R Infix Operator (With Examples)

R Infix Operator (With Examples)

In this article, you'll learn about infix operators; how they actually work in R and how you can create an infix operator yourself. Also, you'll learn different predefined infix operators in R programming.

Most of the operators that we use in R are binary operators (having two operands). Hence, they are infix operators, used between the operands. Actually, these operators do a function call in the background.

For example, the expression `a+b` is actually calling the function `` `+`() `` with the arguments `a` and `b`, as `` `+`(a, b) ``.

Note: the back tick (`` ` ``), this is important as the function name contains special symbols.

Following are some example expressions along with the actual functions that get called in the background.

Example: How infix operators work in R?

```
> 5+3
[1] 8
> `+`(5,3)
[1] 8
> 5-3
[1] 2
> `-`(5,3)
[1] 2
> 5*3-1
[1] 14
> `-`(`*`(5,3),1)
[1] 14
```

It is possible to create user-defined infix operators in R. This is done by naming a function that starts and ends with `%`.

Following is an example of user-defined infix operator to see if a number is exactly divisible by another.

Example: User defined infix operator

```
`%divisible%` <- function(x,y)
{
if (x%%y ==0) return (TRUE)
else        return (FALSE)
}
```

This function can be used as infix operator `a %divisible% b` or as a function call `` `%divisible%`(a, b) ``. Both are the same.

```
> 10 %divisible% 3
[1] FALSE
> 10 %divisible% 2
[1] TRUE
> `%divisible%`(10,5)
[1] TRUE
```

Things to remember while defining your own infix operators are that they must start and end with `%`. Surround it with back tick (`` ` ``) in the function definition and escape any special symbols.

Predefined infix operators in R

Predefined infix operators in R

%%	Remainder operator
%/%	Integer division
%*%	Matrix multiplication
%o%	Outer product
%x%	Kronecker product
%in%	Matching operator