Chapter 2: SAS® Programs

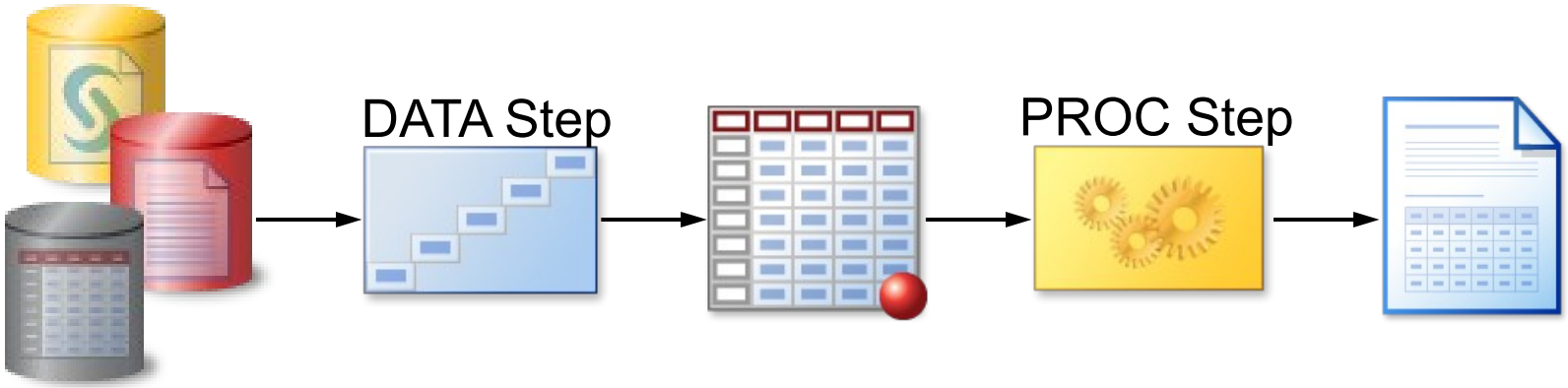
Chapter 2: SAS® Programs

Objectives

- List the components of a SAS program.

SAS Programs

A *SAS program* is a sequence of one or more steps.



- *DATA steps* typically create SAS data sets.
- *PROC steps* typically process SAS data sets to generate reports and graphs, and to manage data.

SAS Program Steps

A *step* is a sequence of SAS statements. This program has a DATA step and a PROC step.

```
data work.newemps;
   infile "&path\newemps.csv" dlm=',';
   input First $ Last $ Title $ Salary;
run;

proc print data=work.newemps;
run;
```

Step Boundaries

SAS steps begin with either of the following:

- a DATA statement
- a PROC statement

SAS detects the end of a step when it encounters one of the following:

- a RUN statement (for most steps)
- a QUIT statement (for some procedures)
- the beginning of another step (DATA statement or PROC statement)

2.01 Quiz

How many steps are in this program?

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```


p102d01

2.01 Quiz – Correct Answer

How many steps are in this program? **three**

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```

DATA Step

PROC Step

PROC Step

SAS Program Example

This DATA step creates a temporary SAS data set named **work.newsalesemps** by reading four fields from a file.

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```


p102d01

SAS Program Example

This PROC PRINT step lists the **work.newsalesemps** data set.

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```


p102d01

SAS Program Example

This PROC MEANS step summarizes the **Salary** variable in the **work.newsalesemps** data set.

