

The INPUT Statement: Where It's @

Ron Cody

email: ron.cody@gmail.com

Author page: Support.sas.com/cody

List Directed Input

```
data list;  
    input X Y A $ Z;  
datalines;  
1 2 hello 3  
4      5      goodbye 6  
;  
title 'List Directed Input';  
proc print data=list;  
run;
```

List Directed Input				
Obs	X	Y	A	Z
1	1	2	hello	3
2	4	5	goodbye	6

Other Delimiters

```
data delim;  
    infile datalines dlm='#';  
    input X Y A $ Z;  
datalines;  
1#2#hello#3  
4 # 5 # goodbye # 6  
;  
title 'Other Delimiters';  
proc print data=delim;  
run;
```

Other Delimiters				
Obs	X	Y	A	Z
1	1	2	hello	3
2	4	5	goodbye	6

Reading CSV Files

```
data special;  
    infile datalines dsd;  
    input X Y A $ Z;  
datalines;  
1,2,hello,3  
4 , 5 , goodbye , 6  
7,, "hi there",8  
9,10, "hi,there",11  
;  
title 'Special Comma Delimited Format';  
proc print data=special;  
run;
```

Reading CSV Files

1,2,hello,3

4 , 5 , goodbye , 6

7,, "hi there",8

9,10, "hi,there" ,11

Special Comma Delimited Format

Obs	X	Y	A	Z
1	1	2	hello	3
2	4	5	goodbye	6
3	7	.	hi there	8
4	9	10	hi,there	11

Combining DSD with Other Delimiters

```
data special;  
  infile datalines dsd dlm='09'x;  
  *Note: A hex value of 09 is an ascii tab;  
  input X Y A $ Z;  
datalines;  
1      2      hello      3  
4      5      goodbye    6  
7      "hi there"      8  
9      10     "hi,there"    11  
;
```

Tab Delimited Data and DSD				
Obs	X	Y	A	Z
1	1	2	hello	3
2	4	5	goodbye	6
3	7	.	hi there	8
4	9	10	hi,there	11

Column Input

```
data col1;  
  input X      1-2  
        Y      3  
        A $ 4-10  
        Z      11;  
datalines;  
  12hello 3  
4 5goodbye6  
;  
title 'Column Input';  
proc print data=col1;  
run;
```

Column Input				
Obs	X	Y	A	Z
1	1	2	hello	3
2	4	5	goodbye	6

Column Input (selected variables)

```
data col2;  
    input X 1-2  
          Z 11;  
datalines;  
  12hello 3  
4 5goodbye6  
;  
title 'Column Input';  
proc print data=col2;  
run;
```

Column Input		
Obs	X	Z
1	1	3
2	4	6

Column Input (different order)

```
data col3;  
  input Y          3  
        A $ 4-10  
        Z          11  
        X          1-2;  
  
datalines;  
  12hello 3  
  4 5goodbye6  
;  
title 'Column Input';  
proc print data=col3;  
run;
```

Column Input				
Obs	Y	A	Z	X
1	2	hello	3	1
2	5	goodbye	6	4

Formatted Input

```
data inform1;  
    input @1 X      2.  
          @3 Y      1.  
          @4 A      $7.  
          @11 Z     1.  
          @12 Date  mmddyy10.;  
    format Date date9.;  
datalines;  
    12hello  310/21/1946  
4 5goodbye611/12/1997  
;  
title 'Pointers and Informats';  
proc print data=inform1;  
run;
```

Pointers and Informats						
Obs	X	Y	A	Z	Date	
1	1	2	hello	3	21OCT1946	
2	4	5	goodbye	6	12NOV1997	

Using Informats with List Input (Colon Modifier)

```
data colon;
  input X :      2.
        Y :      1.
        A :     $11.
        Z :      1.
        Date : mddy10.;
  format Date date9.;
datalines;
1 2 hello 3 10/21/1946
4 5 arrivederci 6 11/12/1997
;
title 'Informats: Colon Modifier';
proc print data=colon;
run;
```

Using Informats with List Input (Informat Statement)

```
data inform2;  
    informat X 2. Y Z 1. A $11.  
            Date mmddy10.;  
    input X Y A Z Date;  
    format Date date9.;  
datalines;  
1 2 hello 3 10/21/1946  
4 5 arrivederci 6 11/12/1997  
;  
title 'Informat Statement';  
proc print data=inform2;  
run;
```

Using Informats with List Input (Ampersand Modifier)

```
data amper;  
  input X : 2.  
    Y : 1.  
    A & $11.  
    Z : 1.;  
datalines;  
1 2 hello there 3  
4 5 a bientot 6  
;  
title 'Ampersand Modifier';  
proc print data=amper;  
run;
```

Ampersand Modifier

Obs	X	Y	A	Z
1	1	2	hello there	3
2	4	5	a bientot	6

Without Variable and Informat Lists

```
data noinlist;  
  input @1 Q1 1.  
        @2 Q2 1.  
        @3 Q3 1.  
        @4 Q4 1.  
        @5 Q5 1.  
        @6 A $1.  
        @7 B $1.  
        @8 C $1.;  
  
datalines;  
12345xyz  
;  
title 'Without Variable and Informat Lists';  
proc print data=noinlist;  
run;
```

