

**eltype()** — Element type, organizational type, and type name of object

Description  
Diagnostics

Syntax  
Also see

Remarks and examples

Conformability

## Description

`eltype()` returns the current *eltype* of the argument.

`orgtype()` returns the current *orgtype* of the argument.

`classname()` returns the name of the class for a Mata class scalar.

`structname()` returns the name of the struct for a Mata struct scalar.

See [M-6] [Glossary](#) for a definition of *eltype* and *orgtype*.

## Syntax

*string scalar* `eltype(X)`

*string scalar* `orgtype(X)`

*string scalar* `classname(X)`

*string scalar* `structname(X)`

## Remarks and examples

stata.com

If  $X$  is a matrix (syntax 1), returned is

<code>eltype(X)</code>	<code>orgtype(X)</code>
<code>real</code>	<code>scalar</code>
<code>complex</code>	<code>rowvector</code>
<code>string</code>	<code>colvector</code>
<code>pointer</code>	<code>matrix</code>
<code>struct</code>	
<code>class</code>	

The returned value reflects the current contents of  $X$ . That is,  $X$  might be declared a **transmorphic matrix**, but at any instant, it contains something, and if that something were 5, returned would be "real" and "scalar".

For `orgtype()`, returned is "scalar" if the object is currently  $1 \times 1$ ; "rowvector" if it is  $1 \times k$ ,  $k \neq 1$ ; "colvector" if it is  $k \times 1$ ,  $k \neq 1$ ; and "matrix" otherwise (it is  $r \times c$ ,  $r \neq 1$ ,  $c \neq 1$ ).

$X$  can be a function (syntax 2). Returned is

<code>eltype*(<i>&amp;func()</i>)</code>	<code>orgtype*(<i>&amp;func()</i>)</code>
<code>transmorphic</code>	<code>matrix</code>
<code>numeric</code>	<code>vector</code>
<code>real</code>	<code>rowvector</code>
<code>complex</code>	<code>colvector</code>
<code>string</code>	<code>scalar</code>
<code>pointer</code>	<code>void</code>
<code>struct</code>	
<code>structdef</code>	
<code>class</code>	
<code>classdef</code>	

These types are obtained from the declaration of the function.

Aside: `struct` and `structdef` have to do with structures; see [M-2] [struct](#). `structdef` indicates that the function not only returns a structure but is the routine that defines the structure as well. `class` and `classdef` have to do with Mata classes; see [M-2] [class](#). `classdef` indicates the function not only returns a class but is the routine that defines the class as well.

`classname()` returns the name "cA" if the object is a class cA scalar. The function returns "" if the object has element type other than class or has organizational type other than scalar.

`structname()` returns the name "sA" if the object is a struct sA scalar. The function returns "" if the object has element type other than struct or has organizational type other than scalar.

## Conformability

`eltype(X)`, `orgtype(X)`, `classname(X)`, `structname(X)`:

$X$ :  $r \times c$   
*result*:  $1 \times 1$

## Diagnostics

None.

## Also see

[M-5] [isreal\(\)](#) — Storage type of matrix

[M-5] [isview\(\)](#) — Whether matrix is view

[M-4] [Utility](#) — Matrix utility functions