## ****Informats And Formats In SAS****

It is important that you understand this topic well if you want to be good at SAS programming. If you can recall, I mentioned earlier that SAS has two standard variable types:

* Numeric
* Character

When SAS comes across non standard variables, SAS will throw an error or you won’t get the desired output. To overcome this problem, SAS uses Informats and Formats.

## ****Informat****

Informats are typically used to read or input data from external files or flat files (like text files or sequential files). The informat instructs SAS on how to read data into SAS variables. SAS  has three types of Informats: character, numeric, and date/ time. Informats are named according to the following syntax structure:

* Character Informat: $INFORMATw.
* Numeric Informat: INFORMATw.d
* Date/ Time Informat: INFORMATw.

The ‘$’ indicates a character informat. INFORMAT refers to the sometimes optional SAS informat name. The ‘w’ indicates the width (bytes or number of columns) of the variable. The ‘d’ is used for numeric data to specify the number of digits to the right of the decimal place. All informats must contain a decimal point(.) so that SAS can  
differentiate an informat from a SAS variable.

Let us go back to our previous code and see if Date/ Time Informat helps us. So let’s change the code accordingly and add a Date Informat to it as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | DATA Employee\_Info;  input Emp\_ID Emp\_Name$ Emp\_Vertical$ DOJ;  INFORMAT DOJ ddmmyy10.;  datalines;  101 Mak SQL 18/08/2013  102 Rama SAS 25/06/2015  103 Priya Java 21/02/2010  104 Karthik Excel 19/05/2007  105 Mandeep SAS 11/09/2016  ;  Run;  PROC PRINT DATA=Employee\_Info;  Run; |

Line number 3 in the code instructs SAS to read in the variable ‘date of joining’ (DOJ) using the date  
informat MMDDYYw. For each date field occupies 10 spaces, the ‘w.’ qualifier is set to 10.

## ****Format****

Informats are the instructions for reading data, whereas formats are the instructions used to display or output data. Defining a format for a variable is how you tell SAS to display the values in the variable. Formats are grouped into the same three classes as informats (character, numeric, and date-time) and also always contain a dot.

The general form of a format statement is:

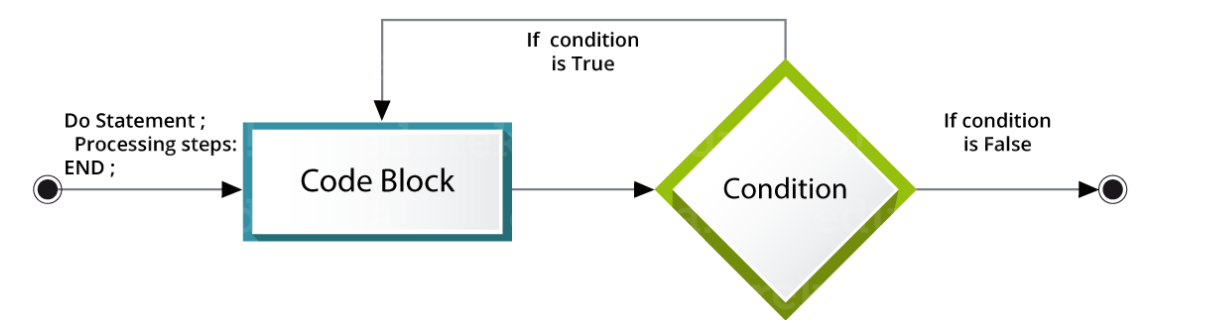
* FORMAT variable-name FORMAT-NAME.;

Let us go back to our code having dataset Employee\_Info to see if we can display the date correctly using FORMAT command.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | DATA Employee\_Info;  input Emp\_ID Emp\_Name$ Emp\_Vertical$ DOJ;  INFORMAT DOJ ddmmyy10.;  FORMAT DOJ ddmmyy10.;  datalines;  101 Mak SQL 18/08/2013  102 Rama SAS 25/06/2015  103 Priya Java 21/02/2010  104 Karthik Excel 19/05/2007  105 Mandeep SAS 11/09/2016  ;  Run;  PROC PRINT DATA=Employee\_Info;  Run; |

## ****SAS Loops****

While doing SAS programming, we may encounter situations where we repeatedly need to execute a block of code several number of times. It is inconvenient to write the same set of statements again and again. This is where loops come into picture. In SAS, the Do statement is used to implement loops. It is also known as the Do Loop. The image below shows the general form of the Do loop statements in SAS.