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```


p102d01

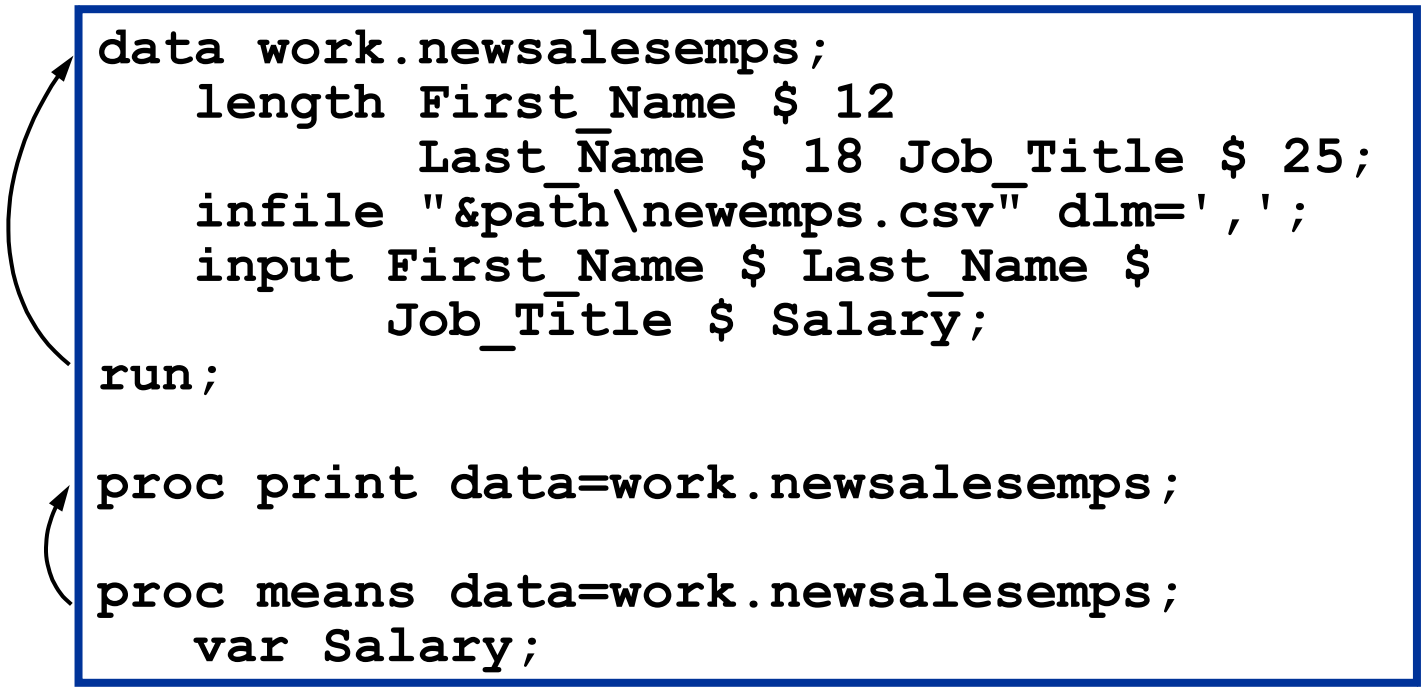
2.02 Quiz

How does SAS detect the end of each step in this program?

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;

proc means data=work.newsalesemps;
   var Salary;
```

2.02 Quiz – Correct Answer

How does SAS detect the end of each step in this program?



```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;

