

# Essentials

Access Data

Explore Data

Prepare Data

Analyze and Report on Data

Export Results

Foundation

write  
and  
submit  
code

```
data myclass;  
    set sashelp.class;  
run;  
  
proc print data=myclass;  
run;
```

Editor

view  
messages  
from SAS

```
1 data myclass;  
2   set sashelp.class;  
3 run;
```

NOTE: There were 19 observations read from the data set SASHELP.CLASS.  
NOTE: The data set WORK.MYCLASS has 19 observations and 5 variables.  
NOTE: DATA statement used:  
real time 0.01 seconds  
cpu time 0.00 seconds

```
4  
5 proc print data=myclass;  
NOTE: Writing HTML Body file: sashtml.htm  
6 run;
```

NOTE: There were 19 observations read from the data set WORK.MYCLASS.

Log

view  
results

Name	Sex	Age	Height	Weight
Alfred	M	14	69.0	112.5
Alice	F	13	56.5	84.0
Barbara	F	13	65.3	98.0
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.2	82.0
Janet	F	15	62.5	112.5

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5

Results and  
Output Data

write  
and  
submit  
code

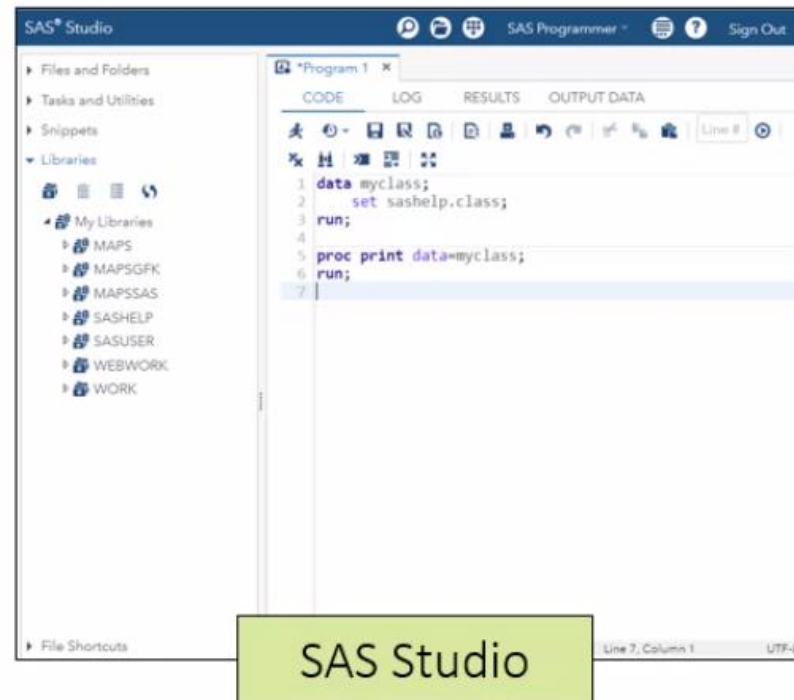
Editor



code  
completion

formatting

syntax  
coloring



## Practice: Exploring SAS Studio

1. Start SAS Studio. The main window of SAS Studio consists of a navigation pane on the left and a work area on the right.
2. Options are available in the banner area to customize your SAS Studio environment.

Option	Description
<b>Search</b>	Search files and folders.
<b>Open</b>	Open files from your files and shortcuts.
<b>New Options</b>	New program, new import data, new query, close all tabs, and maximize view.
<b>SAS Programmer</b>	SAS Studio includes two different perspectives: the SAS Programmer perspective and the Visual Programmer perspective. A perspective is a predetermined set of features that is customized to meet the needs of a specific type of user. This course is about programming in SAS, so you need to make sure that the SAS Programmer perspective is selected on the toolbar at the top of the application. You can find more information about both perspectives in SAS® Studio: <i>User's Guide</i> .
<b>More application options</b>	More application options, including edit autoexec file, a view menu, preferences, tool options, background submission status, and reset SAS session.
<b>Help</b>	A Help menu including SAS Studio Help, SAS Product Documentation, and About SAS Studio.

