

3-2강

DATA 읽기 : 외부 파일로부터

```
DATA company;  
INFILE "E:\data\기업이미지.txt";  
INPUT id 1-2 age 3 gender $ 4 item1 5 item2 6 item3 7;  
LABEL id='고객번호' age='나이' gender='성별'  
      item1='좋은 제품만들기 위해 노력한다'  
      item2='소비자를중요하게여긴다'  
      item3='신뢰할만한 기업이다'  
;  
  
RUN;  
PROC PRINTDATA=company LABEL; /*LABEL option을 넣어줘야 label이 적혀나온다.*/  
RUN;
```

파일:기업이미지.txt

```
1 1M111  
2 2M333  
3 4F331  
4 4M332  
5 4M111  
6 5F299  
7 3M111  
8 5F111  
9 5M113  
102F232
```

OBS	고객번호	나이	성별	좋은 제품을 만들기 위해 노력한다	소비자를 중요하게 여긴다	신뢰할만한 기업이다
1	1	1	M	1	1	1
2	2	2	M	3	3	3
3	3	4	F	3	3	1
4	4	4	M	3	3	2
5	5	4	M	1	1	1
6	6	5	F	2	9	9
7	7	3	M	1	1	1
8	8	5	F	1	1	1
9	9	5	M	1	1	3
10	10	2	F	2	3	2

DATA 읽기

- RAW Data File 읽기
 - DATA 문장
 - Data Step 시작 문장
 - DATA 키워드 옆에 생성될 SAS Data Set 이름을 적음
 - INFILE 문장

- 읽어올 외부 파일을 지정하는 문장
- INFILE 키워드 옆의 읽어 올 외부 파일의 경로 및 파일명 따옴표 안에 적음
예) infile "F:\data\sales.csv";
- firstobs : 자료를 불러들이기 시작하는 obs를 지정 (infile option)
→ 맨 첫줄에는 obs가 아닌 변수명이 있을 수도 있기에 잘 확인하라
- expandtabs : 자료의 사이가 tab으로 띄어져 있는 경우 사용 (infile option)

```
DATA output-SAS-data-set;
INFILE 'raw-data-file-name' firstobs=2 expandtabs;
INPUT specification;
RUN;
```

- INPUT 문장
 - Raw Data File을 어떻게 읽어올 것인가를 지정
 - Raw Data File의 데이터 값을 어떻게 읽어서, 어떤 변수에 저장할 것인지를 지정함
 - INPUT 문장 작성 방법에 따라 Column input, Formatted input, List input 등의 방법이 있음

Input 방식	방식 결정 요소		구문
	파일형식	비표준 데이터 처리	
Column	고정너비	처리 불가능	Input 변수명 <\$>시작위치-끝위치 ... ;
Formatted		처리 가능	Input <@start> 변수명 입력 형식 ... ;
List	구분자로 구분	처리 불가능	Input 변수명 <\$> ... ;
		처리 가능	Input 변수명 입력형식 ... ;

입력형식

구분	INFORMAT	내용	데이터값	입력포맷	저장
숫자	w.	w 자릿수의 정수로 표현	123	5.	123
	w.d	정수부분 w + 소수부분 d	123	5.1	12.3
	COMMAw.d	coma와 \$포함. ()는 음수	(\$1,100)	COMMA10.	-1100
	PERCENTw.d	%부호 포함. ()는 음수	(20%)	PERCENT5.	-0.20
문자	\$w.	문자이전 공백 삭제	__Min Ho	\$8.	Min Ho
	\$CHARw.	문자이전 공백 포함	__Min Ho	\$CHAR8.	__Min Ho
날짜	MMDDYYw.	MM-DD-YY의 형태	01-01-1961	MMDDYY10.	366
	TIMEw.	HH:MM:SS의 형태	10:30	TIME5.	37800
	DATEw.	DD-MON-YY	01JAN1961	DATE9.	366

- YYMMDD8. : 67-08-13 YYMMDD10. : 1967-08-13
- MON : JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
→ DATEw 형식은 꼭 기억해둬라.
- SAS의 날짜/시간 기준 : 1960년 1월 1일(=0) 00:00:01(=1)

원시 DATA 형태

	고정 (Fixed-format)	자유(free-format)
표준 데이터 유형 문자, 숫자(포함)	<p>Raw Data File Exercise</p> <pre> 1-----10-----20 2810 61 MOD F 2804 38 HIGH F 2807 42 LOW M 2816 26 HIGH M 2833 32 MOD F 2823 29 HIGH M </pre>	<pre> 1-----10-----20 BARNES NORTH 360.98 FARLSON WEST 243.94 LAWRENCE NORTH 195.04 NELSON EAST 169.30 STEWART SOUTH 238.45 TAYLOR WEST 318.87 </pre>
비표준 데이터 유형	<p>Raw Data File Staff</p> <pre> 1-----10-----20----- EVANS DONNY 112 29,996.63 HELMS LISA 105 18,567.23 HIGGINS JOHN 111 25,309.00 LARSON AMY 113 32,696.78 MOORE MARY 112 28,945.89 </pre>	<pre> 1 1 Male 1,1,16% 2 2 Man 3,1,39\$ 3 4 Female 3,3,19% 4 4 Man 3,3,24% 5 4 M 1,1,101% 6 5 Female 2,... 7 3 MR 1,1,1105% 8 5 Female 1,1,130% 9 5 Man 1,1,32% 10 2 Female 2,3,26% </pre>

고정(Fixed-format) : DATA들이 column에 고정이 되어있다는 의미이다.

자유(free-format) : DATA들이 column에 고정이 되어있지 않다는 의미이다.

표준 DATA 유형 : 문자, 숫자, 마침표(.) 만이 포함되었다는 의미이다.

