

Simple SAS Program;

```
data work.table;

input Team$ 1-17 Wins Draws Losses Points;

Games_Played=Wins+Draws+Losses;

if points>=75 then Europe="Y";

else Europe="N";

if wins<20 then Squad="Bad";

else Squad="Pro";

datalines;

Manchester United 28 6 4 90

Arsenal      26 7 5 85

Chelsea      25 8 5 83

Tottenham Hotspur 24 10 4 82

Manchester City  23 8 7 77

Everton      22 7 9 73

Leicester City  21 7 10 70

Swansea City   20 8 10 68

Sunderland    18 12 8 66

Liverpool     15 13 10 58

;

run;

proc print data=work.table;

run;
```

WHERE Statement;

```
data cars;  
  
set sashelp.cars;  
  
where (or if) make in ('Acura','BMW','Audi')  
  
and horsepower >= 150;  
  
run;  
  
proc print data=cars;  
  
run;
```

```
data work.cars2;  
  
set sashelp.cars;  
  
where (or if) type='Hybrid' or origin='Europe';  
  
run;  
  
proc print data=work.cars2;  
  
run;
```

PROC CONTENTS and DATASETS;

```
proc contents data=sashelp.fish varnum; run;  
  
proc contents data=sashelp._all_ nods; run;  
  
proc contents data=work._all_ nods; run; OR proc contents data=_all_ nods; run;  
  
proc datasets;  
  
contents data=sashelp.baseball;
```

Proc REPORT and proc PRINT;

```
proc report data=sashelp.baseball;  
columns name team position salary;  
run;
```

```
proc print data=sashelp.baseball noobs;  
var name team position salary;  
where salary >= 250;  
run;
```

```
proc print data=sashelp.baseball label;  
var nAtBat nHits nAssts;  
label nAtBat='Bat'; label nHits='Hits'; label nAssts='Assists';  
run;
```

Proc FREQ;

```
proc freq data=sashelp.cars;  
where msrp >25000;  
tables make; run;
```

```
proc freq data=sashelp.cars;  
where msrp >25000;  
tables make / nocum; run;
```

```
proc freq data=sashelp.cars;

tables make*type / list nopercnt;

run;

proc freq data=sashelp.cars;

tables make*type*drivetrain / crosslist norow nocol (nofreq nopercnt);

run;
```

Proc TABULATE;

title1 'Different'; *(title1; to cancel the title in effect. Similar for title2; and title3; / title; to cancel all titles;) {also called null title; statement;}*

```
title2 'Fish';

title3 'Types';

proc tabulate data=sashelp.fish;

keylabel all='Total';

class species;

var weight height width;

table species*(weight height width), mean n median all;

run;
```

Proc SORT;

```
proc sort data=sashelp.cars out=work.cars;

by drivetrain make descending horsepower EngineSize Length Origin Type;

where Cylinders >4;

run;
```

```
footnote "World Car Data";
```

```
proc print data=work.cars;
```

```
var horsepower EngineSize Length Origin Type;
```

```
sum Invoice MSRP;
```

```
by drivetrain make; (+ id make);
```

```
pageby make; run;
```

```
footnote; (to cancel the footnote in this or all footnotes in general;) {also called null footnote;}
```

```
proc tabulate data=work.cars;
```

```
class make;
```

```
var MPG_City MPG_Highway;
```

```
table make*(MPG_City MPG_Highway), mean n;
```

```
run;
```

```
proc sort data=sashelp.baseball out=team;
```

```
by team descending nAtBat;
```

```
run;
```

```
proc print data=team label;
```

```
by team;
```

```
var nAtBat nHits nAssts;
```

```
label nAtBat='Bat'; label nHits='Hits'; label nAssts='Assists';
```

```
run;
```

FORMAT statement;

```
proc print data=sashelp.cars;  
  
format msrp dollar10.2 invoice comma9. origin $8. weight 5.;  
  
run;
```

