### Description

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

### Syntax

\[
a ? b : c
\]

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

\[
\begin{align*}
\text{if } (k==0) & \quad \text{dof} = n-1 \\
\text{else} & \quad \text{dof} = n-k
\end{align*}
\]

and they can be used as parts of expressions:

\[
\text{mse} = \text{ess}/(k==0 ? n-1 : n-k)
\]

### Conformability

\[
a ? b : c:
\]

<table>
<thead>
<tr>
<th></th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>1 × 1</td>
<td>( r_1 \times c_1 )</td>
<td>( r_2 \times c_2 )</td>
<td>( r_1 \times c_1 ) or ( r_2 \times c_2 )</td>
</tr>
</tbody>
</table>

### Diagnostics

In \( a ? b : c \), 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