Following are the  types of DO loops in SAS:

* **Index**: The loop continues from the start value till the stop value of the index variable.
* **While**: The loop continues as long as the **While** condition becomes false.
* **Until**: The loop continues till the **Until** condition becomes True.

## ****Do Index loop****

We use an index variable as a start and stop value for **Do Index loop**. The SAS statements get executed repeatedly till the index variable reaches its final value.  
**Syntax:**

|  |  |
| --- | --- |
| 1  2  3 | Do indexvariable = initialvalue to finalvalue;  SAS statements;  End; |

Let us take a look at sample code to understand Do Index Loop. In the below code, VAR is the index variable.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | DATA SampleLoop;  SUM=0;  Do VAR = 1 to 10;  SUM = SUM + VAR;  END;  PROC PRINT DATA = SampleLoop;  Run; |

**Do While Loop**

The **Do While** loop uses a WHILE condition. This Loop executes the block of code when the condition is true and keeps executing it, till the condition becomes false. Once the condition becomes false, the loop is terminated.

**Syntax:**

|  |  |
| --- | --- |
| 1  2  3 | Do While (condition);  SAS statements;  End; |

Following sample code will help you understand DO WHILE loop.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | DATA SampleLoop;  SUM=0;  VAR=1;  Do While(VAR<15);  SUM = SUM + VAR;  VAR+1;  END;  PROC PRINT DATA = SampleLoop;  Run; |

**Do Until Loop**

The Do Until loop uses an **Until** condition.This Loop executes the block of code when the condition is false and keeps executing it, till the condition becomes true. Once the condition becomes true, the loop is terminated.

**Syntax:**

|  |  |
| --- | --- |
| 1  2  3 | Do Until (condition);  SAS statements;  END; |

Let us take a look at sample program.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | DATA SampleLoop;  SUM=0;  VAR=1;  Do Until(VAR>15);  SUM=SUM+VAR;  VAR+1;  END;  PROC PRINT;  Run; |

[Home](https://www.listendata.com/) » [SAS](https://www.listendata.com/search/label/SAS?&max-results=8) » [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners?&max-results=8) » Importing Data into SAS

# IMPORTING DATA INTO SAS

[Deepanshu Bhalla](https://plus.google.com/+DeepanshuBhalla/posts) [27 Comments](https://www.listendata.com/2013/04/sas-reading-importing-raw-data-into-sas.html#comment-form) [SAS](https://www.listendata.com/search/label/SAS), [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners)

This tutorial will show you how to input data into SAS. It also covers how to import external data to SAS. It includes examples of importing most common formats such as CSV, Excel File and Text Files etc. After finishing this tutorial, you would be comfortable how to extract data into SAS.

|  |
| --- |
| [https://1.bp.blogspot.com/-n4snA3XqkxA/WMxBN8-sXVI/AAAAAAAAF-0/TT6E3VUjpjQ0TX76k-8Lo8uRAo-zebAGwCLcB/s1600/Import%2Bdata%2Bsas.png](https://1.bp.blogspot.com/-n4snA3XqkxA/WMxBN8-sXVI/AAAAAAAAF-0/TT6E3VUjpjQ0TX76k-8Lo8uRAo-zebAGwCLcB/s1600/Import+data+sas.png) |
| Import Data to SAS |

**I. Entering Data Directly in SAS Program**  
You can enter your lines of data directly in your SAS program by using a **DATALINES**statement.  
  