Proc FORMAT;

```
libname newlib '/folders/myfolders/newfolder';  
  
proc format library=newlib;  (or lib=newlib)  
  
value epl  
  
90='Champions'  
  
86='Runner Up'  
  
82='UCL'  
  
79='UCL Qual'  
  
76='Europa'  
  
;  
  
run;
```

```
libname newlib '/folders/myfolders/newfolder';  
  
data newlib.table;  
  
infile '/folders/myfolders/NewFolder/LeagueTable.txt' firstobs=3;  
  
input Position Team$ 10-26 Wins Draws Losses Total;  
  
format total epl.;  
  
run;
```

```
proc print data=newlib.table noobs label;  
label total='Points at the end of the season';  
run;
```

Proc MEANS;

```
proc means data=sashelp.cars min max mode range  
clm cv css nmiss skewness std var n maxdec=0;  
run;
```

```
proc means data=sashelp.cars maxdec=0  
median p1 p5 p10 q1 q3 qrange;  
run;
```

```
proc means data=sashelp.fish maxdec=2;  
var weight height width;  
class species;  
run;
```

```
proc sort data=sashelp.fish out=work.fish;  
by species;  
run;  
proc means data=fish maxdec=2;  
by species;  
run;
```

```
proc means data=sashelp.fish noprint;
var weight height width;
class species;
output out=work.fish2
mean= weight height width
min= length1 length2
max= length3;
run;
```

Proc SUMMARY;

```
proc summary data=sashelp.fish print;
var weight height width;
class species;
output out=work.fish2
mean= weight height width
min= length1 length2
max= length3;
run;
```

MIXED Program;

```
data newlib.revenue;
set sashelp.electric;
keep customer year revenue totalrev;
if not (year=2005);
totalrev+revenue;
format totalrev dollar5.;  run;
```


IF-THEN/ELSE statement;

```
data newlib.revnew (keep= customer revenue year Totalrev Remark);

set sashelp.electric;

if revenue<10 then delete;

if year not in(2002 2003);

TotalRev=revenue;

length Remark $7;

if revenue >=100 then Remark='Better';

else if 50<revenue<100 then Remark='Good';

else if revenue <=50 then Remark='Average';

label TotalRev='Cumulative Revenue';

format Totalrev dollar5.;

run;


proc print data=newlib.revnew label;

run;
```

POINT;

```
data boots;

obsnum=25;

set sashelp.shoes point=obsnum;

output;  (without output the program will not produce any output)

stop;

run;
```

SELECT-WHEN/OTHERWISE statement;

```
data newlib.cars;  
  
set sashelp.cars;  
  
select;  
  
when (make='Acura') class='semiluxury';  
  
when (make='Audi') class='premium';  
  
when (make='BMW') class='luxury';  
  
otherwise class='standard';  
  
end;  run;
```

```
data newlib.cars;  
  
set sashelp.cars;  
  
select (make);  
  
when ('Acura') class='semiluxury';  
  
when ('Audi') class='premium';  
  
when ('BMW') class='luxury';  
  
otherwise class='standard';  
  
end;  run;
```

FIRST. & LAST. ;

```
proc sort data=sashelp.shoes out=shoes;  
  
by product region;  
  
run;
```

```
data newlib.shoes;

set shoes;

by product region;

if first.region then NewShoe=0;

if last.region then NewShoe=1;

run;
```

LAST option;

```
data newlib.shoes;

set sashelp.shoes end=last;

Difference=Inventory-Sales;

format difference dollar9.;

if last;

run;
```

MERGE statements; Sas data sets –

```
data newlib.patients;

input obs id sex $ age;

datalines;

1 156 m 29

3 458 f 69

5 862 m 42

7 445 m 78

9 163 f 45

;    run;
```

```
data newlib.measure;  
  
input obs id ht wt;  
  
datalines;  
  
2 156 35 53  
  
4 458 63 69  
  
6 862 55 42  
  
8 445 89 94  
  
;  
  
run;
```

One-one merging;

```
data newlib.patientdata;  
  
set newlib.patients;  
  
set newlib.measure;  
  
run;
```

Concatenating;

```
data newlib.datapatients;  
  
set newlib.measure newlib.patients;  
  
run;
```

APPEND;

```
data work.patients;
```

```
input obs id;
```

```
datalines;
```

```
1 156
```

```
3 458
```

```
5 862
```

```
7 445
```

```
9 163
```

```
;
```

```
run;
```

```
data work.measure;
```

```
input obs id ;
```

```
datalines;
```

```
2 156
```

```
4 458
```

```
6 862
```

```
8 445
```

```
;
```

```
run;
```

```
proc append base=work.patients data=work.measure;
```

```
run;
```

