

Description

The conditional operator returns b if a is true (a is not equal to 0) and c otherwise.

Syntax

$$a \text{ ? } b : c$$

where a must evaluate to a real scalar, and b and c may be of any type whatsoever.

Remarks and examples

Conditional operators

```
dof = (k==0 ? n-1 : n-k)
```

are more compact than the `if-else` alternative

```
if (k==0) dof = n-1
else      dof = n-k
```

and they can be used as parts of expressions:

```
mse = ess/(k==0 ? n-1 : n-k)
```

Conformability

$a \text{ ? } b : c$:

a :	1×1	
b :	$r_1 \times c_1$	
c :	$r_2 \times c_2$	
$result$:	$r_1 \times c_1$	or $r_2 \times c_2$

Diagnostics

In $a \text{ ? } b : c$, only the necessary parts are evaluated: a and b if a is true, or a and c if a is false. However, the `++` and `--` operators are always evaluated:

```
(k==0 ? i++ : j++)
```

increments both i and j , regardless of the value of k .

Also see

[M-2] [exp](#) — Expressions

[M-2] [Intro](#) — Language definition

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

