INPUT STATEMENT

The order which data values are entered the name of the SAS variables and their type.

We should use the input statement only for data values stored in external files or for data immediately following a cards or datalines statement.

Syntax:– INPUT <specification(s)><@|@@>;

Example:-

```
Data ds;
Infile datalines;
Input idno name$ team$ strtwght endwght;
Cards;
1023 David red 189 165
1049 Amelia yellow 145 124
1219 Alan red 210 192
1246 Ravi yellow 194 177
1078 Ashley red 127 118
1221 Jim yellow 220 .
;
Run;
```

Types of Input Statement

1) Column Input
2) List Input
   a) Simple list Input
   b) Modifier List Input
3) Formatted Input
4) Named Input
5) Null Input
6) Mixed Input

COLUMN INPUT:

The column numbers follow the variable name in the input statement that numbers indicate where the variable values are found in the input data records.

When to Use Column Input

With column input, the column numbers that contain the value follow a variable name in the INPUT statement. To read with column input, data values must be in

1. The same columns in all the input data records
2. Standard numeric form or character form
3. Useful features of column input are that
4. Character values can contain embedded blanks.
5. Character values can be from 1 to 32,767 characters long.
6. Input values can be read in any order, regardless of their position in the record.

EX: input id 1-3 name $ 5-15 Sal 16-20;

```
Data ds;
Infile datalines;
Input idno 1-4 name $ 6-23 team $ 25-30 strtwght 32-34 endwght 36-38;
Datalines;
1023 David Shaw          red    189 165
1049 Amelia Serrano      yellow 145 124
1219 Alan Nance          red    210 192
1246 Ravi Sinha          yellow 194 177
1078 Ashley McKnight     red    127 .
;
Run;
```

LIST INPUT:

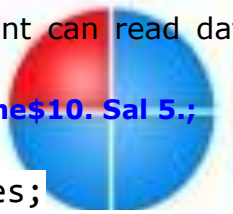
List inputs are 2 types.

1. Simple list input
2. Modified List input

Simple List Input:

The variables names are simply listed in the input statement.
A $ follows the name of each character variable.

The input statement can read data values that are separated by blanks or aligned in columns.



EX: input id 3. Name$10. Sal 5.;

```
Data ds;
Infile datalines;
Input idno 4. Name $19. Team $7. Strtwght 4. Endwght 4. ;
Cards;
1023 David Shaw          red    189 165
1049 Amelia Serrano      yellow 145 124
1219 Alan Nance          red    210 192
1078 Ashley McKnight     red    127 118
1221 Jim Brown           yellow 220 232
;
Run;
```

Modifier List Input

Modified list input makes the INPUT statement more versatile, because you can use a format modifier symbols to overcome several restrictions of simple list input.

List input is more versatile when you use format modifiers.

Format Modifier - Purpose

$ Indicates to store a variable value as a character value.

& reads character values that contain embedded blanks.

: reads data values that need the additional instructions that informats can provide but that are not aligned in columns.

~ reads delimiters within quoted character values as characters and retains the quotation marks.

+ Moves pointer to column N

Moves the pointer record N

/ Advances the pointer to column 1 of the next input record.

$

Indicates to store a variable value as a character value.

```
Data ds;
Infile Datalines;
Input name$ subject1 subject2 subject3 team$;
Datalines;
Joe 11 32 76 red
Mitchel 13 29 82 blue
Susan 14 27 74 green
;
Run;
```

&

Indicates that a character value can have one or more single embedded blanks.

This format modifier reads the value from the next non-blank column until the pointer reaches two consecutive blanks, the defined length of the variable, or the end of the input line, whichever comes first.