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

FORCE;

```
proc append base=newlib.patients data=newlib.measure force;  
  
run;  
  
proc print data=newlib.patients;  
  
run;
```

Interleaving;

```
data newlib.patcomb;  
  
set newlib.patients newlib.measure;  
  
by obs;  run;
```

Match MERGE;

```
data newlib.patients1;  
  
input obs id sex $ age;  
  
datalines;  
  
1 156 m 29  
  
3 458 f 69  
  
5 862 m 42  
  
7 445 m 78  
  
9 163 f 45  
  
;  
  
run;
```

```
data newlib.measure1;
```

```
input obs id ht wt;
```

```
datalines;
```

```
1 156 35 53
```

```
3 458 63 69
```

```
6 862 55 42
```

```
8 445 89 94
```

```
;
```

```
run;
```

```
data newlib.patdata;
```

```
merge newlib.patients1 newlib.measure1;
```

```
by obs; run;
```

Import an EXCEL file;

```
libname newlib '/folders/myfolders/newfolder';
```

```
proc import datafile='/folders/myfolders/newfolder/SAS_Data.xlsx'
```

```
out=newlib.testdata
```

```
dbms=xlsx (sheet 1 imported by default)
```

```
replace;
```

```
run;
```

```
proc import datafile='/folders/myfolders/newfolder/SAS_Data.xlsx'  
out=newlib.datatest  
dbms=xlsx  
replace;  
sheet="Sheet2"; (to import sheet 2)  
run;
```

OR

```
proc import datafile='/folders/myfolders/newfolder/SAS_Data.xlsx'  
out=newlib.testdata  
dbms=xls (to import the entire excel file)  
replace;  
run;
```

RENAME;

```
data work.merger;  
merge newlib.datatest (rename=(Date=VisitDate))  
newlib.testdata (rename=(Date=BirthDate));  
by id;  
run;
```


Merging only the common variables;

```
data work.merged;  
merge newlib.datatest (in=new1 rename=(Date=VisitDate))  
      newlib.testdata (in=new2 rename=(Date=BirthDate));  
by id;  
if new1=1 and new2=1; (if new1 and new2)  
run;
```

DO Loop;

```
data multiply;  
answer=0;  
do i=1 to 25;  
answer+10;  
end;  
run;
```

```
data multiply;  
answer=0;  
do i=1,2,3,4;  
answer+25;  
end;  
run;
```

```
data multiply;  
multiple=0;  
do i=10 to 50 by 5;  
multiple+25;  
output;  
end;  
run;
```

```
data sub;  
value=0;  
do i=20 to 1 by -2;  
value+2;  
output;  
end;  
run;
```

```
data work.earn;  
capital=2000;  
do month=1 to 12;  
interest=capital*(.075/12);  
capital+interest;  
output;  
end;  
run;
```

```
data invest (drop=i);  
input type$ 1-7 rate months;  
investment=5000;  
do i=1 to months;  
investment+(rate/12)*investment;  
end;  
format investment dollar8.2;  
datalines;  
03month 0.019 03  
06month 0.022 06  
09month 0.023 09  
12month 0.024 12  
18month 0.025 18  
24month 0.026 24  
;  
run;
```

NESTED DO Loop;

```
data factor;  
do i=10 to 50 by 10;  
  do j=2 to 10 by 2;  
    output;  
  end;  
end;  
run;
```

```
data work.earn;  
  
do year=1 to 5;  
  
    capital+1000;  
  
    do month=1 to 12;  
  
        interest=capital*(.075/12);  
  
        capital+interest;  
  
    output;  
  
end;  
  
end;  
  
run;
```

DO UNTIL;

```
data work.invest;  
  
do until(capital>=50000);  
  
    capital+2500;  
  
    capital+capital*(.10);  
  
    year+1;  
  
end;  
  
run;
```

DO WHILE;

```
data work.invest;  
  
do while (capital>=50000);  
  
capital+2500;  
  
capital+capital*(.10);  
  
year+1;  
  
end;  
  
run;
```

```
data work.invest;  
  
do while (capital<=50000);  
  
capital+2500;  
  
capital+capital*(.10);  
  
year+1;  
  
output;  
  
end;  
  
run;
```