proc means data=work.newsalesemps;
   var Salary;
```

The DATA step ends at the RUN statement. The PROC PRINT step ends at the PROC MEANS statement. The end of the PROC MEANS step might or might not be detected. A RUN statement is recommended.

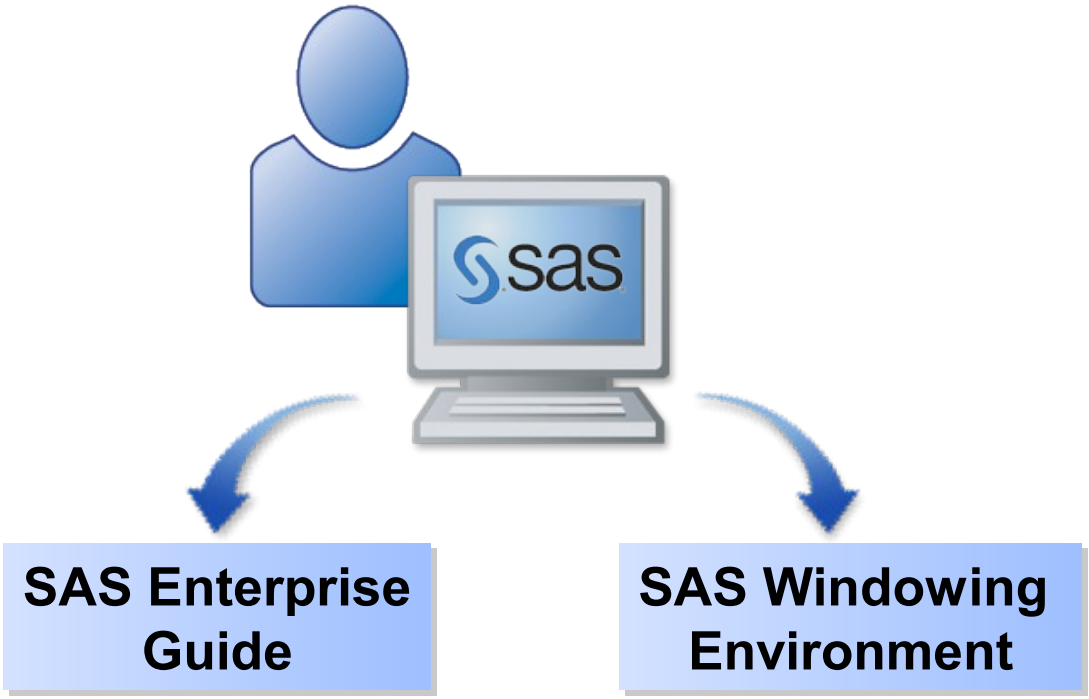
Chapter 2: SAS® Programs

Objectives

- Use SAS Enterprise Guide to open and submit a SAS program and browse the results.
- Use the SAS windowing environment to open and submit a SAS program and browse the results.

Business Scenario

Orion Star programmers will create and execute SAS programs and view results in an interactive environment. They must become familiar with both environments.

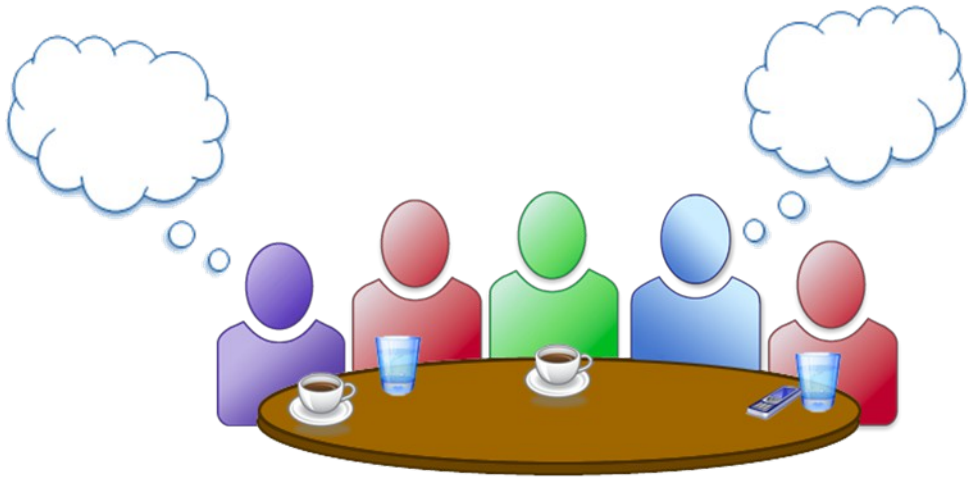


Submitting a SAS Program: SAS Enterprise Guide

This demonstration illustrates how to open and submit a SAS program and view the results using SAS Enterprise Guide.

Idea Exchange

Have you worked with SAS Enterprise Guide? If so, what do you like about it?



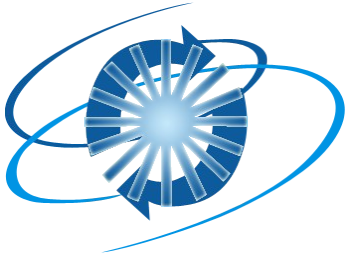
Submitting a SAS Program: SAS Windowing Environment

This demonstration illustrates how to open and submit a SAS program and view the results using the SAS windowing environment.

2.03 Multiple Choice Poll

Which environment (or environments) will you use during this course?

- SAS windowing environment
- SAS Enterprise Guide
- both the SAS windowing environment and SAS Enterprise Guide



Exercise

This exercise reinforces the concepts discussed previously.

Chapter 2: SAS® Programs

Objectives

- Identify the characteristics of SAS statements.
- Define SAS syntax rules.
- Document a program using comments.
- Diagnose and correct a program with errors.
- Save the corrected program.

Business Scenario

Well-formatted, clearly documented SAS programs are an industry best practice.

SAS Syntax Rules: Statements

SAS statements

- usually begin with an ***identifying keyword***
- always end with a ***semicolon***.

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```

p102d01

2.04 Quiz

How many statements make up this DATA step?

- one
- three
- five
- seven

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;
```

