#기본적인 원소Type

# Vector object [객체]

# 1. logical [TRUE, FLASE]

v1 <- TRUE

print(class(v1))

# 2. Numeric

v2 <- 12.5

print(class(v2))

# 3. Integer

v3 <- 10L

print(class(v3))

# 4. Complex

v4 <- 23 + 3i

print(class(v4))

# 5. Charactor

v5 <- "Hi"

print(class(v5))

# 6. Raw

v6 <- charToRaw("Test")

print(class(v6))

# R-object == Vectors 라고 생각하면 됨.

# < DataType >

# Vectors

# Lists

# Matrics

# Arrays

# Data Frames

# Vectors 설정

apple <- c('red','yellow','green')

print(apple) #character

a <- c(1,2,3,4)

class(a) #numeric

a1 <- "test"

class(a1) #character

a2 <- 2

class(a2) #numeric

a3 <- 2L

class(a3) #integer 숫자 뒤에 L을 붙이면 integer로 선언이 된다

a <- c(1,2,3,4)

class(a) #numeric

#Lists

list1 <-list(c(1,2,3),c(4,5,6),c(7,8,9))

class(list1) #list

list2 <- list(1,2)

class(list2) #list

# List에 Function 을 집어 넣기

list1 <-list(c(1,2,3),c(4,5,6),c(7,8,9), sin)

list1

[[1]]

[1] 1 2 3

[[2]]

[1] 4 5 6

[[3]]

[1] 7 8 9

[[4]]

function (x) .Primitive("sin")

# Matrixs

m <- matrix(c("a","b","c","d","a","b"),

nrow = 2,

ncol = 3)

m # 세로로 순차적으로 입력된다.

# 출력값:

[,1] [,2] [,3]

[1,] "a" "c" "a"

[2,] "b" "d" "b"

# Arrays

arr <- array(c('yellow','red'), dim = c(3,3,1))

arr # 3X3의 Table이 형성

# 출력값 :

, , 1

[,1] [,2] [,3]

[1,] "yellow" "red" "yellow"

[2,] "red" "yellow" "red"

[3,] "yellow" "red" "yellow"

arr2 <- array(c('yellow','red'), dim = c(3,3,2))

arr2 # 3X3의 Table이 형성

# 출력값 :

, , 2

[,1] [,2] [,3]

[1,] "red" "yellow" "red"

[2,] "yellow" "red" "yellow"

[3,] "red" "yellow" "red"

# Factors

apple\_colors <- c('green','yellow','red','red',

'green','green','red')

apple\_colors

#출력 값:

[1] "green" "yellow" "red" "red" "green" "green" "red"

factor\_apple <-factor(apple\_colors)

factor\_apple

#출력 값:

[1] green yellow red red green green red

Levels: green red yellow

print(nlevels(factor\_apple)) #factor(Distinct)된 값의 개수를 출력함

#출력 값:

[1] 3

# DataFrame

test\_data <- data.frame(gender = c("Male",'Female','Male'),

hight = c("170",'180','155'),

weight = c("75",'90','55'),

age = c('40','25','45'))

test\_data

#출력 값: (DataFrame의 형태)

gender hight weight age

1 Male 170 75 40

2 Female 180 90 25

3 Male 155 55 45

# DataTable

#install.packages("data.table")

library(data.table) #Package를 사용하겠다는 신호

test\_table <- data.table(gender = c("Male",'Female','Male'),

hight = c("170",'180','155'),

weight = c("75",'90','55'),

age = c('40','25','45'))

test\_table

#출력 값: (DataTable의 형태)

gender hight weight age

1: Male 170 75 40

2: Female 180 90 25

3: Male 155 55 45

# data.table 을 사용하기 위해서는 Package가 필요하다.

# Strings

data <- "test"

data #(OK)

data2 <- "test"

data2 #(OK) / 같은 값을 변수로 지정 가능함

data3 <- "test' # " " 을 잘 묶어야 함

data3 #(Disable)

#Paste [합치기]

data1 <- "Hello"

data2 <- "good"

data3 <- "Good bye"

data4 <- cat(data1,data2,data3)

data4 #출력 값:

Hello good Good bye

data5 <- paste(data1,data2,data3)

data5 #출력 값:

"Hello good Good bye"

data6 <- paste(data1,data2,data3,sep = "-")

data6 #출력 값: - 으로 나누기

"Hello-good-Good bye"

# format

format(120.12222222,digits = 4) # 소수점을 제외한 나머지 4자리 까지 출력 : 120.1

format(120.2200000, scientific = TRUE) #

[1] "1.2022e+02"

format(14.7,width = 10) # 전체 폭이 10칸 인 상태로 정렬 default 값은 오른쪽

[1] " 14.7"

format("Test",width = 10, justify = "c") # 가운데 정렬 "c"

[1] " Test "

format("Test",width = 10, justify = "r") # 오른쪽 정렬 "r"

[1] " Test"

format("Test",width = 10, justify = "l") # 왼쪽 정렬 "l"

[1] "Test "

# substring

data7 <- "This is mine good morning"

data7

[1] "This is mine good morning"

substring(data7,2,4) # 인덱스를 이용하여 출력 / 출력 값 : his / 인덱스는 1부터임

[1] "his"