When raw data values are not aligned and character values contains one or more single spaces then to read data properly we need to use &. When you use & in input statement, you must specify 2 spaces after that variable value.

```
Data ds;
Infile datalines;
Input idno name&$ team$ strtwght endwght;
Datalines;
1023 David Shaw   red 189 165
1049 Amelia Serrano  yellow 145 124
1219 Alan Nance  red 210 192
1246 Ravi Sinha  yellow 194 177
1078 Ashley McKnight  red 127 118 s
1221 Jim Brown  yellow 220 .
;
Run;
Data ds;
Infile datalines;
Input idno name&$18. team$ strtwght endwght;
Cards;
1023 David Shaw   red 189 165
1049 Amelia Serrano  yellow 145 124
```

```
1219 Alan Nance  red 210 192
1246 Ravi Sinha  yellow 194 177
1078 Ashley McKnight  red 127 118
1221 Jim Brown  yellow 220 .
;
Run;
```

Restriction: The & modifier must follow the variable name and $ sign that it affects.

:

Enables you to specify an informat that INPUT statement uses to read the variable value.

For a character variable, this format modifier reads the value from the next non-blank column until the pointer reaches the next blank column, the defined length of the variable, or the end of the data line, whichever comes first. For a numeric variable, this format modifier reads the value from the next non-blank column until the pointer reaches the next blank column or the end of the data line, whichever comes first.

```
Data ds;
Infile datalines;
Input id name:$11. sal;
Datalines;
111 ramakrishna 34000
112 padmaja 20000
113 mohanprasad 50000
;
Run;
```



```
Data ds;
Infile datalines;
Input id name&:$12. sal;
Datalines;
111 rama krishna  34000
112 padmaja  20000
113 mohan prasad  50000
;
Run;
```

```
Data ds;
Infile datalines dsd;
Input id name&:$12. sal;
Datalines;
111,rama krishna,34000
112,padmaja,20000
113,mohan prasad,50000
;
Run;
```

Difference between &(ampersand)and : (colon)

&(Ampersand)we can use

When character values contain one more single space,
Raw data values having more than 8 characters,
Unaligned data,
With space delimiter.

:(Colon) we can use

When character values doesn't contain one more single space,
Raw data values having more than 8 characters,
Unaligned data,
With Space/ any special character as a delimiter.

```
Data ds;
Input item $10. Amount;
Datalines;
Trucks 1382
Vans 1235
Sedans 2391
;
Run;
```

wrong program

Just observe what result which it produces

```
Data ds;
Input item: $10. Amount;
Datalines;
Trucks 1382
Vans 1235
Sedans 2391
;
Run;
```

Right program

Just observe what result which it produces

~

Indicates to treat single quotation marks, double quotation marks, and delimiters in character values in a special way. This format modifier reads delimiters within quoted character values as characters instead of as delimiters and retains the quotation marks when the value is written to a variable.
Restriction: You must use the DSD option in an INFILE statement. Otherwise, the INPUT statement ignores this option.


```
Data ds;
Infile datalines dsd;
Input id name ~ $ sex$ age sal;
Datalines;
001,"abc",m,23,45000
002,"def",f,34,67000
003,"mno",m,21,36000
004,"xyz",f,27,45000
;
Run;
```

```
Data ds;
Infile datalines dsd;
Input Name: $9. Score1-Score3 Team ~ $25. Div $;
Datalines;
Joseph,11,32,76,"Red Racers, Washington",AAA
Mitchel,13,29,82,"Blue Bunnies, Richmond",AAA
Sue Ellen,14,27,74,"Green Gazelles, Atlanta",AA
;
Run;
```

+

Moves pointer columns N

```
Data ds;
Infile datalines;
Input team $6. +6 points 2.;
Cards;
red          59
blue         95
yellow       63
green        76
;
Run;
```

