

Chapter 4. 다양한 INPUT 명령문의 형식

원자료의 위치 지정

❖ INFILE 명령문

- 데이터를 외부 파일로부터 읽는 경우에 외부 파일의 경로와 파일 이름을 지정하는 명령문
- INPUT 명령문
 - 변수의 이름 및 유형을 지정하기 위해 사용
- INFILE 명령문은 항상 INPUT 명령문 앞에 위치
- 경로를 포함한 파일 이름을 인용부호로 닫아주어야 함

외부파일로 작성된 원자료 읽기	성적.txt
<pre>DATA mysas.person; INFILE "c:\SAS_Programming\Raw_Data\성적.txt"; INPUT name \$ dept \$ salary age; RUN;</pre>	<pre>김철수 stat 96 38 최민지 mate 89 42 이영희 econ 98 55 오인수 stat 102 39</pre>

원자료의 위치 지정

❖ CARDS 명령문

- 원자료가 프로그램 내부에 입력되어 있음을 지정하는 명령문
- CARDS 뒤에 위치하는 모든 자료값들은 ;(semicolon)이 나타날 때까지 원자료로 인식

외부파일로 작성된 원자료 읽기

```
DATA mysas.person;  
INPUT name $ dept $ salary age;  
CARDS;  
김철수 stat 96 38  
최민지 mate 89 42  
이영희 econ 98 55  
오인수 stat 102 39  
;  
RUN;
```

공백문자로 구분된 데이터 읽기

❖ List Input

- 자유포맷
 - 자료값이 하나 이상의 공백문자에 의해서 분리되어 있는 경우
 - 문자를 자료값으로 가지는 문자변수 위에 특수기호 \$를 표시
- 자유포맷에 의한 입력 규칙
 - 데이터의 모든 자료값들은 적어도 하나 이상의 공백문자에 의해 분리
 - 결측값은 한 개의 마침표(.)로 표시
 - 문자변수의 자료값은 공백문자를 포함하지 않아야 하고, 길이가 8자를 넘지 않아야 함
 - 8자리를 넘게 되면 앞에서부터 8자리만을 데이터 셋에 저장

자유포맷으로 데이터 읽기

```
DATA mysas.grade;  
INPUT id name $ class $ score grade $;  
CARDS;  
2590 john Stat 80.3 A  
    3569 willy Math . B  
7048 Younghee . 30.6 C  
9087 . Eng 100    A  
. e.t . . .  
;  
RUN;
```

고정포맷으로 입력된 데이터 읽기

❖ Column Input

- 고정포맷
 - 모든 데이터의 자료값들이 열에 맞추어 정리되어 있는 경우
 - 입력포맷 또는 열지시자에 의해서 데이터를 읽어들이 수 있음
 - 열지시자: 고정포맷으로 입력된 각 변수의 시작 열과 끝 열을 지정하는 것
 - 고정포맷을 사용하는 것에 대한 이점
 - 자료값들 사이에 공백문자를 입력하지 않아도 됨
 - 문자변수가 공백문자를 포함할 수 있음
 - 결측값을 공백문자로 처리할 수 있음
 - 필요 없는 변수에 대한 자료값을 읽지 않을 수 있음
 - 읽을 열의 위치가 지정되므로 어떤 순서로 읽어도 무방함
 - 같은 열을 반복하여 읽을 수도 있음

고정포맷으로 데이터 읽기

```
DATA mysas.grade1;  
INPUT name $ 11-18 class $ 20-35 classid 1-4 score 36-40 grade $ 41;  
CARDS;  
259067048 김철수 Statistics      80.3 A  
356989087 박민호 Mathematics    B  
704842534 이영희                30.6 C  
;  
RUN;
```

특수 포맷으로 입력된 데이터 읽기

❖ Formatted Input

- 표준 데이터
 - 영문자 또는 한글문자, 숫자, - 부호, 소수점 등으로만 입력되어 있는 데이터
- 비표준 데이터
 - 표준 데이터 외의 다른 문자와 함께 입력되어 있는 데이터
 - 날짜자료: 가장 흔한 비표준 데이터
 - 1960년 1월 1일을 기준으로 그때부터의 날수로 바꾸어서 저장
 - 입력포맷(informat): 특별한 형식으로 입력된 비표준 데이터를 읽을 때 유용

입력포맷 사용형식

INPUT variable <\$> <informat>w.d

- w: 전체 자릿수, d: 소수점 이하의 자릿수(d가 생략되면 0으로 인식)

