LaTeX notations in HTML format. To do so, notations should begin with "**\(**" and end with "**\)**" for rendering notations within the text and double dollar sign "**$$**" or alternatively, the "**\[**" and "**\]**" for rendering notations in a separate line. For more information in this regard, see mathematical notations documentation.

When Weaver package is running, the **code** subcommand appends the dynamic text to the

Examples

As a hand calculator:

```
. txt 2 * 2
```

As might be used in do-files and programs:

```
. sysuse auto
. summarize price
. txt "mean of Price variable is " r(mean) " and SD is " %9.3f r(sd)
```

If the text only includes string and macro, the double quotations can be ignored. The **txt** command will interpret all of the *display_directives* and scalars as string (so it's not recommended):

```
. local n 9.9
. txt Not recommended, but you may also print the value of `n' without double quote
```

Author

E. F. Haghish
Center for Medical Biometry and Medical Informatics
University of Freiburg, Germany
and
Department of Mathematics and Computer Science
University of Southern Denmark
haghish@imbi.uni-freiburg.de

Weaver Homepage
Package Updates on Twitter