Multiple Input Statements

We can write multiple input statements or # format modifier to read the data when data is available in multiple lines for one record

```
Data ds;
Infile datalines;
Input Idno 1-4 name $7-20;
Input team $1-6;
Input strtwght 1-3 endwght 5-7;
Cards;
1023 David Shaw
red
189 165
1049 Amelia Serrano
yellow
145 124
1219 Alan Nance
red
210 192
1078 Ashley McKnight
red
127 118
1221 Jim Brown
yellow
220   .
;
Run;
```



```
Data ds;
Input Idno 1-4;
Input;
Input strtwght 1-3 endwght 5-7;
```

```
Cards;
1023 David Shaw
red
189 165
1049 Amelia Serrano
yellow
145 124
1219 Alan Nance
red
210 192
1246 Ravi Sinha
yellow
194 177
1078 Ashley McKnight
red
127 118
1221 Jim Brown
yellow
220   .
;
Run;
```


```
#
```

Moves the pointer to record N.

```
Data ds;
Input #1 name $ 6-23 idno 1-4
      #2 team $ 1-6
      #3 strtwght 1-3 endwght 5-7;
Cards;
1023 David Shaw
red
189 165
1049 Amelia Serrano
yellow
145 124
1219 Alan Nance
red
210 192
1246 Ravi Sinha
yellow
194 177
1078 Ashley McKnight
red
127 118
```

```
1221 Jim Brown
yellow
220   .
;
Run;
Data ds;
Infile datalines;
Input #2 team $ 1-6
      #1 name $ 6-23 idno 1-4
      #3 strtwght 1-3 endwght 5-7;
Cards;
1023 David Shaw
red
189 165
1049 Amelia Serrano
yellow
145 124
1219 Alan Nance
red
210 192
1246 Ravi Sinha
yellow
194 177
1078 Ashley McKnight
red
127 118
1221 Jim Brown
yellow
220   .
;
Run;
```

/

Advances the pointer to column 1 of the next input record.

```
Data ds;
Infile datalines;
Input idno 1-4 / / strtwght 1-3 endwght 5-7;
Datalines;
1023 David Shaw
red
189 165
1049 Amelia Serrano
yellow
```

```
145 124
1219 Alan Nance
red
210 192
1246 Ravi Sinha
yellow
194 177
1078 Ashley McKnight
red
127 118
1221 Jim Brown
yellow
220   .
;
Run;
```

FORMATTED INPUT:

An in format follows with the variable name in the input statement.

The in format gives the data type and the field width of an input value. In formats also to read data that are stored in non standard form, such as packed decimals or numbers that contain special characters such as command.

Ex: input @1 id 3. @5 name $10. @16 Sal 5.;


```
Data ds;
Infile datalines;
Input @1idno 4. @6name $18. @25team $5. @32strtwght 3. @36endwght 3.;
Datalines;
1023 David Shaw          red    189 165
1049 Amelia Serrano      yellow 145 124
1219 Alan Nance          red    210 192
1246 Ravi Sinha          yellow 194 177
1078 Ashley McKnight     red    127 118
1221 Jim Brown           yellow 220
;
Run;
```

NAMED INPUT:

We specify the name of the variable followed by an equal sign. SAS looks for n variable name and an equal sign in the input record.

EX: input id=3. Name=$10. Sal=5.;

```
Data ds;
Infile datalines;
Input Id= Name=$18. Team=$6. Strtwght= Endwght=3. ;
Cards;
```

```
ID=1023 NAME=David Shaw          TEAM=red    strtwght=189 endwght=165
ID=1049 NAME=Amelia Serrano      TEAM=yellow strtwght=145 endwght=124
ID=1219 NAME=Alan Nance          TEAM=red    strtwght=210 endwght=192
ID=1246 NAME=Ravi Sinha          TEAM=yellow strtwght=194 endwght=177
ID=1078 NAME=Ashley McKnight     TEAM=red    strtwght=127 endwght=118
ID=1221 NAME=Jim Brown           TEAM=yellow strtwght=220
;
Run;
Data ds;
Infile datalines;
Input Id= Name=$18. Team=$6. Strtwght= Endwght=3. ;
Cards;
NAME=David Shaw          TEAM=red    strtwght=189 endwght=165 ID=1023
NAME=Amelia Serrano      TEAM=yellow strtwght=145 endwght=124 ID=1049
NAME=Alan Nance          TEAM=red    strtwght=210 endwght=192 ID=1219
ID=1246 NAME=Ravi Sinha          TEAM=yellow strtwght=194 endwght=177
ID=1078 NAME=Ashley McKnight     TEAM=red    strtwght=127 endwght=118
ID=1221 NAME=Jim Brown           TEAM=yellow strtwght=220
;
Run;
```


