

220927 R Essential Review

Jongrak

2022-12-31

```
friends1 <- c("jenny", "tom", "steve", "pheobe", "ross")
"jenny" == friends1

## [1] TRUE FALSE FALSE FALSE FALSE

"rachel" == friends1 # comparison of container and variable > 하나씩 비교

## [1] FALSE FALSE FALSE FALSE FALSE

apple2 <- matrix(c("pheobe", "ross", "monica", 2), ncol = 2) # 다른 타입 >
같은 타입으로 통일
#byrow = TRUE or FALSE > row 방향으로 할지 안 할지

friends2 <- matrix(c("pheobe", "ross", "monica", "joey"), ncol = 2)

# FALSE * FALSE = 0, TRUE + FALSE = 1 이용
sum("rachel" == friends1)

## [1] 0

sum("ross" == friends2)

## [1] 1

"rachel" %in% friends1

## [1] FALSE

"ross" %in% friends1

## [1] TRUE

# The if Statement
# if , if/else, if elif 문 > else if랑 같은 것.
# if elif > elif는 여러 조건을 순서대로 점검. if if if 해도 괜찮 but 여러 개를
만족하면 여러 개 도출.

a <- 330
b <- 200

if (b > a) {
  print("b is greater than a")
} else {
```

```

    print("BOO")
}

## [1] "BOO"

# The if Statement
# if , if/else, if elif 문 > else if랑 같은 것.
# if elif > elif는 여러 조건을 순서대로 점검. if if if 해도 괜찮 but 여러 개를
만족하면 여러 개 도출.

a <- 33
b <- 20

if (b > a) {
  print("b is greater than a")
} else if (b == a) {
  print("a is same as b")
} else {
  b <- b + 1
}

print(a)

## [1] 33

print(b)

## [1] 21

x <- 20
y <- ifelse(x > 30, 5, 3) # 조건, 만족하면, 만족 못하면

y

## [1] 3

h <- 168
j <- 179
r <- 178

if (j > r & j > h) {
  print("j is tallest")
}

## [1] "j is tallest"

if (r > h | r > j) {
  print("r is tall")
}

## [1] "r is tall"

```

```

i <- 1
#조건
while(i < 6) {
  print(i)
  i <- i + 1
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5

i
## [1] 6

i <- 1
#조건
while(TRUE) {
  print(i)
  i <- i + 1
  if (i > 6) {
    break
  }
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6

friends <- c("ross", "monica", "rachel", "joey", "chandler")

for(friend in friends) {
  print(friend)
}

## [1] "ross"
## [1] "monica"
## [1] "rachel"
## [1] "joey"
## [1] "chandler"

# 1:10 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

s <- 0
for(i in 1:10) {
  s <- s + i
}

```

```

}
s

## [1] 55

my_function <- function(x, y) {
  print(2 * x * y)
}

my_function(5, 7)

## [1] 70

# 함수 안에서 지정되는 variable은 밖에서는 저장이 되지 않는다.
my_function <- function(x, y) {
  z = 10 * x * y
  return(2 * x * y)
}

my_function(5, 7)

## [1] 70

s <- my_function(5, 7)

s

## [1] 70

s + 56

## [1] 126

my_function <- function(x, y) {
  return(x * y)
  if (x > y) {
    return (x - y)
  } else if (x == y) {
    return (1000)
  } else {
    return (y - x)
  }
}

# 중간에 return을 만나면 끝.
my_function <- function(x, y) {
  return(x * y)
  if (x > y) {
    return (x - y)
  } else if (x == y) {
    return (1000)
  } else {

```

```

    return (y - x)
  }
}

my_function <- function(z) {
  while(TRUE) {
    z <- z * z
    if ( z > 10000 ) {
      break
    }
  }
  return(z)
}

# break 없이 return만 써주어도 된다.
my_function <- function(z) {
  while(TRUE) {
    z <- z * z
    if ( z > 10000 ) {
      return(z)
    }
  }
}

is.even <- function (n) {
  return(ifelse(n %% 2 == 0, TRUE, FALSE))
}
is.even(10:20)

## [1] TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE

evens.in <- function (x) {
  y <- c()
  while (i < 8)
    if(is.even(x) == "TRUE") {
      y[i] <- x
      i <- i + 1
    } else {
      i <- i + 1
    }
  }
  return(y)
}

consecutive.differences <- function (x) {
  y <- c()
  i = 1
  while (i < length(x)) {
    y[i] <- x[i+1]- x[i]
    i <- i + 1
  }
  return(y)
}

```

```
}  
  
nums = c(1, 6, 7, 9, 11, 15)  
consecutive.differences(nums)  
  
## [1] 5 1 2 2 4  
  
nums[1]  
  
## [1] 1
```