Without Variable and Informat Lists								
Obs	Q1	Q2	Q3	Q4	Q5	A	B	C
1	1	2	3	4	5	x	y	z

Variable and Informat Lists (1)

```
data inlist1;  
    input @1 (Q1-Q5 A B C) (5*1. 3*$1.);  
datalines;  
12345xy  
;  
title 'Variable Lists (1)';  
proc print data=inlist1;  
run;
```

Variable Lists (1)

Obs	Q1	Q2	Q3	Q4	Q5	A	B	C
1	1	2	3	4	5	x	y	z

Variable and Informat Lists (2)

```
data inlist2;  
    input @1 (Q1-Q5) (1.)  
          @6 (A B C) ($1.);  
datalines;  
12345xyz  
;  
title 'Variable Lists (2)';  
proc print data=inlist2;  
run;
```

Variable Lists (2)									
Obs	Q1	Q2	Q3	Q4	Q5	A	B	C	
1	1	2	3	4	5	x	y	z	

Reading X,Y Pairs

```
data pairs;  
    input @1(X1 Y1 X2 Y2 X3 Y3)(1.);  
datalines;  
123456  
;  
title 'Reading (X,Y) Pairs';  
proc print data=pairs;  
run;
```

Reading X,Y Pairs						
Obs	X1	Y1	X2	Y2	X3	Y3
1	1	2	3	4	5	6

Relative Column Pointers

```
data relative;  
    input @1 (X1-X3) (1. + 1)  
          @2 (Y1-Y3) (1. + 1);  
datalines;  
123456  
;  
title 'Relative Column Pointers';  
proc print data=relative;  
run;
```

Relative Column Pointers						
Obs	X1	X2	X3	Y1	Y2	Y3
1	1	3	5	2	4	6

Text Pointer

```
data test;  
    input @ "xyz" Value;  
datalines  
this line has xyz 76 numbers  
none on this line  
xyz 20  
;  
title 'Listing Of Test';  
proc print data=test;  
run;
```

Listing of Test

Obs	Value
-----	-------

1	76
---	----

2	20
---	----

Named Input

```
data new;  
    input ID= X= Y= Z=;  
datalines;  
ID=123 X=1 Y=2 Z=3  
ID=234 Z=9 Y=7  
ID=999 Z=8  
;  
title 'Listing of Data Set New';  
proc print data=new noobs;  
run;
```

Listing of Data Set New			
ID	X	Y	Z
123	1	2	3
234	.	7	9
999	.	.	8

Reading from an External File (Method 1)

```
data extern1;  
    infile 'c:\sastalks\data1.txt';  
    input X Y A $ Z;  
run;
```

Reading from an External File (Method 2)

```
data extern2;  
    filename pat 'c:\sastalks\data1.txt';  
    infile pat;  
    input X Y A $ Z;  
run;
```

Infile Options

```
data none;
```

```
infile 'c:\sastalks\ext.dat';
```

```
input @1 X 1.  
      @2 A $9.;
```

```
run;
```

```
title 'No Options Used';
```

```
proc print data=none;
```

```
run;
```

File EXT.DAT

1long text|

2short|

3|

4last line|

No Options Used

Obs	X	A
1	1	long text
2	2	3
3	4	last line

NOTE: SAS went to a new line when INPUT statement reached past the end of a line.

Infile options: PAD

```
data pad;  
  infile 'c:\sastalks\ext.dat' pad;  
  input  @1 X 1.  
        @2 A $9.;  
  
run;  
title 'PAD Option';  
proc print data=pad;  
run;
```

```
File EXT.DAT  
1long text |  
2short |  
3 |  
4last line |
```

PAD Option

Obs	X	A
1	1	long text
2	2	short
3	3	
4	4	last line

Infile options: TRUNCOVER

```
data trunc;  
  infile 'c:\sastalks\ext.dat' truncover;  
  input  @1 X  1.  
        @2 A  $9.;  
  
run;  
title "TRUNCOVER Option";  
proc print data=trunc;  
run;
```

```
File EXT.DAT  
1long text |  
2short |  
3 |  
4last line |
```

TRUNCOVER Option		
Obs	X	A
1	1	long text
2	2	short
3	3	
4	4	last line

Infile Options: MISSOVER (1)

```
data miss;  
  infile 'c:\sastalks\miss.dat';  
  input X Y Z;  
run;  
  title 'No MISSOVER Option';  
proc print data=miss;  
run;
```

No MISSOVER Option			
Obs	X	Y	Z
1	1	2	3
2	4	5	6
3	9	10	11

File		
MISS.DAT		
1	2	3
4	5	
6	7	8
9	10	11

Infile Options: MISSOVER (2)

```
data miss;  
  infile "c:\sastalks\miss.dat" missover;  
  input X Y Z;  
run;  
title "MISSOVER Option";  
proc print data=miss;  
run;
```