***Let's start out by clarifying the main keywords associated with the following program.***

The keywords are as follows:  
**1. DATA -**  The DATA step always begins with a **DATA statement**. The purpose of the DATA statement is to tell SAS that you are creating a new data set i.e. **outdata.**  
**2. INPUT -**To define the variables used in data set.  
**3. Dollar sign ($) -**To identify variable as character.

**4. DATALINES**- To indicate that lines following DATALINES statement a real data.  
**5. PROC PRINT** - To print out the contents of data set in output window.  
**6. RUN** - The DATA step ends with a RUN statement.

DATA outdata;

INPUT age gender $ dept obs1 obs2 obs3;

DATALINES;

1 F 3 17 6 24

1 M 1 19 25 7

3 M 4 24 10 20

3 F 2 19 23 8

2 F 1 14 23 12

2 M 5 1 23 9

3 M 1 8 21 7

1 F 1 7 7 14

3 F 2 2 1 22

1 M 5 20 5 2

3 M 4 21 8 18

1 M 4 7 9 25

2 F 5 10 17 20

3 F 4 21 25 7

3 F 3 9 9 5

3 M 3 7 21 25

2 F 1 1 22 13

2 F 5 20 22 5

;

proc print;

run;

[Home](https://www.listendata.com/) » [SAS](https://www.listendata.com/search/label/SAS?&max-results=8) » [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners?&max-results=8) » Importing Data into SAS

# IMPORTING DATA INTO SAS

[Deepanshu Bhalla](https://plus.google.com/+DeepanshuBhalla/posts) [27 Comments](https://www.listendata.com/2013/04/sas-reading-importing-raw-data-into-sas.html#comment-form) [SAS](https://www.listendata.com/search/label/SAS), [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners)

This tutorial will show you how to input data into SAS. It also covers how to import external data to SAS. It includes examples of importing most common formats such as CSV, Excel File and Text Files etc. After finishing this tutorial, you would be comfortable how to extract data into SAS.

|  |
| --- |
| [https://1.bp.blogspot.com/-n4snA3XqkxA/WMxBN8-sXVI/AAAAAAAAF-0/TT6E3VUjpjQ0TX76k-8Lo8uRAo-zebAGwCLcB/s1600/Import%2Bdata%2Bsas.png](https://1.bp.blogspot.com/-n4snA3XqkxA/WMxBN8-sXVI/AAAAAAAAF-0/TT6E3VUjpjQ0TX76k-8Lo8uRAo-zebAGwCLcB/s1600/Import+data+sas.png) |
| Import Data to SAS |

**I. Entering Data Directly in SAS Program**  
You can enter your lines of data directly in your SAS program by using a **DATALINES**statement.  
  
***Let's start out by clarifying the main keywords associated with the following program.***

The keywords are as follows:  
**1. DATA -**  The DATA step always begins with a **DATA statement**. The purpose of the DATA statement is to tell SAS that you are creating a new data set i.e. **outdata.**  
**2. INPUT -**To define the variables used in data set.  
**3. Dollar sign ($) -**To identify variable as character.

**4. DATALINES**- To indicate that lines following DATALINES statement a real data.  
**5. PROC PRINT** - To print out the contents of data set in output window.  
**6. RUN** - The DATA step ends with a RUN statement.

  
  
  
You can also use **CARDS**instead of **DATALINES**. Both means the same. There is no difference between these two keywords. See the program below -

*DATA outdata;  
   INPUT age gender $ dept obs1 obs2 obs3;****CARDS;*** *1 F 3 17 6 24  
;  
proc print;  
run;*

**Reading Delimited Data**  
 **The default delimiter is blank.** If you have a data file with other delimiters such as comma or tab you need to define the delimiter before defining the variables using **INFILE** and **DLM =**options.