2.04 Quiz – Correct Answer

How many statements make up this DATA step?

- one
- three
- five
- seven

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;
```

SAS Program Structure

SAS code is free format.

```
data work.newsalesemps;
length First_Name $ 12
Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
Job_Title $ Salary;run;
proc print data=work.newsalesemps; run;
   proc means data  =work.newsalesemps;
var Salary;run;
```

This program is syntactically correct but difficult to read.

p102d02

SAS Program Structure

Rules for SAS Statements

- Statements can begin and end in any column.
- A single statement can span multiple lines.
- Several statements can appear on the same line.
- Unquoted values can be lowercase, uppercase, or mixed case.

unconventional formatting

```
data work.newsalesEmps;
length First_Name $ 12
Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
Job_Title $ Salary;run;
proc print data=work.newsalesemps; run;
   proc means data  =work.newsalesemps;
var Salary;run;
```

Recommended Formatting

- Begin each statement on a new line.
- Use white space to separate words and steps.
- Indent statements within a step.
- Indent continued lines in multi-line statements.

```
data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```

conventional formatting

Automatic Formatting with SAS Enterprise Guide

This demonstration illustrates the use of the auto-formatting tool in SAS Enterprise Guide.

Program Documentation

You can embed comments in a program as explanatory text.

```
  /* create a temporary data set, newsalesemps */
  /* from the text file newemps.csv            */

data work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   *read a comma delimited file;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;
```

/ comment */*

** comment statement ;*

SAS ignores comments during processing but writes them to the SAS log.

SAS Comments

This program contains four comments.

```
*-------------------------------------*
|   This program creates and uses the   |
|   data set called work.newsalesemps.  |
*-------------------------------------*;
data work.newsalesemps;
   length First_Name $ 12 Last_Name $ 18
          Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary /*numeric*/;
run;
  /*
proc print data=work.newsalesemps;
run;
  */
proc means data=work.newsalesemps;
   *var Salary;
run;
```

p102d03

2.05 Quiz

Open and examine **p102a01**. Based on the comments, which steps do you think will execute and what output will be generated?

Submit the program. Which steps are executed?

2.05 Quiz – Correct Answer

Open and examine **p102a01**. Based on the comments, which steps do you think will execute and what output will be generated?

Submit the program. Which steps are executed?

- **The DATA step executes and creates an output data set.**
- **The PROC PRINT step executes and produces a report.**
- **The PROC MEANS step is "commented out" and therefore does not execute.**

Business Scenario

Orion Star programmers must be able to identify and correct syntax errors in a SAS program.

Syntax Errors

A *syntax error* is an error in the spelling or grammar of a SAS statement. SAS finds syntax errors as it compiles each SAS statement, before execution begins.

