

## R – Lists

Lists are the R objects which contain elements of different types like – numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using **list()** function.

### Creating a List

Following is an example to create a list containing strings, numbers, vectors and a logical values.

```
# Create a list containing strings, numbers, vectors and a logical
# values.
list_data <- list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)
print(list_data)
```

When we execute the above code, it produces the following result –

```
[[1]]
[1] "Red"

[[2]]
[1] "Green"

[[3]]
[1] 21 32 11

[[4]]
[1] TRUE

[[5]]
[1] 51.23

[[6]]
[1] 119.1
```

### Naming List Elements

The list elements can be given names and they can be accessed using these names.

```
# Create a list containing a vector, a matrix and a list.
list_data <- list(c("Jan", "Feb", "Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),
  list("green", 12.3))

# Give names to the elements in the list.
```

```
names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")
```

```
# Show the list.
```

```
print(list_data)
```

When we execute the above code, it produces the following result –

```
$`1st_Quarter`  
[1] "Jan" "Feb" "Mar"
```

```
$A_Matrix  
  [,1] [,2] [,3]  
[1,]   3   5  -2  
[2,]   9   1   8
```

```
$A_Inner_list  
$A_Inner_list[[1]]  
[1] "green"
```

```
$A_Inner_list[[2]]  
[1] 12.3
```

### Accessing List Elements

Elements of the list can be accessed by the index of the element in the list. In case of named lists it can also be accessed using the names.

We continue to use the list in the above example –

```
# Create a list containing a vector, a matrix and a list.
```

```
list_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),  
  list("green",12.3))
```

```
# Give names to the elements in the list.
```

```
names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")
```

```
# Access the first element of the list.
```

```
print(list_data[1])
```

```
# Access the third element. As it is also a list, all its elements will be printed.
```

```
print(list_data[3])
```

```
# Access the list element using the name of the element.
```

```
print(list_data$A_Matrix)
```

When we execute the above code, it produces the following result –

```
$`1st_Quarter`  
[1] "Jan" "Feb" "Mar"
```

```
$A_Inner_list  
$A_Inner_list[[1]]  
[1] "green"
```

```
$A_Inner_list[[2]]  
[1] 12.3
```

```
      [,1] [,2] [,3]  
[1,]   3   5  -2  
[2,]   9   1   8
```

### Manipulating List Elements

We can add, delete and update list elements as shown below. We can add and delete elements only at the end of a list. But we can update any element.

```
# Create a list containing a vector, a matrix and a list.
```

```
list_data <- list(c("Jan", "Feb", "Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),  
  list("green", 12.3))
```

```
# Give names to the elements in the list.
```

```
names(list_data) <- c("1st Quarter", "A_Matrix", "A Inner list")
```

```
# Add element at the end of the list.
```

```
list_data[4] <- "New element"
```

```
print(list_data[4])
```

```
# Remove the last element.
```

```
list_data[4] <- NULL
```

```
# Print the 4th Element.
```

```
print(list_data[4])
```

```
# Update the 3rd Element.
```

```
list_data[3] <- "updated element"
```

```
print(list_data[3])
```

When we execute the above code, it produces the following result –

```
[[1]]  
[1] "New element"  
  
$<NA>  
NULL  
  
$`A Inner list`  
[1] "updated element"
```

## Merging Lists

You can merge many lists into one list by placing all the lists inside one list() function.

```
# Create two lists.
```

```
list1 <- list(1,2,3)
```

```
list2 <- list("Sun","Mon","Tue")
```

```
# Merge the two lists.
```

```
merged.list <- c(list1,list2)
```

```
# Print the merged list.
```

```
print(merged.list)
```

When we execute the above code, it produces the following result –

```
[[1]]  
[1] 1  
  
[[2]]  
[1] 2  
  
[[3]]  
[1] 3  
  
[[4]]  
[1] "Sun"  
  
[[5]]  
[1] "Mon"  
  
[[6]]  
[1] "Tue"
```

### Converting List to Vector

A list can be converted to a vector so that the elements of the vector can be used for further manipulation. All the arithmetic operations on vectors can be applied after the list is converted into vectors. To do this conversion, we use the **unlist()** function. It takes the list as input and produces a vector.

```
# Create lists.  
  
list1 <- list(1:5)  
print(list1)  
  
list2 <- list(10:14)  
print(list2)  
  
# Convert the lists to vectors.  
  
v1 <- unlist(list1)  
v2 <- unlist(list2)  
  
print(v1)
```

```
print(v2)
```

```
# Now add the vectors
```

```
result <- v1+v2
```

```
print(result)
```

When we execute the above code, it produces the following result –

```
[[1]]  
[1] 1 2 3 4 5
```

```
[[1]]  
[1] 10 11 12 13 14
```

```
[1] 1 2 3 4 5  
[1] 10 11 12 13 14  
[1] 11 13 15 17 19
```