MISSOVER Option

Obs	X	Y	Z
1	1	2	3
2	4	5	.
3	6	7	8
4	9	10	11

File		
MISS.DAT		
1	2	3
4	5	
6	7	8
9	10	11

Using MISSOVER with DATALINES

```
***Without MISSOVER option;
```

```
data miss;
```

```
    input x y z;
```

```
datalines;
```

```
1 2 3
```

```
4
```

```
5 6 7
```

```
;
```

```
title 'Without MISSOVER';
```

```
proc print data=miss;
```

```
run;
```

Without MISSOVER			
Obs	X	Y	Z
1	1	2	3
2	4	5	6

Using MISSOVER with DATALINES

```
data miss;  
    infile datalines missover;  
    input X Y Z;  
datalines;  
1 2 3  
4  
5 6 7  
;  
title 'With MISSOVER';  
proc print data=miss;  
run;
```

With MISSOVER			
Obs	X	Y	Z
1	1	2	3
2	4	.	.
3	5	6	7

Where It's @

```
data trailing;  
  input @6 Type $1. @;  
  if Type = '1' then input Age 1-2;  
  else if Type = '2' then  
    input Age 3-4;  
  drop Type;  
datalines;  
23 1  
44 2  
;  
title 'Single Trailing @';  
proc print data=trailing;  
run;
```

Single Trailing @	
Obs	AGE
1	23
2	44

Another @ Example

```
data trail2;  
  input @1 Gender $1. @;  
  if Gender ne 'f' then delete;  
  input @3 Age 2.  
        @5 Height 2.;  
datalines;  
m 2368  
f 4462  
;  
title 'Another @ Example';  
proc print data=trail2;  
run;
```

Another @ Example			
Obs	Gender	Age	Height
1	F	44	62

Where It's Really @@

```
data double;  
    input X Y @@;  
datalines;  
1 2 3 4 5 6  
7 8  
;  
title 'Double Trailing @';  
proc print data=double;  
run;
```

Double Trailing @		
Obs	X	Y
1	1	2
2	3	4
3	5	6
4	7	8

Reading Multiple Lines for One Observation

```
Data mult1;  
  Input #1  @1  ID      $11.  
           @13 DOB     mmddy8.  
  #2      @5  Height  2.  
           @8  Weight  3.;  
  Format dob mmddy10.;  
datalines;  
123-45-6789 10211946  
  68 158  
253-65-5455 11111960  
  62 102  
;
```

Reading Multiple Lines for One Observation

```
data mult2;  
  input #1   @1   ID           $11.  
         @13  DOB           mmddyy8.  
        #2   @5   Height      2.  
         @8   Weight      3.  
        #4;  
  format DOB mmddyy10.;  
datalines;  
123-45-6789 10211946  
   68 158  
9879876987698769876987  
0987098709870987098709  
253-65-5455 11111960  
   62 102  
9876987698769876987698  
0987098709870987098709  
;
```

Suppressing Error Messages

```
data error;  
    input X Y Z;  
datalines;  
1 na 3  
4 5 #  
;  
title 'Suppressing Error Messages';  
proc print data=error;  
run;
```

Suppressing Error Messages

```
94 data error;  
95     input X Y Z;  
96 datalines;
```

NOTE: Invalid data for Y in line 97 3-4.

RULE:-----1-----2-----3-----4-----+-

```
97 1 NA 3
```

```
X=1 Y=. Z=3 _ERROR_=1 _N_=1
```

NOTE: Invalid data for Z in line 98 5-5.

```
98 4 5 #
```

```
X=4 Y=5 Z=. _ERROR_=1 _N_=2
```

NOTE: The data set WORK.ERROR has 2 observations and 3 variables.

Suppressing Error Messages

```
data noerror2;  
    input X ?? Y ?? Z ??;  
datalines;  
1 na 3  
4 5 #  
;  
title 'Suppressing Error Messages';  
proc print data=noerror2;  
run;
```

```
112 DATA NOERROR2;  
113     INPUT X ?? Y ?? Z ??;  
114 DATALINES;
```

NOTE: The data set WORK.NOERROR2 has 2 observations
and 3 variables.

Suppressing Error Messages

```
data noerror3;  
    input @1 Date ?? mmddy10. @12 X 1.;  
    format Date mmddy8.;  
datalines;  
10/21/1946 3  
missing    4  
99/99/99   5  
;  
title 'Suppressing Error Messages';  
proc print data=noerror3;  
run;
```

NOTE: The data set WORK.NOERROR3 has 3 observations and 2 variables.

NOTE: DATA statement used (Total process time):

real time	0.02 seconds
-----------	--------------

cpu time	0.03 seconds
----------	--------------

Contact Information

Dr. Ron Cody

PO Box 5049 FM 480

Camp Verde, TX 78010

email: ron.cody@gmail.com

Author page: support.sas.com/cody