Examples of syntax errors:

- misspelled keywords
- unmatched quotation marks
- missing semicolons
- invalid options

2.06 Quiz

This program includes three syntax errors. One is an invalid option. What are the other two syntax errors?

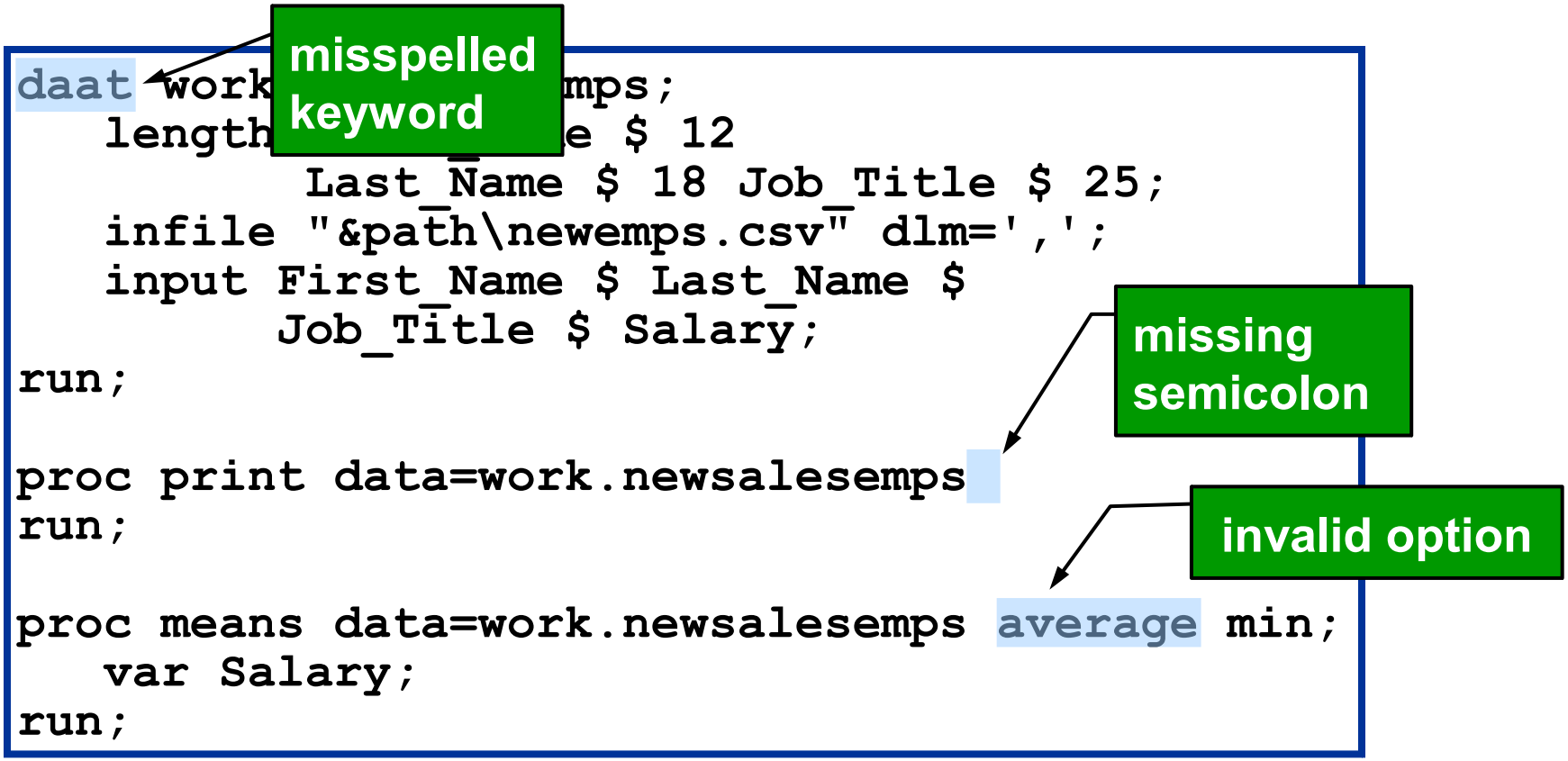
```
daat work.newsalesemps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
   var Salary;
run;
```