비표준 DATA 유형 : 문자, 숫자, 마침표(.)이외의 다른 것도 포함되었다는 의미이다.

고정 포맷 & 표준데이터 유형 (COLUMN INPUT)

데이터 유형	011M111 022M333 034F331 044M332 054M111 065F2.. 073M111 085F111 095M113 102F232	01,1,M,1,1,1 02,2,M,3,3,3 03,4,F,3,3,1 04,4,M,3,3,2 05,4,M,1,1,1 06,5,F,2,... 07,3,M,1,1,1 08,5,F,1,1,1 09,5,M,1,1,3 10,2,F,2,3,2	01 1 Male 1 1 1 02 2 Man 3 3 3 03 4 Female 3 3 1 04 4 Man 3 3 2 05 4 M 1 1 1 06 5 Female 2 . . 07 3 MR 1 1 1 08 5 Famme 1 1 1 09 5 Man 1 1 3 10 2 Female 2 3 2
특징	각 변수의 값을 읽는 위치가 모든 레코드에서 동일함		
문법	입력방법 : 변수명 변수유형 시작위치-끝위치 ex) age 1-2 , gen \$ 6-18, gen \$ 3-3 = gen \$ 3		

비표준 데이터 유형 (Formatted INPUT)

데이터 유형	고정 (Fixed-format)	자유(free-format)
	<pre> 1-----10-----20----- EVANS DONNY 112 29,996.63 HELMS LISA 105 18,567.23 HIGGINS JOHN 111 25,309.00 LARSON AMY 113 32,696.78 MOORE MARY 112 28,945.89 </pre>	<pre> 1 1 Male 1,1,16% 2 2 Man 3,1,39\$ 3 4 Female 3,3,19% 4 4 Man 3,3,24% 5 4 M 1,1,101% </pre>
특징	<p>각 레코드의 변수별 데이터값 시작위치가 동일</p> <p>사전에 정의된 유형</p> <ul style="list-style-type: none"> - 00,000,000 : 천단위 숫자 구분 - ddMONyy : 날짜 표시 	<p>변수LIST 순서에 따라 구분자로 분리됨</p> <p>사전에 정의된 유형</p> <ul style="list-style-type: none"> - 00,000,000 : 천단위 숫자 구분 - ddMONyy : 날짜 표시
문법	<p>@입력시작위치 변수명 입력포맷 ex) @1 id 2. , @6 gen \$12.</p>	<p>변수명 입력끝 포인터 입력포맷 ex) id & 2. , gen : \$12.</p>

이 경우에는 &, 이나 :을 이용한다.

포맷수정자 (& 와 :)

```
INPUT 변수명 & ($자릿수.);
```

- 공백을 포함한 문자열을 읽음
- &(ampersand)는 2칸 이상의 공백 다음에 오는 값 전까지 인식하는 것으로
& 다음에 자릿수를 지정하면 공백까지 값을 읽은 후에 자릿수만큼 유효 자리로 인정한다.
- 관찰값 보다 자릿수가 작은 경우에는 해당 자릿수만큼 관찰값으로 읽는다.

```
INPUT 변수명 : ($자릿수.);
```

- 포맷 문자형 자료를 읽을 때 지정 길이에 관계없이 처음 공백이 나올 때까지 읽는다.
- :(colon)은 해당 변수를 쓰고 그 다음에 :을 한 후에 자릿수를 지정하면 공백까지 관찰값을 읽은 후에 자릿수만큼 유효자리로 인정한다.
즉, 관찰값의 길이를 지정하는데 사용한다.

- 중간에 공백을 포함하지 않은 비표준데이터 값과 길이 8이상의 문자값을 읽는다.

→ 시험에는 잘 안낸다.

포맷수정자 : &

```
DATA STL;
INPUT SEASON $ WIN_LOSE $ WINNING_RATE $ DISTRICT $ PO $ BEST_bWAR&$16.;
CARDS:
2011 90-72 0.556 2위 우승 푸홀스(5.3)
2012 88-74 0.543 2위 NLCS 몰리나(6.9)
2011 97-65 0.599 1위 WS 웨인라이트(6.4)
2015 100-62 0.617 1위 NLDS 헤이워드(6.5)
RUN;

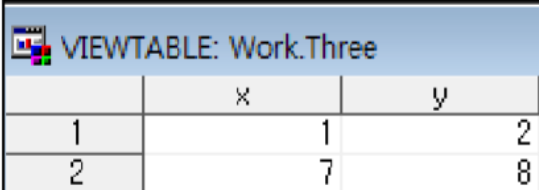
PROC PRINT;
RUN;
```

OBS	SEASON	WIN_LOSE	WINNING_RATE	DISTRICT	PO	BEST_bWAR
1	2011	90-72	0.556	2위	우승	푸홀스 (5.3)
2	2012	88-74	0.543	2위	NLCS	몰리나 (6.9)
3	2011	97-65	0.599	1위	WS	웨인라이트 (6.4)
4	2014	90-72	0.556	1위	NLCS	웨인라이트 (6.4)
5	2015	100-62	0.617	1위	NLDS	헤이워드 (6.5)

DATA 읽기 : @@

- Input 문에서 cards 의 값을 연속적으로 읽을 수 있게 해주는 옵션
 - Input 변수만큼 읽으면 더 이상 읽지 않는다.
- @@를 이용하면 한 줄을 다 읽는다.

```
data three;
input x y;
card;
1 2 3 4 5 6
7 8 9 7 2 6
;
run;
```



	x	y
1	1	2
2	7	8

```

data three;
  input x y @ @;
  card;
1 2 3 4 5 6
7 8 9 7 2 6
;
run;

```

VIEWTABLE: Work.Four		
	x	y
1	1	2
2	3	4
3	5	6
4	7	8
5	9	7
6	2	6