When to Use Named Input:

Named input reads the input data records that contain a variable name followed by an equal sign and a value for the variable

The INPUT statement reads the input data record at the current location of the input pointer. If the input data records contain data values at the start of the record that the INPUT statement cannot read with named input, use another input style to read them.

Using Named Input with another Input Style

```
Data ds;
Infile datalines;
Input Id Name=$18. Team=$6. Strtwght= Endwght=3. ;
Cards;
1023 NAME=David Shaw          TEAM=red    strtwght=189 endwght=165
1049 NAME=Amelia Serrano      TEAM=yellow strtwght=145 endwght=124
1219 NAME=Alan Nance          TEAM=red    strtwght=210 endwght=192
1246 NAME=Ravi Sinha          TEAM=yellow strtwght=194 endwght=177
1078 NAME=Ashley McKnight     TEAM=red    strtwght=127 endwght=118
1221 NAME=Jim Brown           TEAM=yellow strtwght=220
;
Run;
/*Not reading Team values */
Data ds;
Infile datalines;
```

```
Input Id Name=$18. Team $6. Strtwght= Endwght=3.;
Cards;
1023 NAME=David Shaw          red    strtwght=189 endwght=165
1049 NAME=Amelia Serrano      yellow strtwght=145 endwght=124
1219 NAME=Alan Nance          red    strtwght=210 endwght=192
1246 NAME=Ravi Sinha          yellow strtwght=194 endwght=177
1078 NAME=Ashley McKnight     red    strtwght=127 endwght=118
1221 NAME=Jim Brown           yellow strtwght=220
;
Run;
Data ds;
Infile datalines;
Input Id name=$18. Team $30-36 Strtwght= Endwght=3. ;
Cards;
1023 NAME=David Shaw          red    strtwght=189 endwght=165
1049 NAME=Amelia Serrano      yellow strtwght=145 endwght=124
1219 NAME=Alan Nance          red    strtwght=210 endwght=192
1246 NAME=Ravi Sinha          yellow strtwght=194 endwght=177
1078 NAME=Ashley McKnight     red    strtwght=127 endwght=118
1221 NAME=Jim Brown           yellow strtwght=220
;
Run;
```


Reading Character Variables with Embedded Blanks

```
Data ds;
Informat Header $30. Name $15. ;
Input header= name=;
Datalines;
Header= age=60 AND UP Name=PHILIP
;
Run;
```

NULL INPUT:

The INPUT statement with no arguments (variables) is called a null INPUT.
The DATA step copies records from the input file to the output file without creating any SAS variables.

```
Data ds;
Input;
Datalines;
1023 David Shaw          red    189 165
1049 Amelia Serrano      yellow 145 124
1219 Alan Nance          red    210 192
1246 Ravi Sinha          yellow 194 177
1078 Ashley McKnight     red    127 118
```

```
1221 Jim Brown           yellow 220
;
Run;
```

In above program when input statement executes, one record at a time is storing into input buffer then pdv picks data from input buffer and assigns to corresponding variables But there is no variables so it's create zero variable dataset.

