#result = result + 100

var1<- 100

var2 <- 3

myResult = 200

myResult <- 300

400 -> myResult

myResult

print(myResult)

print("결과값은 : " myResult) #. 플러스 연산을 문자, 숫자 연결을 위해 사용할 수 없다. R에서는 숫자를 더할 때만 플러스 연산을 사용한다

#, 여러개의 값을 출력하려면 cat()을 사용한다

cat ("결과값은 : " , myResult)

# 출력되는 형식

myResult

# [1] 400 으로 출력된다고 하면, [1]은 출력물의 인덱스임

mySeq = seq(100) # 1부터 100까지 1씩 증가하는 숫자의 집합

# seq : 순서, 순열 (시퀀스)

mySeq

# [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

[18] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34

[35] 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51

[52] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68

[69] 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85

[86] 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

# 맨 앞의 숫자는 출력되는 인덱스의 순서를 이야기함

mySeq = seq(5,100) # 5부터 100까지 1씩 증가하는 숫자의 집합

mySeq = seq(1,100, by = 2)

mySeq

# [1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 [26] 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

###################################

#. 멤버를 이용한 변수 선언

goods.price = 3000

goods.code = "001"

goods.name = "냉장고"

###################################

#Operator (연산자)

var1<- 100

var2 <- 3

var1 / var2 #자바언어 => 33

#R은 마치 실세계의 연산처럼 수행(33.3333)

#몫을 구해보자

var1 %/% var2 # 33

#나머지를 구해보자

var1 %% var2 #1

#비교연산자

var1 == var2

var1 != var2 # TRUE (T)

## 논리연산자

# &, && : 양쪽 값이 TRUE 이면 TRUE

TRUE & FALSE

TRUE && FALSE

# 둘 다 논리연산자 AND 연산

# |,|| : 하나라도 TRUE이면 결과 TRUE

TURE | FALSE #TRUE

# 둘 다 논리연산자 AND 연산

# |,|| : 하나라도 TRUE이면 결과 TRUE

TRUE | FALSE #TRUE

FALSE || FALSE #FALSE

###################################

# 단일변수는 1개짜리 공간

# R은 여러가지 자료구조가 있어요

# 가장 대표적인 자료구조(데이터를 저장하는 구조)에는 vector가 있어요

# R에서 vector는 연속적인 저장공간(1차원, 선형 저장공간)

# 다른 언어의 array와 같은 구조

# vector는 1차원 자료구조

# 저장공간안에 모두 같은 데이터 타입이 들어와요

# 함수를 이용해서 vector를 생성

# c() => combine의 약자

c(10,20,30) # 10 20 30

c(TRUE,20,3.14) # 데이터 타입이 다르지만 vector를 만들 수 있다. 우선순위가 존재함. TURE 값이 1.0으로 바뀜

#벡터 중에 원소가 하나짜리 연산이 스칼라

c(TRUE,FALSE) & c(TRUE,TRUE) #vector연산 결과값 vector

c(TRUE,FALSE) && c(TRUE,TRUE) #맨 앞의 벡터만 연산하고 나머지는 안함

c(TRUE,F,TRUE)& c(FALSE,T) #개수가 안맞아서 warning!

#FALSE FALSE FALSE #벡터의 개수를 맞추자

!c(T,F,T,F)

###################################

# 다른언어와 마찬가지로 다양한

# 수학함수를 내장하고 있어요

# 필요할 때 찾아서 사용해보자

abs(-3)

round(3.1415)

sqrt(100)

###################################

# R의 Data Type

# R에는 Data Type이 크게 2가지가 존재(기본데이터/ 특수데이터)

# 1.데이터 타입

# 1) 숫자형(numeric) : 숫자로 되어있고 정수형,실수형을 의미

100 #실수형

100L #정수형

# 2) 문자열(character) : 하나 혹은 둘 이상의 문자의 집합

"홍길동"

'최길동'

'흥'

# 3) 논리형(logical) : TRUE(T), FALSE(F)

# 4) 복소수형(complex) : 4-3i

# 2.특수 데이터 타입

# 1) NULL : 객체가 존재하지 않음을 지칭하는 객체

var1 = NULL

# 2) NA : Not Available 결측치 표현시 사용(데이터 전처리 과정에서 이상적인 값, 비어있는 값은 모두 NA처리)

# 3) NaN : Not Available Number, Not A Number

sqrt(-3)

# 4) Inf : 양의 무한대

# 5) -Inf : 음의 무한대

###################################

var1 = 100

var2 = 100L

var3 = "Hello"

var4 = TRUE

var5 = 4-3i

var6 = NULL

var7 = sqrt(-3)

# 데이터 타입을 조사하기 위해 제공된

# 함수는 mode()

mode(var1) #numeric

mode(var2) #numeric

mode(var3) #character

mode(var4) #logical

mode(var5) #complex

mode(var6) #NULL

mode(var7) #numeric

#is계열의 함수가 제공됩니다

#is.integer는 정수니? 물어봄!

#is 계열의 함수가 더 있습니다!

#데이터 전처리 작업시 필요해

is.numeric(var1) #TRUE

is.null(var6) #TRUE

is.integer(var1) #FALSE

is.integer(var2) #TRUE

is.numeric(var7) #TRUE

is.nan(var7) #TRUE

#데이터 타입의 우선순위가 있다!!!

#기본데이터 타입 4개에 대하여

# 우선순위가 가장 높은 것은 "character"

# 그 다음은 복소수 "complex"

# 그 다음은 숫자형 "numeric"

# 가장 우선순위가 낮은 것은 "logical"

myVector = c(TRUE,10,30) #numeric

myVector # 1 10 30

myVector = c(TRUE,10,30,"HELLO") #character

myVector # "TRUE" "10" "30" "HELLO"

#하나의 데이터 타입을 다른 데이터 타입으로 바꿀 수 있다(type casting)

#데이터 타입을 변경할 때는 as계열의 함수를 이용하자.

#is는 확인 as는 변경

var1 = 3.14159265358979

var2 = 0

var3 = "3.1415"

var4 = "HELLO"

as.character(var1) #"3.14159265358979"

as.integer(var1) #3

as.logical(var2) #FALSE #0을 제외한 모든 수는 TRUE가 된다

as.double(var3) #3.1415

###################################