

SAS LIBNAME Statement
LIBNAME *libref*'*SAS-data-library*'<*options*>;

Browsing a SAS Library
PROC CONTENTS *DATA=libref.* *_ALL_* <**NOODS**>;
RUN;

Creating a SAS Data Set
PROC IMPORT *DATAFILE=* "filename" or *TABLE=* "tablename"
OUT=<*libref.SAS-data-set*>
 <*DBMS=identifier*>
 <*SAS-data-set-options*>
 <**REPLACE**>;
 <**GETNAMES=** yes | no>;
 <**GUESSINGROWS=***number-of-rows*>;
 <**DATAROW=** *row-number*>;
RUN;

Reading a SAS Data Set
DATA *output-SAS-data-set(s)*;
SET *input-SAS-data-set*;
 <...*more SAS statements*...>;
RUN;

Accessing Microsoft Excel Data with the XLSX Engine
LIBNAME <*libref*>**XLSX** <*physical-path-and-filename.xlsx*><*options*>;

Selecting Variables – Data Set Options
(DROP=*variables***)**
(KEEP=*variables***)**
(RENAME=*old-variable-name=new-variable-name***)**

Selecting Observations – Data Set Options
(IN=*variable***)**

Examining SAS Data Set Descriptor Portion
PROC CONTENTS *DATA=SAS-data-set*;
RUN;

Error Handling
PUT *specification(s)*;
PUTLOG 'message';

Creating New Variables
new-variable=expression;
LENGTH *variables* <*\$*> *length* ...;

Modifying Variables
RETAIN *variable* <*initial-value*>;

Subsetting Data
IF *expression* **THEN DELETE**;
WHERE *expression*;

Subsetting Variables
DROP *variable(s)*;
KEEP *variable(s)*;

Conditional Processing
IF *expression* **THEN** *statement*;
ELSE *statement*;

DO;
SAS statements;
END;

DO *index-variable=specification-1* < , ...*specification-n*>;
 ...*more SAS statements*...
END;

DO UNTIL(*expression*);
 ...*more SAS statements*...
END;

DO WHILE(*expression*);
 ...*more SAS statements*...
END;

Operators in Expressions

Operator	Meaning
= or eq	equal to
^= or ne	not equal to
> or gt	greater than
< or lt	less than
>= or ge	greater than or equal to
<= or le	less than or equal to
AND or &	and, both. If both expressions are true, then the compound expression is true.
OR or	or, either. If neither expression is true, then the compound expression is true.
0	false
1	true
.	false

Applying Formats
FORMAT *variable(s)* *format-name*...;

Creating User-Defined Formats
PROC FORMAT *LIBRARY=libref* or
LIBRARY=FMTLIB;
VALUE *format-name*;
RUN;

Titles and Footnotes
TITLE <*n*> 'text';
FOOTNOTE <*n*> 'text';

Labels
LABEL *variable1='label1'* *variable2='label2'*...;

SAS Base Programming for SAS®9 and SAS® 9.4 Syntax Reference Guide

Copyright: September 4, 2018 Last Modified: February 8, 2019

Combining SAS Data Sets

One-to-One Reading:

```
DATA output-SAS-data-set;  
  SET SAS-data-set-1;  
  SET SAS-data-set-2;  
RUN;
```

RUN;

Concatenating:

```
DATA output-SAS-data-set;  
  SET SAS-data-set-1 SAS-data-set-2;  
RUN;
```

RUN;

Match-Merging:

```
DATA output-SAS-data-set;  
  MERGE SAS-data-set-1 SAS-data-set-2;  
  BY <DESCENDING> variable(s);  
RUN;
```

Sorting Data

```
PROC SORT DATA=SAS-data-set  
  <OUT=SAS-data-set>;  
  BY <DESCENDING> BY-variables(s);  
RUN;
```

Producing Detail Reports

```
PROC PRINT DATA=SAS-data-set  
  <NOOBS LABEL SPLIT='split-character'>;  
  VAR variable(s);  
  SUM variable(s);  
  BY variable(s);  
  ID variable(s);  
RUN;
```

Data Summarization and Validation Procedures

```
PROC MEANS DATA=SAS-data-set  
  <keyword-statistics> <options>;  
  CLASS variable(s);  
  VAR variable(s);  
RUN;
```

```
PROC FREQ DATA=SAS-data-set <options>;  
  TABLES variable(s) <LIST> <CROSSLIST>  
    <NOFREQ>;  
  TABLES variable1* variable2 <LIST>  
    <CROSSLIST> <NOFREQ>;  
  <NOPERCENT> <NOROW> <NOCOL>;  
  <COMPRESS>;  
  <FORMCHAR (1,2,7) = 'formchar-string'>;  
  <NLEVELS>;  
  <NOPRINT>;  
  <ORDER=DATA|FORMATTED|FREQ|INTERNAL>;  
  <PAGE>;  
RUN;
```

Creating Output

```
ODS open-destination;  
ODS close-destination CLOSE;  
ODS EXCEL <(<ID=>identifier)> <action>;  
ODS EXCEL <(<ID=>identifier)> <option(s)>;
```

ODS HTML

```
BODY = body-file-specification;  
CONTENTS=contents-file-specification  
FRAME=frame-file-specification;  
ODS HTML CLOSE;
```

```
ODS PDF <(<ID=>identifier)> <action>;
```

```
ODS RTF <(<ID=>identifier)> <action>;
```

Functions

```
function-name (argument-1(<, argument-n>);  
CATX (separator, string-1<,...string-n>)  
CEIL(argument)  
COMPBL(source)  
COMPRESS (source<, characters><[, modifier(s)>])  
DAY(date)  
DATDIF (start_date, end_date, basis)  
DATE ()  
FIND (string, substring<[, modifiers><[, startpos>])  
FLOOR(argument)  
INDEX (source, excerpt)  
INT(argument)  
INTCK ('interval', from, to)  
INTNX ('interval', start-from, increment, <'alignment'>)  
LEFT(argument)  
LOWCASE(argument)  
MDY (month, day, year)  
MONTH(date)  
PROPCASE (argument<[, delimiter(s)>])  
QTR(date)  
RIGHT(argument)  
ROUND (argument, round-off-unit)  
SCAN (argument, n<[, delimiters>>)  
SUBSTR (argument, position <[, n>])  
TIME ()  
TODAY ()  
TODAYDATE ()  
TRANWRD (source, target, replacement)  
TRIM(argument)  
UPCASE(argument)  
WEEKDAY(date)  
YEAR(date)  
YRDIF (start_date, end_date, basis)
```