invalid option

p102d04

2.06 Quiz – Correct Answer

This program includes three syntax errors. One is an invalid option. What are the other two syntax errors?




```
daat work                mps;
   length                e $ 12
                Last_Name $ 18 Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
   var Salary;
run;
```

p102d04

Syntax Errors

The Enhanced Editor in SAS and the Program Editor in SAS Enterprise Guide use the color red to indicate a potential error in your SAS code.

```
daat work.newsalesemps;
    length First_Name $ 12
           Last_Name $ 18 Job_Title $ 25;
     infile "&path\newemps.csv" dlm=',';
     input First_Name $ Last_Name $
           Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
  var Salary;
run;
```

Syntax Errors

When SAS encounters a syntax error, it writes a warning or error message to the log.

```
ERROR 22-322: Syntax error, expecting one of the following:
              a name, a quoted string, (, /, ;, _DATA_, _LAST_,
              _NULL_.
```

```
WARNING: Data set WORK.TEST was not replaced because this step was stopped.
```

You should always check the log to make sure that the program ran successfully, even if output is generated.

Diagnosing and Correcting Syntax Errors

This demonstration illustrates how to diagnose and correct syntax errors and resubmit and save the corrected program.

p102d04

2.07 Quiz

What is the syntax error in this program?

```
data work.newsalesemps;
   length First_Name $ 12 Last_Name $ 18
          Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',;
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```

p102d05

2.07 Quiz – Correct Answer

What is the syntax error in this program?