3. On the **Program 1** tab in the work area, type or copy and paste the code below. This is a simple SAS program called a DATA step.

**Note:** If you copy and paste the program, you can click the **Format Code** icon on the toolbar to improve the program spacing.

```
data work.shoes;
    set sashelp.shoes;
    NetSales=Sales>Returns;
run;
```

4. Click **Run** or press F3 to submit the code. Examine the LOG and OUTPUT DATA tabs. The RESULTS tab is empty because the program did not create a report.
5. On the CODE tab, add code to compute summary statistics.
  - a. At the end of the program, begin by typing **pr**. Notice that a prompt appears with valid keywords and syntax help.
  - b. Press Enter to add the word **proc** to the program.
  - c. Press the spacebar and type **me**. Press Enter again to add **means** to the program.

**Note:** The Autocomplete prompts also include a window with syntax help and links to documentation and examples.

6. Press the spacebar, use the prompt to select **data=**, and then type **work.shoes**.
  - a. Press the spacebar and notice that the prompt lists all valid options.
  - b. Type or select the options in the window to complete the following statement:

```
proc means data=work.shoes mean sum maxdec=2;
```

**Note:** If you prefer to disable autocomplete, click **More Application options**, and then **Preferences > Editor**. Clear the **Enable autocomplete** check box and click **Save**.

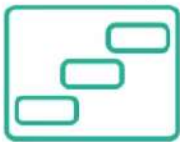
7. Complete the program by adding the following VAR, CLASS, and RUN statements:

```
proc means data=work.shoes mean sum maxdec=2;  
    var NetSales;  
    class region;  
run;
```

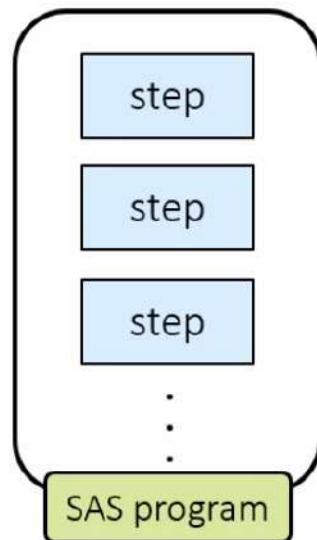
8. Highlight the code from PROC MEANS through the RUN statement and click **Run** or press F3 to run only the selected portion. Confirm the results.
9. To view multiple tabs at the same time, click the **RESULTS** tab and drag it to the right side of the work area until a highlighted region appears. To return to a single window, drag the **RESULTS** tab back to the main tab area.
10. On the RESULTS tab, click the **HTML**, **PDF**, or **Word** icon to open the results in the corresponding file format. You are prompted to open the file in the browser.

**Note:** Additional options for the output formats are available. Click the **More application options** icon on the banner, and then select **Preferences > Results** to view the options.
11. To save the program, return to the CODE tab and click the **Save as** toolbar button.
  - a. Navigate to the **output** folder in the course files.
  - b. Enter **shoesprogram** in the **Name** field and click **Save**.

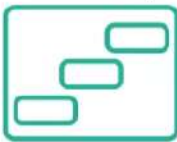
DATA step



PROC step



DATA step

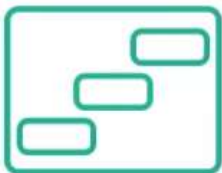


PROC step



```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;  
  
proc print data=myclass;  
run;  
  
proc means data=myclass;  
    var age heightcm;  
run;
```

DATA step



```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;
```

```
proc print data=myclass;  
run;
```

```
proc means data=myclass;  
    var age heightcm;  
run;
```

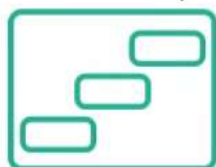


input source

**A DATA step generally reads data from an input source,**



DATA step



```
data myclass;  
  set sashelp.class;  
  heightcm=height*2.54;  
run;
```

```
proc print data=myclass;  
run;
```

```
proc means data=myclass;  
  var age heightcm;  
run;
```



input source



SAS table

processes it, and creates a SAS table.

