

입력 1: scan() 함수

콘솔(키보드) 혹은 외부 파일을 벡터(숫자형 데이터) 혹은 리스트(문자 데이터)로 저장

예제 데이터

```
eg.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
-0.3 4 5.2
1.3 3 4 9
```

```
eg2.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
하늘 아래 내가 받은 가장 커다란 선물은
오늘입니다
오늘 받은 선물 가운데서도 가장 아름다운 선물은
당신입니다
당신 나지막한 목소리와 웃는 얼굴, 콧노래 한 구절이면
한 아름 바다를 안은 듯한 기쁨이겠습니다.
```

`x<-scan(옵션)`

키보드 입력 내용 혹은 외부 파일을 읽어온 루 오브젝트 `x` 에 벡터(리스트) 형식으로 저장됨

주요 옵션

`scan(file=" ")`

- * “파일이름”: 불러들일 파일 이름을 지정한다. 만약 현재 디렉토리를 안다면 파일 이름만 사용하면 되지만 그렇지 않은 경우 폴더이름까지 사용해야 함
- * 현재 디렉터리 알기 원하면 `getwd()` 함수 사용하면 된다.

```
> getwd()
[1] "Z:/Users/wolfpackkwon/Home_iMac/프로그램_Archive/R"
```

- * “ ”: 지정하지 않은 경우 키보드 입력한 내용을 읽어들이, 물론 `file=` 옵션을 사용하지 않아도 키보드 입력임

`x<-scan()` #numeric input

```
> x<-scan()
1: 2 3 6
4: 3.1 -1
6:
Read 5 items
> x
[1] 2.0 3.0 6.0 3.1 -1.0
```

데이터 구별 할 경우에는 스페이스 바(빈칸)로 빈칸을 입력하거나 엔터 키를 입력하여 다음 라인에 입력한다. 콘솔은 현재 입력되는 데이터 개수를 표시한다. (1:, 4:)

입력이 끝나면 엔터 키를 한 번 더 입력하면 현재까지 입력된 데이터가 저장된다.

scan(what=" ")

- * 읽어 들일 데이터 값 형식 numeric(실수, 디폴트), logical(논리), character(문자열)
- * 데이터가 문자일 경우 반드시 문자임을 지정해야 한다.

```
x3<-scan("eg.txt"); x3
x4<-scan("eg2.txt"); x4
x4<-scan("eg2.txt", what="character"); x4
```

```
> x3<-scan("eg.txt"); x3
Read 7 items
[1] -0.3  4.0  5.2  1.3  3.0  4.0  9.0
> x4<-scan("eg2.txt"); x4
Error in scan("eg2.txt") : scan() expected 'a real', got '하늘'
> x4<-scan("eg2.txt", what="chracter"); x4
Read 30 items
[1] "하늘"           "아래"           "내가"
[4] "받은"           "가장"           "커다란"
[7] "서목은"        "우느입니다"    "우느"
```

```
> x5<-scan(what="chracter")
1: 홍길동 John
3: Rayner
4:
Read 3 items
> x5
[1] "홍길동" "John"  "Rayner"
```

문자 데이터 키보드 입력도 숫자 입력과 동일한 형식을 따른다.

scan(n=정수)

- * 읽어 들일 데이터 최대 개수를 지정, 사용하지 않으면(디폴트) 입력한 만큼 읽어 들인다.

```
x6<-scan("eg2.txt", what="character",n=3);x6
```

```
> x6<-scan("eg2.txt", what="chracter",n=3);x6
Read 3 items
[1] "하늘" "아래" "내가"
```

“eg2.txt” 파일에는 문자 중 첫 번째 단어부터 3개 단어를 불러와 저장한다.

scan(nlines=정수, skip=정수)

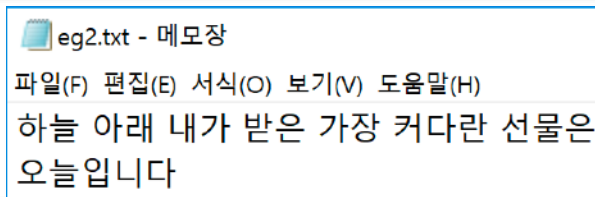
- * nlines : 불러 들일 라인의 개수를 지정할 경우 사용된다.
- * skip : 지정한 라인 개수만큼 건너뛴 후

```
x6a<-scan("eg2.txt", what="character",nlines=2);x6a
```

```
> x6a<-scan("eg2.txt", what="chracter",nlines=2);x6a
Read 8 items
[1] "하늘"      "아래"      "내가"      "받은"
[5] "가장"      "커다란"   "선물은"   "오늘입니다"
```