Conditional clauses with DO Loop;

```
data work.invest;  
  
do year=1 to 15 until (capital>=75000);  
  
capital+2000;  
  
capital+capital*(0.10);  
  
output;  
  
format capital dollar9.2;  
  
end;  
  
run;
```

Creating samples from an existing data set;

```
data work.subset;  
  
do sample=5 to 100 by 5;  
  
set sashelp.mon1001 point=sample;  
  
output;  
  
end;  
  
stop;  
  
run;
```

ARRAY statement;

```
DATA newlib.avgcelsius;  
  
    input City $ 1-18 jan feb mar apr may jun  
           jul aug sep oct nov dec;  
  
    DATALINES;  
  
State College, PA -2 -2 2 8 14 19 21 20 16 10 4 -1  
  
Miami, FL      20 20 22 23 26 27 28 28 27 26 23 20  
  
St. Louis, MO  -1 1 6 13 18 23 26 25 21 15 7 1  
  
New Orleans, LA 11 13 16 20 23 27 27 27 26 21 16 12  
  
Madison, WI    -8 -5 0 7 14 19 22 20 16 10 2 -5  
  
Houston, TX    10 12 16 20 23 27 28 28 26 21 16 12  
  
Phoenix, AZ    12 14 16 21 26 31 33 32 30 23 16 12  
  
Seattle, WA    5 6 7 10 13 16 18 18 16 12 8 6  
  
San Francisco, CA 10 12 12 13 14 15 15 16 17 16 14 11  
  
San Diego, CA  13 14 15 16 17 19 21 22 21 19 16 14  
  
;    RUN;
```

```
data work.ray (drop=month);  
  
set newlib.avgcelsius;  
  
array temp_fh (12) jan feb mar apr may jun jul aug sep oct nov dec;   (or use a * instead of 12)  
  
do month=1 to 12;  
  
temp_fh(month)= 1.8*temp_fh(month)+32;  
  
end;  
  
run;
```

OR

```
data work.ray (drop=month);  
  
set newlib.avgcelsius;  
  
array temp_fh (12)_numeric_;   (or use a * instead of 12)  
  
do month=1 to 12;  
  
temp_fh(month)= 1.8*temp_fh(month)+32;  
  
end;   run;
```

Creating variables with ARRAY statement;

```
data work.rays;  
  
drop month;  
  
set newlib.avgcelsius;  
  
array celcius(12) jan feb mar apr may jun jul aug sep oct nov dec;  
  
array fahrenheit (12) janfh febhf marfh aprfh mayfh junfh julfh augfh sepfh octfh novfh decfh;  
  
do month=1 to 12;  
  
fahrenheit(month)=1.8*celcius(month)+32;  
  
end;  
  
run;
```

```
data work.rays;

drop month;

set newlib.avgcelsius;

array celcius(12) jan feb mar apr may jun jul aug sep oct nov dec;

array fahrenheit (12);  (SAS names the variables when we don't specify variable names)

do month=1 to 12;

fahrenheit(month)=1.8*celcius(month)+32;

end;

run;
```

DIM function;

```
libname newlib '/folders/myfolders/newfolder';

data work.celc (drop=i);

set newlib.avgcelsius;

array temp(12) _numeric_;

do i=1 to dim(temp);

temp(i) = 1.8*temp(i)/32;

end;

run;
```


Reading FIXED FORMAT data;

(formatted input)

```
libname newlib '/folders/myfolders/newfolder';
```

```
DATA newlib.subject_data;
```

```
input @6 FName $11. @18 LName $ @1 SubjID 4. @25 Weight 2.
```

```
      +1 Height 3. +10 Calories comma6. @32 Date mmddyy8.;
```

```
format Date date7. Calories comma6.;
```

```
DATALINES;
```

```
1024 Alice    Smith 65 125 12/1/95 2,036
```

```
1167 Maryann  White 68 140 12/01/95 1,800
```

```
1168 Thomas   Jones 190 12/2/95 2,302
```

```
1201 Benedictine Arnold 68 190 11/30/95 2,432
```

```
1302 Felicia  Ho   63 115 1/1/96 1,972
```

```
;
```

```
RUN;
```

```
PROC PRINT data = newlib.subject_data;
```

```
  title 'Output dataset: TEMP';
```

```
  id SubjID;
```

```
RUN;
```