```
data work.newsalesemps;
   length First_Name $ 12 Last_Name $ 18
          Job_Title $ 25;
   infile "&path\newemps.csv" dlm=',;
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
   var Salary;
run;
```

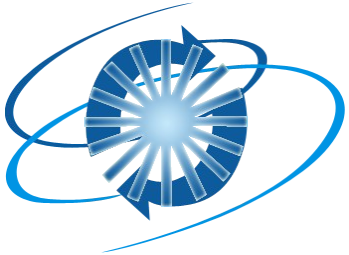
The program contains unbalanced quotation marks in the DLM= option in the INFILE statement.

p102d05

Correcting Unbalanced Quotation Marks

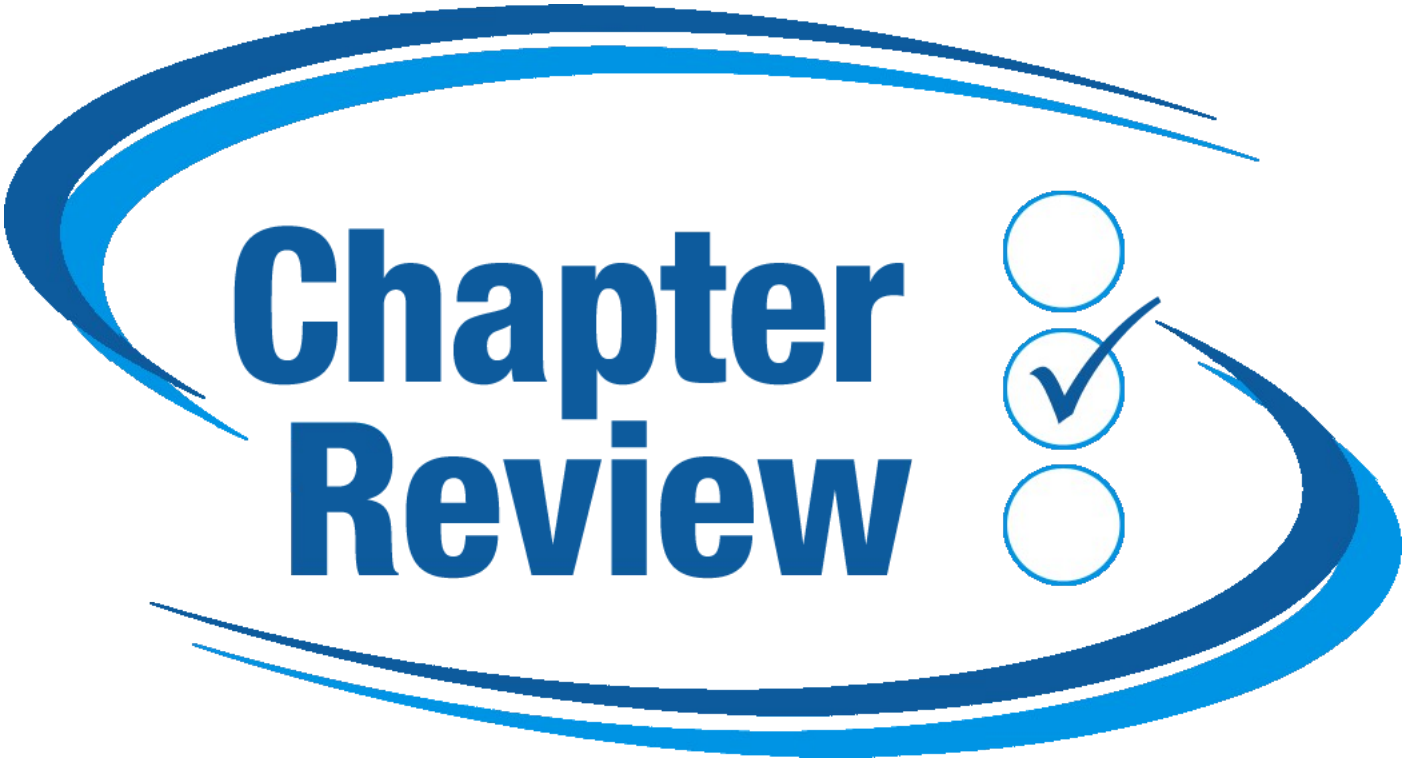
This demonstration illustrates the behavior of SAS and Enterprise Guide when a program contains unbalanced quotation marks.

p102d05



Exercise

This exercise reinforces the concepts discussed previously.



1. How many step boundaries does this program contain?
   - four
   - five
   - six
   - seve

```
data work.staff;
   length First_Name $ 12
          Last_Name $ 18
          Job_Title $ 25;
   infile "&path\newemployees.csv"
           dlm=',';
   input First_Name $ Last_Name$
         Job_Title $ Salary;
run;

proc print data=work.staff;
run;

proc means data=work.staff;
   var Salary;
run;
```

1. How many step boundaries does this program contain?

   - four
   - five
   - six
   - seven

```
data work.staff;
   length First_Name $ 12
          Last_Name $ 18
          Job_Title $ 25;
   infile "&path\newemployees.csv"
           dlm=',';
   input First_Name $ Last_Name$
         Job_Title $ Salary;
run;

proc print data=work.staff;
run;

proc means data=work.staff;
   var Salary;
run;
```

2. Which of the following is a SAS syntax requirement?

   - Begin each statement in column one.
   - Put only one statement on each line.
   - Separate each step with a line space.
   - End each statement with a semicolon.
   - Put a RUN statement after every DATA or PROC step.

2. Which of the following is a SAS syntax requirement?

   - Begin each statement in column one.
   - Put only one statement on each line.
   - Separate each step with a line space.
   - End each statement with a semicolon.
   - Put a RUN statement after every DATA or PROC step.

3. Which of the following steps is typically used to generate reports and graphs?

   - DATA
   - PROC
   - REPORT
   - RUN

3. Which of the following steps is typically used to generate reports and graphs?

   - DATA
   - PROC
   - REPORT
   - RUN

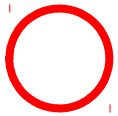
4. Does this comment contain syntax errors?