DATA step



```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;
```

```
proc print data=myclass;  
run;
```

```
proc means data=myclass;  
    var age heightcm;  
run;
```

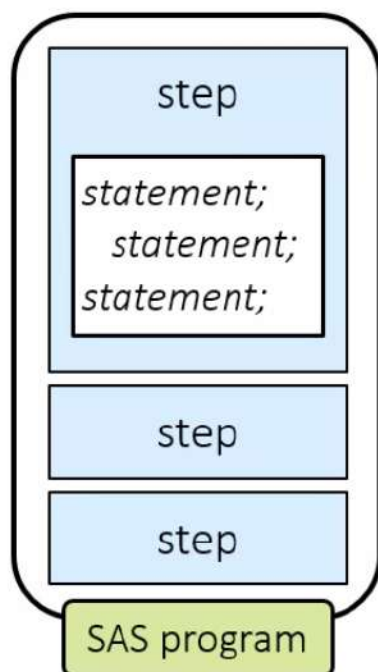
In this program, the DATA step is creating an output table and adding a new column.

PROC step



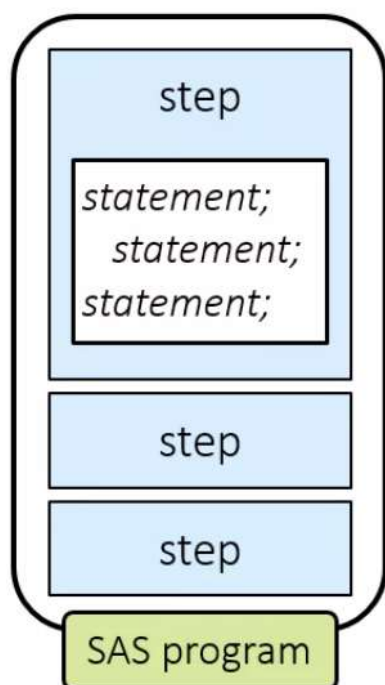
```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;  
  
proc print data=myclass;  
run;  
  
proc means data=myclass;  
    var age heightcm;  
run;
```

A PROC or procedure step processes a SAS table in a specific a predefined way.



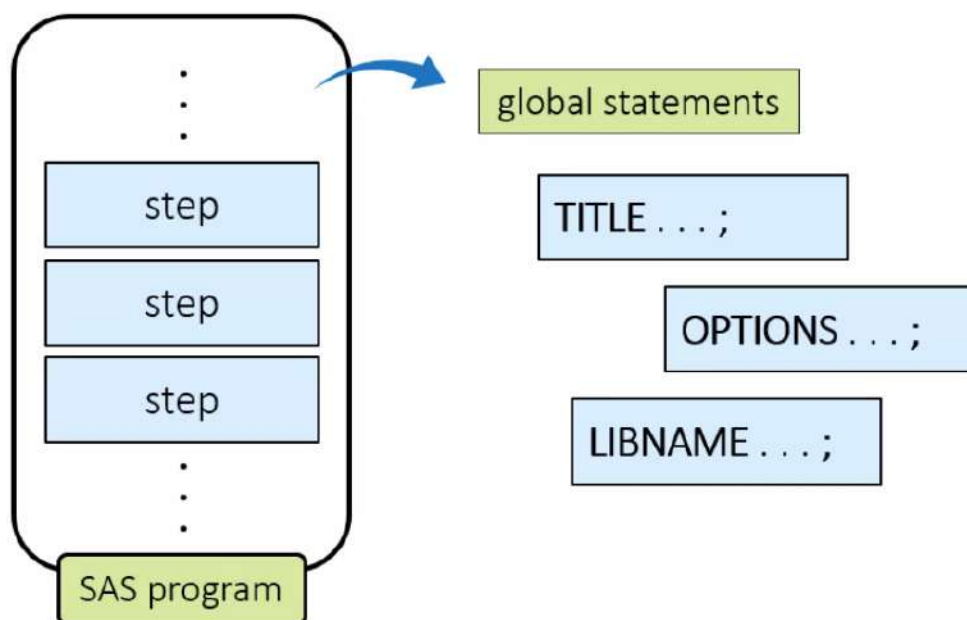
```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;  
  
proc print data=myclass;  
run;  
  
proc means data=myclass;  
    var age heightcm;  
run;
```

