

Topic 1 & 2 Introduction to SAS

basic structure

```
data dataset_name; input num_input string_input$; datalines; 1 2 3 ;
```

Variabel Names

- must start with letter or underscore
- no longer than 32 char
- no special characters such as commas
- sas is not case sensitive

proc print

```
proc print data=data_name;  
title 'some title';  
var var1 var2;  
run;
```

proc sort

```
proc sort data=data_name;  
by subject;  
run;
```

if statement

```
data some_data;  
input some_input;  
if final GE 0 and final LT 41 then grade = 'F';  
else if FINAL GE 41 AND FINAL LT 60 THEN GRADE="D";  
datalines;
```

note:

- GE: greater or equal
- LE: less or equal
- GT: greater than
- LT: less than

proc freq

```
proc freq data=some_data;
order = freq1 # order by frequency from high to low
table var1 var2 # here two tables are requested
run;
```

```
table var1 var2 / nocum; # no cumulative frequency
```

Topic 3: Inputting data

infile statement

```
data some_data;
infile 'path_to_file' delimiter = ',' dsd'
input
var 1
var 2
;
run;
```

note: "dsd":

- ignores delimiters in data values enclosed in quotation marks
- does not read quotation marks as part of the data value
- treats two consecutive delimiters as a missing value

column input

```
Data toads_1;
      Input Name      $      1-6
              Weight      8-10
              Jump1      12-14
              Jump2      16-18
              Jump3      20-22
      ;
      Datalines;
Lucky  2.3 1.9 2.3 3.0
Spot   4.6 2.5 3.1 0.5
Tubs   7.1 2.2 3.1 3.8
Hop     4.5 3.2 1.9 2.6
Noisy  3.8 1.3 1.8 1.5
Winner 5.7 2.2 1.3 2.8
      ;
```

restrictions:

- Each of the variable values in a data line must be in the same position in every data line. (Column Delimited)
- Numeric values include any necessary decimal points.
- Data doesn't include any dates or other values which need special treatment (such as numbers with commas)

Advantages over List Input:

- Spaces are not required between values.
- Missing values can be left blank
- Character data can have embedded spaces
- Skip unwanted variables

formatted input

```
input  
    var1 informat1  
    var2 informat2
```

<\$> informat-namew.<d>

- "\$": using a "\$" indicates a character informat
- "w": total number of column positions to read
- ".": required delimiter
- 'd': number of decimal places if dealing with a quantitative variable

format	Description
\$CHARw	Reads w column positions of character data and preserves leading blanks.
COMMAw	Reads w column positions of numeric data and removes selected nonnumeric characters such as dollar signs and commas.
MMDDYY8.	Reads dates of the form 10/29/01