CS-216- APL

L-Sessions

04th Feb, 2021

Sch Id: 1912160

Branch: CSE-B

B.Tech. 4th Sem

× ——————— ×

* List:

- Generic vector that can contain object of different types.
- list() function is used to create data frame.

Example:

```
> list1 <- list(x= c(10,20,30), y= c("a","b","c"),
                z= c(TRUE, FALSE))
> print(list1)
$x
[1] 10 20 30
$y
[1] "a" "b" "c"
$z
[1] TRUE FALSE
> list2 <- list("music tracks", 100, 5)
> print(list2)
[[1]]
[1] "music tracks"
[[2]]
[1] 100
[[3]]
[1] 5
> names(list2) <- c("product", "count", "rating")
> print(list2)
$product
[1] "music tracks"
$count
[1] 100
$rating
[1] 5
```

* Matrix:

> Collection of data elements arranged in 2-D rectangular layout.

> matrix() is used to create a matrix.

Example:

```
> mat <- matrix (1:9, nrow = 3, ncol = 3, byrow = T)
> mat
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

> cbind (1:3, 1:3)
     [,1] [,2]
[1,]    1    1
[2,]    2    2
[3,]    3    3

> rbind (1:3, 1:3)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    2    3

> rbind (mat, 10:12)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12

> cbind (mat, c(4,7,10,13))
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    4    5    6    7
[3,]    7    8    9   10
[4,]   10   11   12   13

> n <-matrix (1:6, byrow = TRUE, nrow =2)
> rownames (n) <- c ("row1" , "row2")
> n
     [,1] [,2] [,3]
row1    1    2    3
row2    4    5    6
```

```
> alphabet ← matrix (LETTERS [1:6], nrow=4, ncol=3)
> alphabet
     [,1] [,2] [,3]
[1,] "A"  "E"  "C"
[2,] "B"  "F"  "D"
[3,] "C"  "A"  "E"
[4,] "D"  "B"  "F"
```

* Data Frame

> Used to store data in form of table.

> data.frame() function is used to create data frame.

Example:

```
> BMI ← data.frame (gender= c(" Male", "Male",
                    "female"), height = c ( 152, 171.5, 165),
                    weight = c (81, 93, 78), Age = c(42,38,26))
> print ( BMI)
  gender height weight Age
1   Male  152.0     81  42
2   Male  171.5     93  38
3 female  165.0     78  26
```

- [illegible]

```
> addfemale ← data.frame (gender= "Female",
                          height= 150, weight= 75, Age= 32)
> rbind (BMI, addfemale)
  gender height weight Age
1   Male  152.0     81  42
2   Male  171.5     93  38
3 female  165.0     78  26
4 female  150.0     75  32
> sort (BMI $Age)
[1] 26 38 42
```

> BMI [order (BMI$Age, decreasing = FALSE),]

	gender	height	weight	Age
1	Female	165.0	81	26
2	Male	171.5	93	38
3	Male	152.0	78	42

* Comparing Vectors Using Relational Operators.

```
> v1 <- c(19, 12, 45)
> v2 <- c(19, 20, 30)
> v1 < v2
[1] FALSE TRUE FALSE
> v1 > v2
[1] FALSE FALSE TRUE
> v1 != v2
[1] FALSE TRUE TRUE
> v1 == v2
[1] TRUE FALSE FALSE
```

* Vector Slicing and Indexing

```
> price <- seq(550, 670, 20
> names(price) <- paste0("p", 1:7)
> price
 p1  p2  p3  p4  p5  p6  p7
550 570 590 610 630 650 670
> price[3]
 p3
590
> price[3:4]
 p3  p4
590 610
```