Each step consists of a sequence of statements,

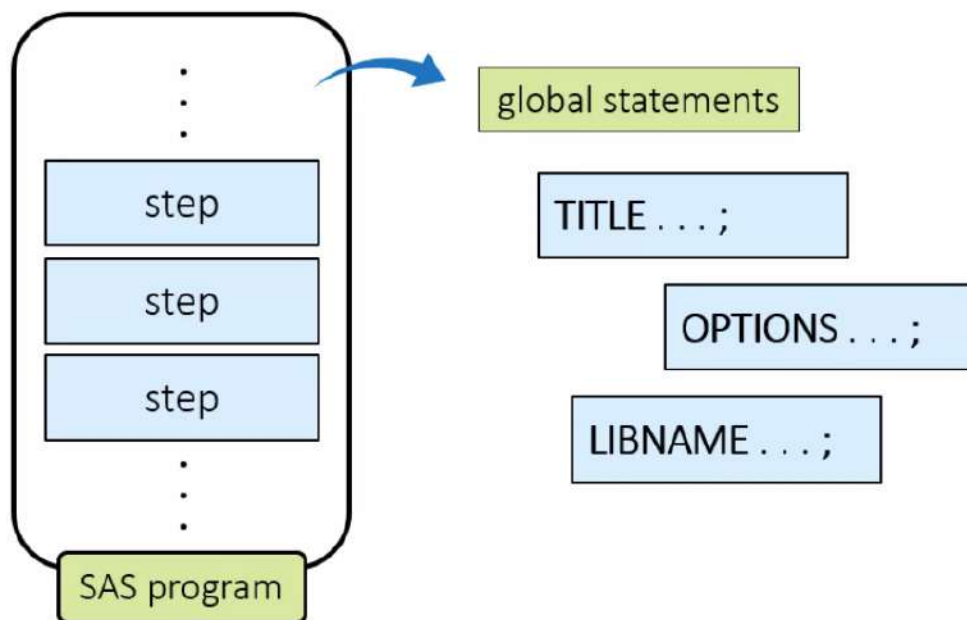


```
data myclass;  
    set sashelp.class;  
    heightcm=height*2.54;  
run;  
  
proc print data=myclass;  
run;  
  
proc means data=myclass;  
    var age heightcm;  
run;
```

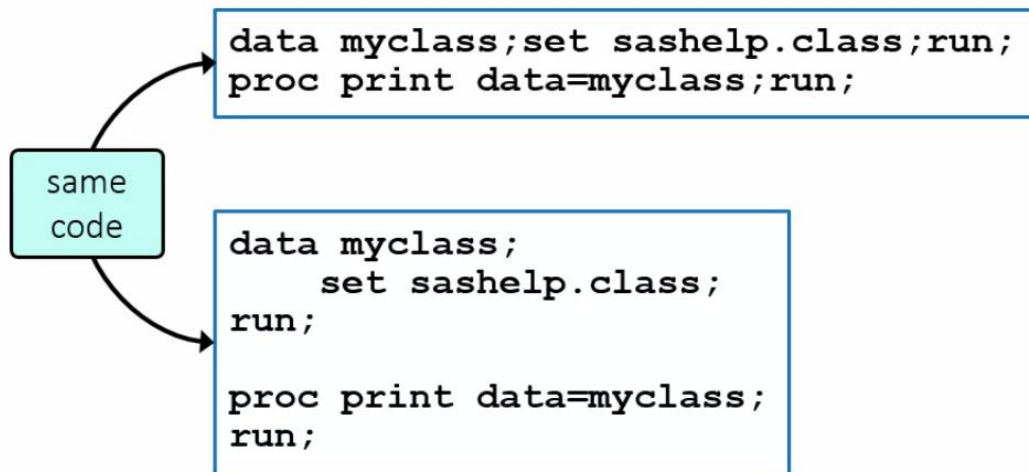
Data, set, and run are keywords.