***Syntax :****Infile 'file-description' dlm=','*

1. For tab delimiter, the syntax would be **infile 'file-description' dlm='09'x**
2. For colon delimiter, the syntax would be **infile 'file-description' dlm=':'**

DATA outdata;

INFILE Datalines dlm =",";

INPUT age gender $ dept obs1 obs2 obs3;

Datalines;

1,F,3,17,6,24

1,M,1,19,25,7

3,M,4,24,10,20

3,F,2,19,23,8

2,F,1,14,23,12

;

proc print;

run;

**Importing External Data into SAS**  
  
**Method I : PROC IMPORT**  
  
**PROC IMPORT** is a SAS procedure to import external files into SAS. It automates importing process. You don't need to specify variable type and variable length to import an external file. It supports various formats such as excel file, csv, txt etc.  
  
**1. Importing an Excel File into SAS**  
The main keywords used in the following program are :

**1. OUT -** To specify name of a data set that SAS creates. In the program below, **outdata** is the data set saved in work library (temporary library)

**2. DBMS -** To specify the type of data to import.

**3. REPLACE -** To overwrite an existing SAS data set.

**4. SHEET -** To import a specific sheet from an excel workbook

**5. GETNAMES -** To include variable names from the first row of data.

PROC IMPORT DATAFILE= "c:\deepanshu\sampledata.xls"

OUT= outdata

DBMS=xls

REPLACE;

SHEET="Sheet1";

GETNAMES=YES;

RUN;

1. **Importing a Tab-Delimited File into SAS**  
   The program below is similar to the code of importing excel file. The only difference is **DBMS = DLM and delimter = '09'x.**

PROC IMPORT DATAFILE= "c:\deepanshu\sampledata.txt"

OUT= outdata

DBMS=dlm

REPLACE;

delimiter='09'x;

GETNAMES=YES;

RUN;

**Method II : Get External File - INFILE**  
  
In SAS, there is one more method called **INFILE**to import an external file. It's a manual method of importing an external file as you need to specify variables and its types and length.  
 **1. Reading a CSV File**  
 **INFILE statement -** To specify path where data file is saved.

**DSD -** To set the default delimiter from a blank to comma.  
**FIRSTOBS=2 :**To tell SAS that first row contains variable names and data values starts from second row.

data outdata;

infile 'c:\users\deepanshu\documents\book1.csv' dsd firstobs=2;

input id age gender $ dept $;

run;

**2. Reading a TAB Delimited File**  
  
We can use DLM='09'x to tell SAS that we are going to import a tab delimited file. The **TRUNCOVER**statement tells SAS to assign the raw data value to the variable even if the value is shorter than expected by the INPUT statement.

*data outdata;  
  infile 'c:\deepanshu\dummydata.txt' DSD dlm='09'x truncover;  
  input employee :$30. DOJ :mmddyy8. state :$20.;  
run;*

**How to handle an external file :**  
Using a **FILENAME** statement to handle an external file.

***FILENAME sample 'c:\deepanshu\sampledata.csv' ;*** *DATA outdata;  
infile****sample****dsd;  
INPUT age gender $ dept obs1 obs2 obs3;  
run;*

[Home](https://www.listendata.com/) » [SAS](https://www.listendata.com/search/label/SAS?&max-results=8) » [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners?&max-results=8) » SAS keyboard shortcuts that every analyst must know

# SAS KEYBOARD SHORTCUTS THAT EVERY ANALYST MUST KNOW

[Deepanshu Bhalla](https://plus.google.com/+DeepanshuBhalla/posts) [8 Comments](https://www.listendata.com/2014/07/sas-keyboard-shortcuts-that-every.html#comment-form) [SAS](https://www.listendata.com/search/label/SAS), [SAS For Beginners](https://www.listendata.com/search/label/SAS%20For%20Beginners)

If you fancy yourself a rockstar analyst, you know how valuable it is to keep your hands on the keyboard. Apart from making you work more efficiently and faster, you can also impress your friends or colleagues by being able to work without a mouse. Here is a list of SAS keyboard shortcuts that can make your life easy while working with SAS.

|  |
| --- |
| [https://2.bp.blogspot.com/-5b9DXYDTfIg/V8b9ABTRfrI/AAAAAAAAFSA/letbXQ78sIw28B3NvbWyyt_TUhuCIDN2QCLcB/s1600/sas%2Bshortcuts.png](https://2.bp.blogspot.com/-5b9DXYDTfIg/V8b9ABTRfrI/AAAAAAAAFSA/letbXQ78sIw28B3NvbWyyt_TUhuCIDN2QCLcB/s1600/sas+shortcuts.png) |
| SAS Keyboard Shortcuts |

**Important SAS Shortcuts for Productivity**  
  
The following is a list of productivity boosting SAS keyboard shortcuts. It would bring efficiency in writing SAS programming code.

