

Operator Syntax and Precedence

Description

Outlines **R** syntax and gives the precedence of operators.

Details

The following unary and binary operators are defined. They are listed in precedence groups, from highest to lowest.

:: :::	access variables in a namespace
\$ @	component / slot extraction
[[[indexing
^	exponentiation (right to left)
- +	unary minus and plus
:	sequence operator
%any% >	special operators (including %% and %/%)
* /	multiply, divide
+ -	(binary) add, subtract
< > <= >= == !=	ordering and comparison
!	negation
& &&	and
	or
~	as in formulae
-> ->>	rightwards assignment
<- <<-	assignment (right to left)
=	assignment (right to left)
?	help (unary and binary)

Within an expression operators of equal precedence are evaluated from left to right except where indicated. (Note that = is not necessarily an operator.)

Examples

```
## Logical AND ("&&") has higher precedence than OR ("||"):  
TRUE || TRUE && FALSE # is the same as  
TRUE || (TRUE && FALSE) # and different from  
(TRUE || TRUE) && FALSE
```

```
## Special operators have higher precedence than "!" (logical NOT).  
## You can use this for %in% :  
! 1:10 %in% c(2, 3, 5, 7) # same as !(1:10 %in% c(2, 3, 5, 7))  
## but we strongly advise to use the "!( ... )" form in this case!
```

```
## '=' has lower precedence than '<-' ... so you should not mix them  
## (and '<-' is considered better style anyway):  
## Consequently, this gives a ("non-catchable") error  
x <- y = 5 #-> Error in (x <- y) = 5 : ....
```

[Package *base* version 4.2.0 [Index](#)]