These statements can be outside DATA and PROC steps,



**Global statements do not need a run statement after them.**



spacing doesn't matter to SAS,



```
data myclass;set sashelp.class;run;  
proc print data=myclass;run;
```

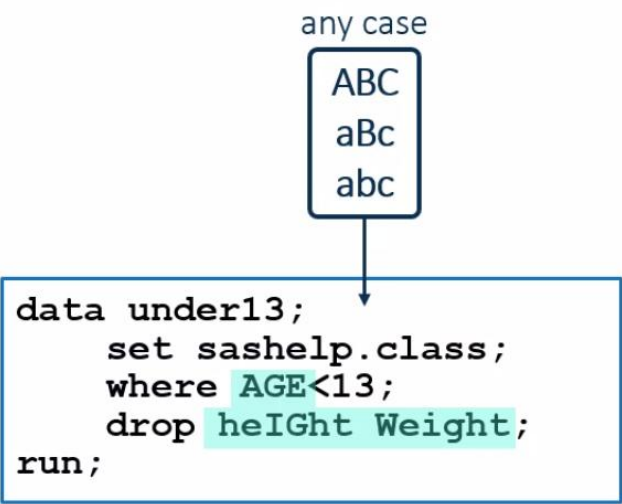


```
data myclass;             
    set sashelp.class;  
run;                             
                                          
proc print data=myclass;  
run;                           
```

There are also tools in your program editor that format code for you.

any case

ABC
aBc
abc

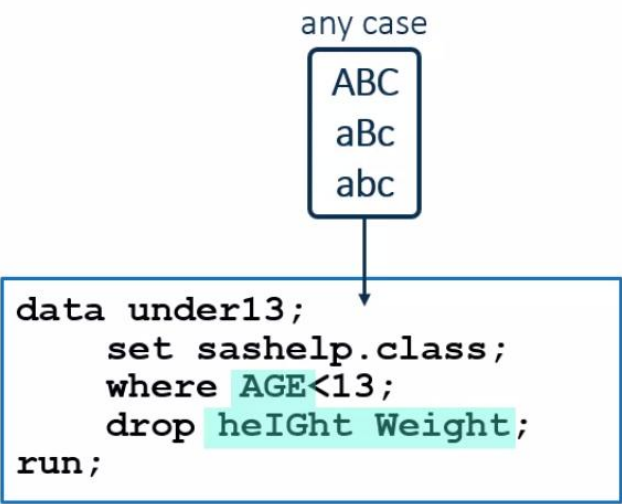


```
data under13;  
  set sashelp.class;  
  where AGE<13;  
  drop heIGht Weight;  
run;
```