PAD Option;

```
data work.tempadd;
```

```
infile '/folders/myfolders/RawData/AddressDataSAS.txt' pad;
```

```
input id 4. name $ 6-23 @27 street $ 19.;
```

```
run;
```

FREE FORMAT data;

(list input)

```
data work.sampdata;
```

```
input gender $ age credit debit accounts;
```

```
datalines;
```

```
male 29 1 2 5
```

```
female 25 2 1 3
```

```
female 26 2 1 5
```

```
male 28 3 3 4
```

```
male 24 1 1 3
```

```
female 29 6 2 7
```

```
male 32 5 1 3
```

```
female 34 5 2 9
```

```
; run;
```

Extra Programs;

```
data work.ques;
```

```
input question (ans1-ans4) ($); (both range and $ sign in parantheses)
```

```
datalines;
```

```
1 a b c d
```

```
2 b d a c
```

```
3 a d c b
```

```
4 d b a c
```

```
5 b a c d
```

```
; run;
```

```
data work.files;  
input age (score1-score3) (7.);  (both range and format in parantheses)  
datalines;  
23 235.50 245.62 654.25  
24 467.24 458.69 625.30  
25 418.65 234.40 487.65  
;  
run;
```

```
data work.score;  
input idnum q1-q4;  (range cz q1-q4 is sequential)  
datalines;  
1001 56 65 52 35  
1002 68 95 57 42  
1003 32 54 87 68  
1004 35 69 85 42  
1005 65 47 89 24  
;  
run;
```

MISSOVER; (used when data is missing at the end of a record)

```
data temp;  
infile '/folders/myfolders/RawData/BankData.txt' dlm=', ' missover;  
input gender $ age credit debit acc;  
run;
```

DSD; (delimiter sensitive data- used when data is missing at the beginning or middle of a record)

```
data temp;  
  
infile '/folders/myfolders/RawData/Bank.txt' dsd dlm=' '; (default delimiter in this case is a comma)  
  
input gender $ age debit credit accounts;  
  
run;
```

LENGTH statement;

```
data survey;  
  
infile datalines dsd dlm=' '; (works with missover option as well)  
  
length name $ 9;  
  
input name (q1-q5)($);  
  
datalines;  
  
Robert A C A D  
  
William B C A D A  
  
Linda C B A C  
  
D D D C A  
  
Katherine A B C D A  
  
;  
  
run;
```

Ampersand (&) and colon (:) modifier;

```
data cities;

infile datalines firstobs=2;

input city & $12. population : comma.;  (comma. works only for modified list input – no width required)

format population comma9.;  (we need to specify the width in comma in formatted input)

datalines;

City Yr2000Popn

New York 8,008,278

Los Angeles 3,694,820

Chicago 2,896,016

Houston 1,953,631

Philadelphia 1,517,550

Phoenix 1,321,045

San Antonio 1,144,646

San Diego 1,223,400

Dallas 1,188,580

San Jose 894,943

; run;
```

Creating raw data files from sas data sets (creating free format data)

```
libname newlib '/folders/myfolders/newfolder';

data _null_;

set newlib.testdata;

file '/folders/myfolders/newfolder/test.txt' dlm=';';  (default delimiter in this case is a space- can also be performed without specifying a delimiter)

put obs id $ age sex $ date : date9.; run;
```

```
data _null_;
```

```
set sashelp.cars;
```

```
file '/folders/myfolders/newfolder/cars.txt' dsd dlm=' '; (default delimiter in dsd is a comma- can also  
be performed without specifying a delimiter)
```

```
put make model type msrp invoice;
```

```
format msrp invoice comma7.;
```

```
run;
```

Reading data within quoted strings;

```
data work.cars;
```

```
infile '/folders/myfolders/NewFolder/cars.txt' dsd dlm=' ';
```

```
length make $10 model $20;
```

```
input make model type $ msrp : comma6. invoice : comma6.;
```

```
format msrp invoice dollar8.;
```

```
run;
```

