Print

# Summary of Lesson 2: Working with SAS Programs

This summary contains topic summaries, syntax, and sample programs.

## Topic Summaries

*To go to the movie where you learned a task or concept, select a link.*

### Exploring SAS Programs

A SAS program consists of DATA steps and PROC steps. A SAS programming step is comprised of a sequence of statements. Every step has a beginning and ending step boundary. SAS compiles and executes each step independently, based on the step boundaries.

A SAS program can also contain global statements, which are outside DATA and PROC steps, and typically affect the SAS session. A TITLE statement is a global statement. After it is defined, a title is displayed on every report, unless the title is cleared or canceled.

SAS statements usually begin with an identifying keyword, and always end with a semicolon. SAS statements are free format and can begin and end in any column. A single statement can span multiple lines, and there can be more than one statement per line. Unquoted values can be lowercase, uppercase, or mixed case. This flexibility can result in programs that are difficult to read.

Conventional formatting, also called *structured formatting*, uses consistent spacing to make a SAS program easy to read. To follow best practices, begin each statement on a new line, indent statements within each step, and indent subsequent lines in a multi-line statement.

Comments are used to document a program and to mark SAS code as non-executing text. There are two types of comments: *block comments* and *comment statements*.

| |
|---|
| */* comment *\/* |
| * comment statement; |

### Diagnosing and Correcting Syntax Errors

Syntax errors occur when program statements do not conform to the rules of the SAS language. Common syntax errors include misspelled keywords, missing semicolons, and invalid options. SAS finds syntax errors during the compilation phase, before it executes the program. When SAS encounters a syntax error, it writes the following to the log: the word ERROR or WARNING, the location of the error, and an explanation of the error. You should always check the log, even if the program produces output.

Mismatched or unbalanced quotation marks are considered a syntax error. In some programming environments, this results in a simple error message. In other environments, it is more difficult to identify this type of error.

## Sample Programs

### Submitting a SAS Program

```
data work.newsalesemps;
   set orion.sales;
   where Country='AU';
run;

title 'New Sales Employees';

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

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

title;
```

### Adding Comments to Your SAS Programs

```
*This program uses the data set orion.sales to create work.newsalesemps.;
data work.newsalesemps;
   set orion.sales;
   where Country='US';
run;

 /*
proc print data=work.newsalesemps;
run;*/
proc means data=work.newsalesemps;
   class Gender;
   var Salary/*numeric variable*/;
run;
```

### Viewing and Correcting 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 max;
   class Job_Title;
   var Salary;
run;
```



Close