“eg2.txt”
파일 데이
터에서 첫
2 라인만
불러들여

저장된다.



```
> x6b<-scan("eg2.txt", what="chracter",nlines=2,skip=1);x6b
Read 8 items
[1] "오늘입니다" "오늘"      "받은"      "선물"
[5] "가운데서도" "가장"     "아름다운" "선물은"
```

skip=1 에 의해 첫 라인을 건너 뛰고 두 번째 라인(“오늘입니다”)부터 nlines=2에 의해 2개 라인을 읽어 들인다.

scan(sep="\n")

- * scan() 함수의 경우 다음 데이터 값을 인지하는 방법은 빈 칸 (white space)인데(디폴트) 엔터 키를 칠 때까지 동일 데이터 값으로 인식하게 하는 옵션

```
x7<-scan(sep="\n")
```

```
> x7<-scan(sep="\n")
1: 2 3 4
2:
Read 1 item
> x7
[1] 234
```

빈 칸을 입력해도 동일 데이터로 인지하여 3개 데이터 2, 3, 4 값으로 저장하는 것이 아니라 ‘234’ 하나의 값으로 저장 된다.

scan(quiet=T)

- * scan() 함수에 의해 데이터 입력이 완료되면 읽어 들인 데이터를 화면 출력한다. 디폴트는 False임. True 옵션은 출력하지 않음

입력 2 : readline() 함수

```
readline(prompt="문장")
```

- * “문장” 화면 출력한 후 사용자에게 의해 키보드에 입력내용을 저장한다.

```
Q1<-readline(prompt="당신은 학생입니까? \n")
```

콘솔 화면에 질문 문장이 출력되고 Q1 오브젝트에 “No”가 저장된다

```
> Q1<-readline(prompt="당신은 학생입니까? ")
당신은 학생입니까? No
```

|| 예제 1 || <https://www.rdocumentation.org/>

```
readinteger <- function() {
  n <- readline(prompt="Enter an integer: ")
  n <- as.integer(n)
  if (is.na(n)){n <- readinteger()}
  return(n)
}
print(readinteger())
```

```
> print(readinteger())
Enter an integer: ab
Enter an integer: 12
[1] 12
Warning message:
In readinteger() : NAs introduced by coercion
```

경고 창이 나타남. 이를 해결하는 스크립터는 예제 2임

```
> print(readinteger())
Enter an integer: 32a
Enter an integer: abc
Enter an integer: 12
[1] 12
```

|| 예제 2 || <https://www.rdocumentation.org/>

```
readinteger <- function() {
  n <- readline(prompt="Enter an integer: ")
  if(!grepl("^[0-9]+$",n)){return(readinteger())}
  return(as.integer(n))
}
print(readinteger())
```

참고 : readLines() 함수

```
readLine("파일이름", warn=False, n= 정수)
```

- * 외부 문자 데이터를 읽어오는 함수

```
readLines("eg2.txt")
readLines("eg2.txt",warn=F)
readLines("eg2.txt",warn=F,n=2)
```

```
- -
> readLines("eg2.txt")
[1] "하늘 아래 내가 받은 가장 커다란 선물은"
[2] "오늘입니다"
[3] "오늘 받은 선물 가운데서도 가장 아름다운 선물은"
[4] "당신입니다"
[5] "당신 나지막한 목소리와 웃는 얼굴, 콧노래 한 구절이면"
[6] "한 아름 바다를 안은 듯한 기쁨이겠습니다."
Warning message:
In readLines("eg2.txt") : incomplete final line found on 'eg2.txt'
> readLines("eg2.txt",warn=F)
[1] "하늘 아래 내가 받은 가장 커다란 선물은"
[2] "오늘입니다"
[3] "오늘 받은 선물 가운데서도 가장 아름다운 선물은"
[4] "당신입니다"
[5] "당신 나지막한 목소리와 웃는 얼굴, 콧노래 한 구절이면"
[6] "한 아름 바다를 안은 듯한 기쁨이겠습니다."
> readLines("eg2.txt",warn=F,n=2)
[1] "하늘 아래 내가 받은 가장 커다란 선물은"
[2] "오늘입니다"
```