Mixed input styles;

```
DATA nationalparks;
```

```
input ParkName $ 1-22 State $ Year @40 Acreage comma9.; (or we can use : comma.)
```

```
DATALINES;
```

```
Yellowstone      ID/MT/WY 1872    4,065,493
```

```
Everglades       FL 1934      1,398,800
```

```
Yosemite         CA 1864      760,917
```

```
Great Smoky Mountains NC/TN 1926    520,269
```

```
Wolf Trap Farm   VA 1966      130
```

```
; RUN;
```

```
( column input list input formatted input )
```

Date formats;

```
data temp (where= (b_date < '01jan1960'd));

input id fname $11. lname $7. +3 wt 2. ht 30-32 @34 wt_date mmddyy8. @43 b_date mmddyy8.;

checkup= wt_date + 14;

age= (wt_date-b_date)/365.25;

format wt_date weekdate21. b_date worddate14.;

format checkup mmddyyd10. age 4.1;

datalines;

1024 Alice    Smith 1 65 125 12/1/05 01/01/60

1167 Maryann  White 1 68 140 12/01/05 01/01/59

1168 Thomas  Jones 2  190 12/2/05 06/15/60

1201 Benedictine Arnold 2 68 190 11/30/05 12/31/60

1302 Felicia  Ho   1 63 115 1/1/06 06/15/58

;

RUN;
```

Use of / and #n pointer controls;

```
data weather;
```

```
input city & $15. state $ /
```

```
high low mean /
```

```
rain;
```

```
datalines;
```

```
State College PA
```

```
32 16 25
```

```
2.4
```

```
Miami FL
```

```
75 58 67
```

```
2.0
```

```
Honolulu HI
```

```
80 65 74
```

```
3.6
```

```
; run;
```

OR

```
data weather;
```

```
input #3 rain
```

```
#1 city & $15. state $
```

```
#2 high low mean;
```

```
datalines;
```

```
State College PA
```

```
32 16 25
```


2.4

Miami FL

75 58 67

2.0

Honolulu HI

80 65 74

3.6

; run;

OR

data weather;

input #3 rain

#1 city & \$15. state \$ /

high low mean;

datalines;

State College PA

32 16 25

2.4

Miami FL

75 58 67

2.0

Honolulu HI

80 65 74

3.6

; run;

Trailing @@ sign;

data temprature;

input month \$ high low @@;

datalines;

Jan 32 16 Feb 35 18 Mar 46 26

Apr 58 37 May 68 47 Jun 78 56

Jul 82 60 Aug 80 58 Sep 72 51

Oct 61 40 Nov 48 32 Dec 37 22

;

run;

Trailing @ sign;

data examscore;

input id @;

do exam= 1,2,3;

input score @;

output;

end;

datalines;

111000234 79 82 100

922232573 87 89 95

252359873 65 72 73

205804679 92 95 99

;

run;

Trailing @ sign and MISSOVER option;

```
DATA dietdata;

  infile DATALINES MISSOVER;

  input id weight @;

    weighin = 0;

    DO WHILE (weight ne .);

      weighin+1;

      output;

      input weight @;

    END;

  DATALINES;

1001 179 172 169

1002 250 249

1003 190 196 195 164 158

1004 232 224 219 212 208

1005 211 208 204 202

;

RUN;
```

Converting data types (char to num and num to char)

```
data newlib.hrd;  
  
infile datalines;  
  
input city $ 1-11 @13 state $2. zip phone enddate startdate payrate $2. days @48 hours 3.;  
  
datalines;  
  
CARY      NC 27513 6224549 14567 14621 10 11 88  
  
CARY      NC 27513 6223251 14524 14565 8 25 200  
  
CHAPEL HILL NC 27514 9974749 14570 14608 40      26 208  
  
RALEIGH   NC 27612 6970450 14516 14527 15 10 80  
  
;  
  
run;  
  
  
  
proc contents data=newlib.hrd;  
  
run;  
  
  
  
data work.newhr;  
  
set newlib.hrd;  
  
wages=input(payrate,2.);  (input function is used for char to num conversion)  
  
salary=wages*hours;  
  
zipcode=put(zip,5.);  (put function is used for num to char conversion)  
  
run;  
  
  
  
proc contents data=work.newhr;  
  
run;
```

