

DATA Step

How do I add comments to code in SAS?

```
/******  
*****/  
/**  
    ***/  
/**          Some comments on comments  
    ***/  
/**  
    ***/  
/******  
*****/
```

```
*** Using the Box Comment style with the slash-  
asterisk pairings is          *** ;  
*** useful for sub-  
dividing code and highlighting significant areas          *** ;
```

```
*** Using the Comment Statement is to be recommended for process comme  
ntary.          *** ;
```

```
/* Comment out this block
```

```
*** If there is a need to block out larger sections of code the slash-  
asterisk *** ;  
*** pairing can then be wrapped around syntax (including comments).  
    *** ;
```

```
data _null_ ;  
run ;
```

```
*/
```

```
*** If however the slash-  
asterisk pairing has been used for inline comments          *** ;  
*** e.g. within an SQL Query and there is a need to comment out a code  
section *** ;  
*** use a macro program definition without a call to remove code secti  
ons          *** ;
```

```
%macro donotrun ;
```

```
*** Process Commentary goes here *** ;  
proc sql ;  
    select a  
        , b /* Inline commentary goes here */
```

DATA Step

```
        ,c  
    from tablename  
    ;  
quit ;
```

```
%mend donotrun ;
```

```
*** Macro Program definitions cause the syntax colouring in the Enhanc  
ed *** ;  
*** Editor to stop working. Use a dummy local macro variable to 'fool  
' the *** ;  
*** Enhanced Editor into reinstating syntax colouring  
*** ;
```

```
%macro donotrun ;
```

```
%local colourit ;  
%let colourit = %nrstr(%mend donotrun ;) ;
```

```
*** Process Commentary goes here *** ;  
proc sql ;  
    select  a  
           ,b /* Inline commentary goes here */  
           ,c  
    from tablename  
    ;  
quit ;
```

```
%mend donotrun ;
```

```
*** In long sections of code where there is a need to add / remove com  
ment *** ;  
*** blocks at multiple points within the code, consider using macro va  
riables *** ;  
*** to control the inclusion or exclusion.  
*** ;
```

```
/* Alternate these macro variable definitions to include code: ;
```

```
%let star  = ;  
%let slash = ;
```

```
/* With these to exclude code: ;
```

```
%let star  = * ;  
%let slash = /;
```

DATA Step

```
&slash&star
```

```
data _null_ ;  
run ;
```

```
proc sql ;  
    select *  
    from table  
    ;  
quit ;
```

```
*/ ;
```

*** Macro Comments should be reserved for sections of code acted upon by the *** ;

*** Macro Processor e.g. macro logic in macro program definitions.
*** ;

```
%macro runit ;  
    %*create local macro variables ;  
    %local a b c ;
```

```
    *** Non-macro logic statements *** ;  
    data _null_ ;  
    run ;  
%mend runit ;
```

```
%runit
```

Unique solution ID: #1001

Author: Alan D Rudland

Last update: 2017-02-10 10:14