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Description** | **Shortcut Key** |
| Run or submit a program | F3 or F8 |
| Comment the selected code (/) | Ctrl + / |
| Uncomment the selected code (/) | Ctrl + Shift + / |
| Stop Processing or Cancel Submitted Statement | Ctrl + Break |
| Convert selected text to upper case | Ctrl + Shift + U |
| Convert selected text to lower case | Ctrl + Shift + L |
| Find text | Ctrl + F |
| Find and replace text | Ctrl + H |
| Copy Selection | Ctrl + C |
| Paste | Ctrl + V |
| Cut Selection | Ctrl + X |
| Go to a particular line | Ctrl + G |
| To move curser to the matching DO/END statemen | Alt + [ or Alt + ] |
| To move cursor to matching brace/parentheses | Ctrl + [ or Ctrl + ] |
| Move to beginning of line | Home |
| Move to top | CTRL+Home |
| Move to end | CTRL+End |
| To close the active window | CTRL+F4 |
| To exit the SAS system | ALT+F4 |

**Important Shortcuts that work only in SAS Enterprise Guide**

|  |  |
| --- | --- |
| **Description** | **Shortcut Key** |
| Ctrl+End | Go to the last record, last column |
| Ctrl+Home | Go to the first record, first column |
| Ctrl+G | Go to specific row or column |
| Ctrl + right arrow | Move to last column |
| Ctrl + left arrow | Move to first column |
| F2 | Rename dataset |
| Ctrl + I | Format ugly code (Select the code and press Ctrl I) |

**Other Useful SAS Keyboard Shortcuts**

|  |  |
| --- | --- |
| **Description** | **Shortcut Key** |
| Bring up word tip | Alt + F1 + No Selection |
| Hide the current word tip | Esc |
| Collapse all folding blocks | Alt + Ctrl + Number pad - |
| Expand all folding blocks | Alt + Ctrl + Number pad + |
| Execute the last recorded macro | Ctrl + F1 |
| Undo edit | Ctrl + Z |
| Redo edit | Ctrl + Y |
| Clear window | Ctrl + E |
| Paste program below | F4 |
| Get Help for a SAS procedure | Place the cursor within a procedure name and press F1 |
| Context Help | F1 |
| Move cursor to next case change | Alt + Right |
| Move cursor to previous case change | Alt + Left |
| File window | Ctrl + Q |
| Explorer window | Ctrl + W |
| Titles window | Ctrl + T |
| System options window | Ctrl + I |
| Open file window | Ctrl + O |
| Save as window | Ctrl + S |
| Active libraries window | Ctrl + D |
| Select all | Ctrl + A |
| Clean up white space | Ctrl + Shift + W |
| Comment the selection with line comments | Ctrl + / |
| Undo the Comment | Ctrl + Shift + / |
| Convert the selected text to lowercase | Ctrl + Shift + L |
| Convert the selected text to uppercase | Ctrl + Shift + U |
| Submit selected code | F8 |
| Log | F6 |
| Output | F7 |
| Editor | F8 |
| Keys | F9 |
| Next Window | Ctrl + Tab |
| Tile | Shift + F4 |
| Cascade | Shift + F5 |
| Next window | Ctrl + F6 |

**How to create keyboard shortcuts in SAS**

1. Open the **Enhanced Editor** window within SAS.
2. From the toolbar, select **Tools --> Options --> Keys.**
3. Scroll down to the keystroke you would like to assign to the series of commands, looking for a keystroke that has no assignment.
4. Add the command code under the definition heading. For example: log; clear; output;clear;
5. Close the Keys window.