MIXED INPUT:

The input statement with all input styles called mixed input

EX: input city = $1-8. State = $6. Date mmddyy8.

```
Data ds;
Infile datalines;
Input Idno Name $ 6-23 @25Team $7. Strtwght 3.  Endwght 36-38;
Cards;
1023 David Shaw          red    189 165
1049 Amelia Serrano      yellow 145 124
1219 Alan Nance          red    210 192
1246 Ravi Sinha          yellow 194 177
1078 Ashley McKnight     red    127 118
1221 Jim Brown           yellow 220
;
Run;
```


INPUT SPECIFICATIONS

@ (Single Trailing)

Holds an input record for the execution of the next INPUT statement within the same iteration of the DATA step.

This line-hold specifier is called trailing @.

Restriction: The trailing @ must be the last item in the INPUT statement.

```
Data redteam;
Infile datalines;
Input team $ 13-18 @;
If team='red';
Input idno 1-4 strtwght 20-23 endwght 24-26;
Cards;
1023 David  red    189 165
1049 Amelia yellow 145 124
1219 Alan   red    210 192
1246 Ravi   yellow 194 177
1078 Ashley red    127 118
1078 Ashley red    127 118
1221 Jim    yellow 220   .
;
```

```
Run;
```

In above program when it's reading data it holds in team value and it check condition, If condition is true it pick the observation otherwise it goes for next record. Again in next record also it works same.
So it is useful to reduce execution time.

```
Data ds;
Input id 1-3 @;
If id in (101,103,105) ;
/*if id=101;    ---- Only for one value*/
Input name$ 5-7 sal 9-12;
Datalines;
101 abc 1000
102 asd 2000
103 dfg 3000
104 hjk 3400
105 xyz 5000
;
Run;
```

Using @ we can reduce the execution time of the program
but it works for align data only

@@ (Double Trailing)

Holds an input record for the execution of the next INPUT statement across iterations of the DATA step.

This line-hold specifier is called double trailing @@.
Restriction: The double trailing @ must be the last item in the INPUT statement.

When our raw data values are in single line for multiple observations to read that we need to use @@

Examples:-

```
Data zone_temp;
Infile datalines;
Input zone $ tempc tempf @@;
Cards;
East 34 89 west 25 78 north 33 82 south 42 98
;
Run;
```

```
Data city_temp;
Infile datalines;
Input city$ tempc tempf@@;
Cards;
Delhi 34 89 Kolkata 25 78
Mumbai 33 82 Chennai 42 98
```

```
Hyderabad 40 94 Bangalore 31 79
Pune 28 71
;
Run;
```

Interview Questions

Q1) What is INPUT statement? Syntax? Example?
Q2) Name the types of INPUT statement?
Q3) Why the input types are required ?
Q4) What is column input? Syntax? Example?
Q5) What is formatted input? Syntax? Example?
Q5) What is simple list input? Syntax? Example?
Q6) What is modifier list input?
Q7) Name the format modifiers in modifier list input?
  a) What is $ in modifier list input? Write about that with an example?
  b) What is & in modifier list input? Write about that with an example?
  c) What is : in modifier list input? Write about that with an example?
  d) What is ~ in modifier list input? Write about that with an example?
  e) What is + in modifier list input? Write about that with an example?
  f) What is # in modifier list input? Write about that with an example?
  g) What is / in modifier list input? Write about that with an example?
  h) What is multiple input statement? Write about that with an example?
Q8) What is named input? Syntax? Example?
Q9) What is null input? Syntax? Example?
Q10) How can you create 0(zero) Variable dataset?
Q11) What is mixed input? Syntax? Example?
Q12) What are the specifications are there in input statements?
Q13) What is @ (single trailing) ? Syntax? Example?
Q14) What is @@ (double trailing) ? Syntax? Example?
Q15) What is PUT Statement ? Syntax? Example?
Q16) Data is available in the location of "E:\SAS\RAW_DATA\IQ\INPUTSTATEMENT"
  tasks are available at TASK notepad in above same location?
  read that data into SAS and write the program for each task?


STANSYS
SOFTWARE SOLUTIONS