```
  /*
Report created for budget
presentation; revised October 15.
  */
proc print data=work.newloan;
run;
```

- No. The comment is correctly specified.
- Yes. Every comment line must end with a semicolon.
- Yes. The comment text incorrectly begins on line one.
- Yes. The comment contains a semicolon, which causes an error message.

4. Does this comment contain syntax errors?

```
  /*
Report created for budget
presentation; revised October 15.
  */
proc print data=work.newloan;
run;
```



- No. The comment is correctly specified.
- Yes. Every comment line must end with a semicolon.
- Yes. The comment text incorrectly begins on line one.
- Yes. The comment contains a semicolon, which causes an error message.

5. What result would you expect from submitting this step?

```
proc print data=work.newsalesemps
run;
```

- an HTML report of the **work.newsalesemps** data set
- an error message in the log
- a LISTING report of the **work.newsalesemps** data set
- the creation of the temporary data set **work.newsalesemps**

5. What result would you expect from submitting this step?

```
proc print data=work.newsalesemps
run;
```

- an HTML report of the **work.newsalesemps** data set
- an error message in the log
- a LISTING report of the **work.newsalesemps** data set
- the creation of the temporary data set **work.newsalesemps**

6. If you submit a program containing unbalanced quotation marks in SAS Enterprise Guide, you can simply correct the error and resubmit the program.
   - True
   - False

6. If you submit a program containing unbalanced quotation marks in SAS Enterprise Guide, you can simply correct the error and resubmit the program.

   - True
   - False

7. What happens if you submit the following program?

```
porc print data=work.newsalesemps;
run;
```

- SAS does not execute the step.
- SAS assumes that the keyword PROC is misspelled and executes the PROC PRINT step.

7. What happens if you submit the following program?

```
porc print data=work.newsalesemps;
run;
```

- SAS does not execute the step.
- SAS assumes that the keyword PROC is misspelled and executes the PROC PRINT step.

8. Suppose you submit a short, simple DATA step. If the active window displays the message **DATA step running** for a long time, what probably happened?

   - You misspelled a keyword.
   - You forgot to end the DATA step with a RUN statement.
   - You specified an invalid data set option.
   - Some data values were not appropriate for the SAS statements that you specified.

8. Suppose you submit a short, simple DATA step. If the active window displays the message **DATA step running** for a long time, what probably happened?

   - You misspelled a keyword.
   - You forgot to end the DATA step with a RUN statement.
   - You specified an invalid data set option.
   - Some data values were not appropriate for the SAS statements that you specified.

9. Suppose that this program contains no errors. What happens when you submit the program in the SAS windowing environment?

```
proc print data=work.sales;
run;
```

- Messages appear in the Log window, and the Explorer window moves to the front.
- Any HTML output appears in the Output window or a browser window, and any LISTING output appears in the Results Viewer window.
- Any HTML output appears in the Results Viewer window or a browser window, and any LISTING output appears in the Output window.

9. Suppose that this program contains no errors. What happens when you submit the program in the SAS windowing environment?

```
proc print data=work.sales;
run;
```

- Messages appear in the Log window, and the Explorer window moves to the front.
- Any HTML output appears in the Output window or a browser window, and any LISTING output appears in the Results Viewer window.
- Any HTML output appears in the Results Viewer window or a browser window, and any LISTING output appears in the Output window.

10. Which of the following is a true statement about SAS output?

- SAS Enterprise Guide 5.1 displays text or LISTING output by default.
- SAS Enterprise Guide 5.1 displays HTML output by default.
- SAS 9.3 in the windowing environment displays LISTING output by default.
- SAS 9.3 in the windowing environment displays HTML output by default.

10. Which of the following is a true statement about SAS output?

- SAS Enterprise Guide 5.1 displays text or LISTING output by default.
- SAS Enterprise Guide 5.1 displays HTML output by default.
- SAS 9.3 in the windowing environment displays LISTING output by default.
- SAS 9.3 in the windowing environment displays HTML output by default.