You can type unquoted values such as columns,

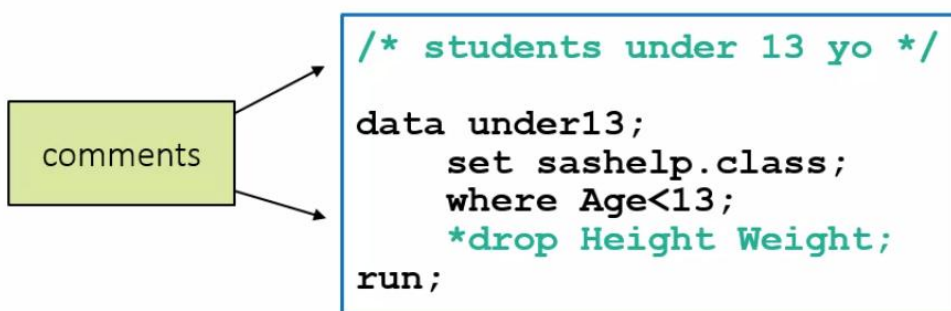
any case

ABC  
aBc  
abc

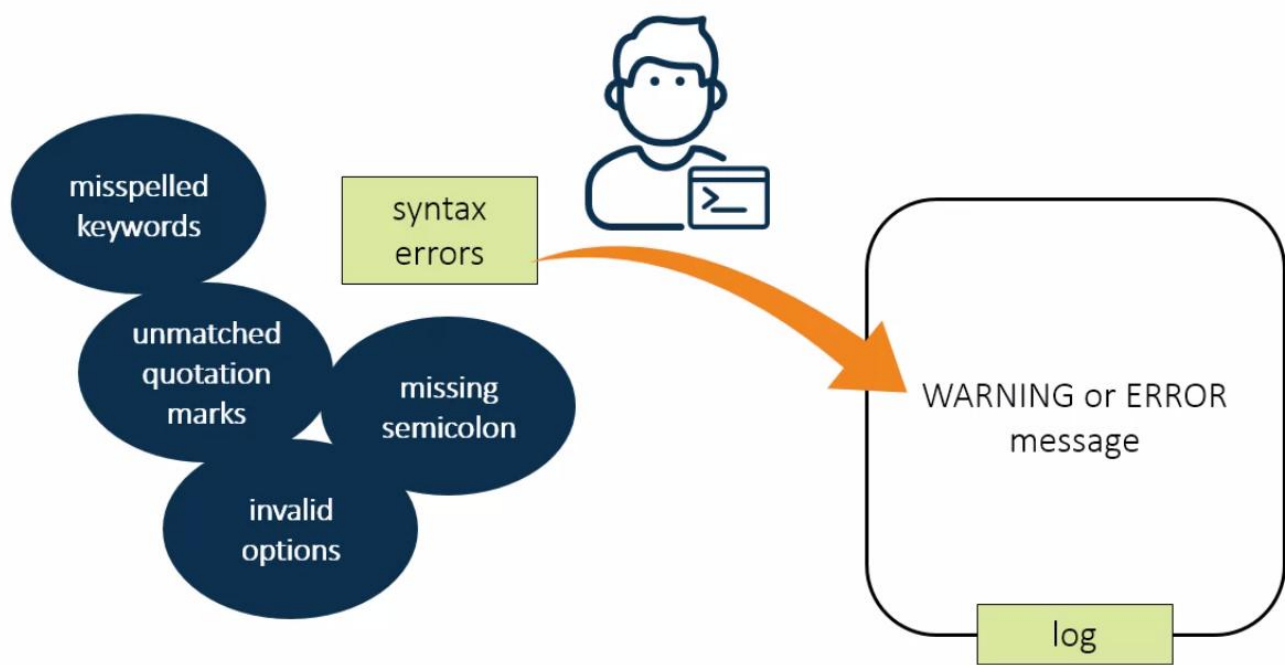


```
data under13;  
  set sashelp.class;  
  where AGE<13;  
  drop heIGht Weight;  
run;
```

table names or keywords in any case.



You can add comments to your code that are ignored when the program executes.



a warning or error message  
is written in the log.

## Module Summary: Essentials

---

### Running a SAS Program

- Programs can be submitted by clicking the **Run** icon or pressing the F3 key.
- To run a subset of a program, highlight the desired statements first. If you are using SAS Studio, click the **Run** icon or press F3. If you are using SAS Enterprise Guide, click the arrow next to **Run** and click **Run Selection** or press F3.
- A program can create a log, results, and output data.
- Submitting a program that has run previously in Enterprise Guide or SAS Studio replaces the log, output data, and results.
- Submitting a program that has run previously in the SAS windowing environment appends the log and results.

### Understanding SAS Syntax

- SAS programs consist of DATA and PROC steps, and each step consists of statements.

```
DATA ... ;  
    other statements  
RUN;
```

```
PROC ... ;  
    other statements  
RUN;
```

- Global statements are outside steps.

```
TITLE ... ;
```

```
OPTIONS ... ;
```

```
LIBNAME ... ;
```

- All statements end with a semicolon.
- Spacing doesn't matter in a SAS program.
- Unquoted values can be lowercase, upper case, or mixed case.

- Consistent program spacing is a good practice to make programs legible.
- Use the following automatic spacing features:

SAS Studio: Click the **Format Code** icon.

Enterprise Guide: Select **Edit > Format Code** or press Ctrl+I.

- Comments can be added to prevent text in the program from executing.
- Some common syntax errors are unmatched quotes, missing semicolons, misspelled keywords, and invalid options.
- Syntax errors might result in a warning or error in the log.
- Refer to the log to help diagnose and resolve syntax errors.