특수 포맷으로 입력된 데이터 읽기

■ 입력포맷의 사용 예

입력포맷의 사용 예					
유형	입력포맷	내용	입력자료	사용 예	결과
문자형	\$w.	문자 이전의 공백문자 제거	Min Ho Min Ho	\$8. \$8.	Min Ho Min Ho
	\$CHARw.	문자이전의 공백문자까지 포함	Min Ho Min Ho	\$CHAR8. \$CHAR8.	Min Ho Min Ho
숫자형	w.d	표준 숫자자료를 읽음	1234	5.1	123.4
	COMMAw.d	콤마와 \$부호가 포함된 숫자자료를 읽음. 괄호는 -부호로 인식	\$1,100,100 (1,234)	COMMA10. COMMA7.	1100100 -1234
	PERCENTw.	%부호가 포함된 숫자자료를 읽음. 괄호는 -부호로 인식	15% (20%)	PERCENT3. PERCENT5.	0.15 -0.20
날짜형	DATEw.	ddmmmyy 또는 ddmmmyyyy형태로 입력된 날짜자료를 읽음	01jan1961 01jan61	DATA9. DATA7.	366 366

특수 포맷으로 입력된 데이터 읽기

❖ 포인터 조절

- +n
 - 현재 읽고 있는 포인터 위치를 n칸 만큼 오른쪽으로 이동하도록 지시
 - 숫자 n은 양의 정수가 되어야 함
 - 포인터를 왼쪽으로 이동시키는 경우

```
m=-5;  
INPUT x1 10. +m x2 5.;
```

❖ 출력포맷

- FORMAT 명령문
 - 사용자가 이해하기 편리한 형태로 다시 출력하기 위해 사용

입력포맷 및 출력포맷의 사용

```
DATA mysas.dept;  
INPUT name $18. +1 hired DATE7. +1 salary 5.;  
FORMAT hired DATE7. ;  
CARDS;  
Martin, Virginia 09aug80 34800  
Leighton, Maurice 16dec83 32600  
;  
RUN;
```


특수 포맷으로 입력된 데이터 읽기

❖ 다양한 입력포맷의 사용

- Mixed Input
 - 여러 가지 INPUT 명령문의 형식을 함께 사용
 - 다양한 입력포맷을 사용하면 특별하게 입력된 데이터를 쉽게 읽을 수 있음
 - 보다 효율적으로 프로그램을 작성할 수 있음

다양한 입력포맷의 사용

```
DATA mysas.club;  
INPUT idno name $ 6-19 @25 team $6. strtwght endwght;  
CARDS;  
1023 David Shaw    red   189 165  
1049 Amelia Serrano yellow 189 165  
;  
RUN;
```

- @n : 입력포인터의 위치를 n번째열로 이동시켜 자료값을 읽도록 지정하는 열지시자

특수 포맷으로 입력된 데이터 읽기

❖ 변수이름의 생략

- 데이터 단계나 프로시저 단계에서 다양한 형태로 변수 이름을 생략하여 사용할 수 있음
- x1-x10: x1부터 x10까지의 10개 변수
- math—stat: math 부터 stat 사이에 저장되어 있는 모든 변수
- math_NUMERIC_stat: math부터 stat 사이의 모든 숫자변수
- math_CHARACTER_stat: math부터 stat 사이의 모든 문자변수
- _NUMERIC_: 데이터 셋에 있는 모든 숫자변수
- _CHARACTER_: 데이터 셋에 있는 모든 문자변수
- _ALL_: 데이터 셋에 있는 모든 변수
 - Ex.
 - (jan feb mar) (+5 COMMA5.): jan +5 COMMA5. feb +5 COMMA5. mar +5 COMMA5.
 - (score1-score3) (4.1): score1 4.1 score2 4.1 score3 4.1

하나의 개체가 여러 줄에 입력되어 있는 데이터 읽기

- 입력해야 할 변수가 매우 많거나 특별한 필요에 의해서 하나의 관찰개체의 자료가 여러 줄에 걸쳐서 입력되는 경우

❖ /(slash)

- 포인터의 위치를 다음 줄의 첫 열로 이동

❖ #n(pound-n)

- 포인터의 위치를 데이터의 매 n 번째 행의 첫 열로 이동시키도록 지정

입력포인터의 행 이동

```
DATA mysas.club1;  
INPUT idno name $ 6-19 / team $6. #3 strtwght endwght;  
CARDS;  
1023 David Shaw  
red  
189 165  
1049 Amelia Serrano  
yellow  
189 165  
;  
RUN;
```

여러 개의 개체가 한 줄에 연속해서 입력되어 있을 때

❖ @@(double trailing at)