*warn=F :
읽어들인데
이터 경고 문
출력하지 않
음

*n=2: 처음
2개 라인만
출력됨

```
tmp <- tempfile(fileext = ".txt")
cat("TITLE extra line", "2 3 5 7", "", "11 13 17", file = tmp,
    sep = "\n")
readLines(tmp)
```

```
> cat("TITLE extra line", "2 3 5 7", "", "11 13 17", file = tmp,
+     sep = "\n")
> readLines(tmp)
[1] "TITLE extra line" "2 3 5 7"      ""
[4] "11 13 17"
```

```
> tmp
[1] "C:\\Users\\WOLFPA~1\\AppData\\Local\\Temp\\RtmpUJBdCm\\file16fc2e5e742b.txt"
```

출력 1 : print() 함수

```
print(오브젝트, "문장")
```

- * 설정된 오브젝트의 내용이나 "문장"을 콘솔 출력한다.
- * 숫자나 오브젝트는 그냥 사용해도 되나, 문자열은 " " 을 사용해야 한다.

```
a<-c(15,"HNU")
print(a); print(3)
print(3, 5); print("a", 3)
print(3, "a")
print(readLines("eg2.txt",warn=F,n=2))
```

```
> a<-c(15,"HNU")
> print(a); print(3)
[1] "15" "HNU"
[1] 3
> print(3, 5); print("a", 3)
[1] 3
[1] "a"
> print(3, "a")
Error in print.default(3, "a") : invalid
In addition: Warning message:
In print.default(3, "a") : NAs introduce
> print(readLines("eg2.txt",warn=F,n=2))
[1] "하늘 아래 내가 받은 가장 커다란 선물은"
[2] "오늘입니다"
```

*print() 함수는 제일 앞에 지정된 오브젝트, 문장만 콘솔 출력된다.

*앞과 뒤의 형식이 다른 경우 처음이 문자인 경우에는 첫 문자만 찍히지만 처음이 숫자이고 뒤가 문자인 경우에는 경고와 함께 오류 발생

- 두 개 이상의 오브젝트를 콘솔 출력하는 경우 paste() 함수를 사용하는 것이 적절하다. 이 경우 차수가 많은 오브젝트가 반복된다. 오브젝트 a에는 "15 HNU" 2개 데이터가 있으므로 3이 두 번 앞에 반복된다.

```
> print(paste(3, a),quote=F)
[1] 3 15 3 HNU
```

- * 만약 " " 없이 출력하려고 한다면 quote=False 옵션을 사용하면 된다.

출력 2 : cat() 함수

```
cat("문장내용", 오브젝트, 수식, sep=" ", paste())
```

- * 문장내용을 화면 출력한다. print() 함수와 동일하나 약간의 차이가 있다.
- * sep 옵션은 설정된 오브젝트를 출력할 때마다 빈 칸 없이 이어 출력하는 것 대신 지정한 단어 삽입할 때 사용
- * paste() 차원이 다른 2개의 오브젝트를 출력할 때 큰 차원 오브젝트 반복 시 사용된다.

```
a<-c(15,"HNU")
cat(a); cat(3)
cat(3, 5); cat("a", 3)
cat(3, a)
cat(paste(3, a))
```

```
> a<-c(15,"HNU")
> cat(a); cat(3)
15 HNU3
> cat(3, 5); cat("a", 3)
3 5a 3
> cat(3, a)
3 15 HNU
> cat(paste(3, a))
3 15 3 HNU
```

- * (차이1) 화면 출력이 줄 바꿈없이 이어 된다.
- * (차이2) “ “ 출력이 없음
- * (차이3) 문자, 숫자 구별없이 입력형식 지정에 따라 된다.
- * paste() : 오브젝트 a에는 “15 HNU” 2개 데이터가 있으므로 하나인 숫자 3은 두 번 반복 출력된다.
- * (결론) 화면 출력을 위하여 cat() 문장을 권한다.

```
cat(paste(3,"*",5,"=", 3*5))
cat(3,"students","\n","are working now",sep="^;")
cat(paste("c",c(1:3),sep="="))
```

```
> cat(paste(3,"*",5,"=", 3*5))
3 * 5 = 15
> cat(3,"students","\n","are working now",sep="^;")
3^;students^;
^;are working now
> cat(paste("c",c(1:3),sep="="))
c=1 c=2 c=3
```