

R - Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. R language is rich in built-in operators and provides following types of operators.

Types of Operators

We have the following types of operators in R programming –

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Miscellaneous Operators

Arithmetic Operators

Following table shows the arithmetic operators supported by R language. The operators act on each element of the vector.

| Operator | Description | Example |
|----------|--|--|
| + | Adds two vectors | <div style="text-align: right;">Live Demo</div> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v+t)</pre> <p>it produces the following result –</p> <pre>[1] 10.0 8.5 10.0</pre> |
| - | Subtracts second vector from the first | <div style="text-align: right;">Live Demo</div> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v-t)</pre> <p>it produces the following result –</p> <pre>[1] -6.0 2.5 2.0</pre> |
| * | Multiplies both vectors | <div style="text-align: right;">Live Demo</div> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v*t)</pre> <p>it produces the following result –</p> <pre>[1] 16.0 16.5 24.0</pre> |
| / | Divide the first vector with the second | <div style="text-align: right;">Live Demo</div> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v/t)</pre> <p>When we execute the above code, it produces the following result –</p> <pre>[1] 0.250000 1.833333 1.500000</pre> |
| %% | Give the remainder of the first vector with the second | <div style="text-align: right;">Live Demo</div> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%%t)</pre> |

| | | |
|-----|---|--|
| | | <p>it produces the following result –</p> <pre>[1] 2.0 2.5 2.0</pre> |
| %/% | The result of division of first vector with second (quotient) | <p style="text-align: right;">Live Demo</p> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%/%t)</pre> <p>it produces the following result –</p> <pre>[1] 0 1 1</pre> |
| ^ | The first vector raised to the exponent of second vector | <p style="text-align: right;">Live Demo</p> <pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v^t)</pre> <p>it produces the following result –</p> <pre>[1] 256.000 166.375 1296.000</pre> |

Relational Operators

Following table shows the relational operators supported by R language. Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

| Operator | Description | Example |
|----------|--|---|
| > | Checks if each element of the first vector is greater than the corresponding element of the second vector. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] FALSE TRUE FALSE FALSE</pre> </div> |
| < | Checks if each element of the first vector is less than the corresponding element of the second vector. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v < t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] TRUE FALSE TRUE FALSE</pre> </div> |
| == | Checks if each element of the first vector is equal to the corresponding element of the second vector. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v == t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] FALSE FALSE FALSE TRUE</pre> </div> |
| <= | Checks if each element of the first vector is less than or equal to the corresponding element of the second vector. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v<=t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] TRUE FALSE TRUE TRUE</pre> </div> |
| >= | Checks if each element of the first vector is greater than or equal to the corresponding element of the second vector. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>=t)</pre> </div> |

| | | |
|----|---|--|
| | | <p>it produces the following result –</p> <pre>[1] FALSE TRUE FALSE TRUE</pre> |
| != | <p>Checks if each element of the first vector is unequal to the corresponding element of the second vector.</p> | <p>Live Demo</p> <pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v!=t)</pre> <p>it produces the following result –</p> <pre>[1] TRUE TRUE TRUE FALSE</pre> |

Logical Operators

Following table shows the logical operators supported by R language. It is applicable only to vectors of type logical, numeric or complex. All numbers greater than 1 are considered as logical value TRUE.

Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

| Operator | Description | Example |
|----------|---|---|
| & | It is called Element-wise Logical AND operator. It combines each element of the first vector with the corresponding element of the second vector and gives a output TRUE if both the elements are TRUE. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(3,1,TRUE,2+3i) t <- c(4,1,FALSE,2+3i) print(v&t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] TRUE TRUE FALSE TRUE</pre> </div> |
| | It is called Element-wise Logical OR operator. It combines each element of the first vector with the corresponding element of the second vector and gives a output TRUE if one the elements is TRUE. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(3,0,TRUE,2+2i) t <- c(4,0,FALSE,2+3i) print(v t)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] TRUE FALSE TRUE TRUE</pre> </div> |
| ! | It is called Logical NOT operator. Takes each element of the vector and gives the opposite logical value. | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Live Demo <pre>v <- c(3,0,TRUE,2+2i) print(!v)</pre> </div> <p>it produces the following result –</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <pre>[1] FALSE TRUE FALSE FALSE</pre> </div> |

The logical operator && and || considers only the first element of the vectors and give a vector of single element as output.

| Operator | Description | Example |
|----------|--|---|
| && | Called Logical AND operator. Takes first element of both the vectors and gives the TRUE only if both are TRUE. | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre style="border: 1px solid #ccc; padding: 5px;">v <- c(3,0,TRUE,2+2i) t <- c(1,3,TRUE,2+3i) print(v&& t)</pre> <p>it produces the following result –</p> <pre style="border: 1px solid #ccc; padding: 5px;">[1] TRUE</pre> |
| | Called Logical OR operator. Takes first element of both the vectors and gives the TRUE if one of them is TRUE. | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre style="border: 1px solid #ccc; padding: 5px;">v <- c(0,0,TRUE,2+2i) t <- c(0,3,TRUE,2+3i) print(v t)</pre> <p>it produces the following result –</p> <pre style="border: 1px solid #ccc; padding: 5px;">[1] FALSE</pre> |

Assignment Operators

These operators are used to assign values to vectors.

| Operator | Description | Example |
|----------------------------|-------------------------|---|
| <- or = or <<- | Called Left Assignment | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre> v1 <- c(3,1,TRUE,2+3i) v2 <<- c(3,1,TRUE,2+3i) v3 = c(3,1,TRUE,2+3i) print(v1) print(v2) print(v3) </pre> <p>it produces the following result –</p> <pre> [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i </pre> |
| -> or ->> | Called Right Assignment | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre> c(3,1,TRUE,2+3i) -> v1 c(3,1,TRUE,2+3i) ->> v2 print(v1) print(v2) </pre> <p>it produces the following result –</p> <pre> [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i </pre> |

Miscellaneous Operators

These operators are used to for specific purpose and not general mathematical or logical computation.

| Operator | Description | Example |
|----------|--|--|
| : | Colon operator. It creates the series of numbers in sequence for a vector. | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre>v <- 2:8 print(v)</pre> <p>it produces the following result -</p> <pre>[1] 2 3 4 5 6 7 8</pre> |
| %in% | This operator is used to identify if an element belongs to a vector. | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre>v1 <- 8 v2 <- 12 t <- 1:10 print(v1 %in% t) print(v2 %in% t)</pre> <p>it produces the following result -</p> <pre>[1] TRUE [1] FALSE</pre> |
| %*% | This operator is used to multiply a matrix with its transpose. | <div style="text-align: right; margin-bottom: 5px;">Live Demo</div> <pre>M = matrix(c(2,6,5,1,10,4), nrow = 2, ncol = 3, byrow = TRUE) t = M %*% t(M) print(t)</pre> <p>it produces the following result -</p> <pre> [,1] [,2] [1,] 65 82 [2,] 82 117</pre> |