- 모든 자료의 값을 전부 읽을 때까지 다음 줄로 포인터가 이동하지 않도록 지정

@@의 사용

```
DATA mysas.double;  
INPUT x y @@;  
CARDS;  
1 2 3 4  
5 6 7  
8 9 10  
;  
RUN;
```

- @@을 사용한 경우와 사용하지 않은 경우 비교

자료값에 따라 서로 다른 INPUT 명령문 사용하기

❖ @(trailing at)

- 끝나지 않는 INPUT 명령문이 나타날 때까지 다음 줄로 포인터를 이동하지 않고 기다리도록 지정
- 즉, @이 사용되지 않으면 SAS는 INPUT 명령문에 나열된 마지막 변수의 자료값을 읽은 후에 자동적으로 포인터의 위치를 다음줄로 이동

@의 사용

```
DATA mysas.year;  
INPUT year 1-4 @;  
IF year=1988 THEN INPUT dat 5-7 amount 8-10;  
IF year=1989 THEN INPUT dat 6-8 amount 10-12;  
CARDS;  
19883.2149  
19885.7614  
1989 7.9 764  
1989 6.8 875  
;  
RUN;
```

기본적인 INFILE 옵션

❖ FIRSTOBS=n 옵션

- n번째 자료부터 자료를 읽기 시작하도록 지정하는 옵션
- 자료값의 시작부분에 데이터에 대한 기술이나 불필요한 정보가 포함되어 있을 때, 이 부분을 읽지 않게 할 경우에 유용

FIRSTOBS 옵션의 사용

Score of Midterm-Exam, 2000

ID	NAME	SUBJECT	SCORE
259067048	Chulsu Kim	Statistics	80.3
356989087	Minho Park	Mathmatics	100
704842534	Younghee Lee	English	30.6

```
DATA mysas.midsocre;
```

```
INFILE "c:\SAS_Programming\Raw_Data\중간시험.txt" FIRSTOBS=3;
```

```
INPUT id 1-9 name $ 11-22 class$ 24-33 score 34-38;
```

```
RUN;
```

기본적인 INFILE 옵션

❖ MISSOVER 옵션

- INPUT 명령문에 나열된 변수의 개수보다 어떤 개체의 자료값의 개수가 적게 입력되어 있는 경우
- 자료값을 읽지 못한 변수에 대해서는 결측값으로 처리하도록 지정
- INFILE 명령문에 있는 옵션을 사용하기 위해 INFILE CARDS 명령문을 사용

MISSOVER의 사용

```
DATA mysas.miss;  
INFILE CARDS MISSOVER;  
INPUT a b c;  
CARDS;  
1 2 3  
4 5  
6 7 8  
9 0 1 2  
3 4 5  
;  
RUN;
```

기본적인 INFILE 옵션

❖ DLM=, DSD, EXPANDTABS 옵션

- DLM= 옵션
 - 사용자의 환경에 따라서 공백문자가 아닌 다른 구분자를 사용할 경우
 - DLM=옵션을 사용하여 하나 이상의 문자 또는 특수기호 등을 자료값의 구분자로 지정할 수 있음
- DSD 옵션
 - DLM 옵션과 유사하면서 보다 확장된 기능을 가지고 있음
 - 콤마를 구분자로 사용
 - 연속된 구분자들 사이를 결측값으로 처리
 - 인용부호 안에 있는 문자는 구분자로 인식하지 않음
 - 인용부호는 자료값으로 받아들이지 않음
- EXPANDTABS 옵션
 - 데이터의 구분자가 탭 문자로 되어 있는 파일을 읽는 경우

MISSOVER의 사용

```
DATA mysas.score;  
INFILE CARDS DSD;  
INPUT test1 test2 test3;  
CARDS;  
91,87,95  
97,,92  
;  
RUN;
```


연습문제

- ❖ 연습문제4_1.txt파일을 이용하여 SAS 데이터 셋을 생성하시오. (단, 변수 이름은 id, dept, age, test1, test2, gender로 할 것)
- ❖ 연습문제4_2.txt파일을 이용하여 SAS 데이터 셋을 생성하시오. (단, 변수이름은 id, name, item1, item2, ..., item6(item1 ~ item6까지 모두 1자리), height(3자리), weight(2자리), age(2자리), region 순으로 할 것)
- ❖ 연습문제4_3.txt파일을 이용하여 SAS 데이터 셋을 생성하시오.(단, 변수 이름은 id, dept, age, test1, test2, gender로 할 것)
- ❖ 연습문제4_4.txt파일을 이용하여 SAS 데이터셋을 생성하시오. (단, 변수 이름은 last, first, gender, math, engl로 할 것)