Concatenate num and char values;

```
data work.statezip;  
  
set newlib.hrd;  
  
statepin=zip||'/'||state;  (|| ' ' || - concatenation operator)  
  
run;
```

OR

```
data work.statezip;  
  
set newlib.hrd;  
  
statepin=put(zip,5.)||'/'||state;  
  
run;
```

YEAR and MONTH functions;

```
data dates;  
  
set hrd;  
  
if year(enddate)=1999 and month(enddate)=11;  
  
run;
```

WEEKDAY function;

```
data dates;  
  
set hrd;  
  
if weekday(enddate) in (2,4,5);  
  
run;
```

MDY function;

```
data newlib.hrd;  
  
input city $11. @13 state $ zip month day year phone $ rate days hours;  
  
datalines;  
  
CARY      NC 27513 01 12 2000 6224549 10 11 88  
  
CARY      NC 27513 11 17 1999 6223251 8 25 200  
  
CHAPEL HILL NC 27514 12 30 1999 6774749 40 26 208  
  
RALEIGH   NC 27612 10 10 1999 6970450 15 10 80  
  
;  
  
run;
```

```
data hr1 (drop= month day year);  
  
set newlib.hrd;  
  
date=mdy(month,day,year);  
  
comparedate=mdy(6,11,2000);  
  
run;
```

OR

```
data work.appraisal;  
  
input site & $11. day name &: $11. rate $;  
  
datalines;  
  
westin 12 mitchell, k a2  
  
stockhon 7 worton, m a5  
  
city centre 17 smith, a b1  
  
; run;
```

```
data incr;  
  
set appraisal;  
  
reviewdate=mdy(12,day,2012);  
  
format reviewdate mmddyy10.;  
  
run;
```

Date() and today() functions;

```
data incr;  
  
set appraisal;  
  
dates=date();  
  
dated=today();  
  
format dates mmddyy10. dated date9.;  
  
run;
```

More programs;

```
data createdates;  
  
current1=date();  
  
current2=today();  
  
current3=datejul(2015254); (2015254 specifies the 254th day of year 2015)  
  
current4=yyq(2013,3);  
  
format current1-current4 weekdate20.;  
  
run;
```

YEAR, QTR, MONTH, WEEKDAY, DAY functions;

```
DATA massaged (drop= wt_date weight) ;
```

```
input subj 1-4 l_name $ 18-23 weight 30-32
```

```
+1 wt_date mmddyy8. @43 b_date mmddyy8.;
```

```
yr=year(b_date);
```

```
qtr=qtr(b_date);
```

```
mnth=month(b_date);
```

```
wkday=weekday(b_date);
```

```
days=day(b_date);
```

```
format b_date mmddyy10.;
```

```
DATALINES;
```

```
1024 Alice    Smith 1 65 125 12/1/05 01/01/60
```

```
1167 Maryann  White 1 68 140 12/01/05 01/01/59
```

```
1168 Thomas   Jones 2  190 12/2/05 06/15/60
```

```
1201 Benedictine Arnold 2 68 190 11/30/05 12/31/60
```

```
1302 Felicia  Ho   1 63 115 1/1/06 06/15/58
```

```
;
```

```
RUN;
```


YRDIF and DATDIF functions;

```
DATA massaged (drop= weight) ;

input subj 1-4 |_name $ 18-23 weight 30-32

+1 wt_date mmddyy8. @43 b_date mmddyy8.;

ageinyrs=yrdif(b_date,wt_date,'act/act');

ageindays=datdif(b_date,wt_date,'act/act');

ageinyrs=round(ageinyrs,.2);

format b_date wt_date mmddyy10.;

DATA LINES;

1024 Alice    Smith 1 65 125 12/1/05 01/01/60

1167 Maryann  White 1 68 140 12/01/05 01/01/59

1168 Thomas  Jones 2  190 12/2/05 06/15/60

1201 Benedictine Arnold 2 68 190 11/30/05 12/31/60

1302 Felicia  Ho   1 63 115 1/1/06 06/15/58

;

RUN;
```

SCAN and SUBSTR function;

```
data bases (keep=team divsn position lastname firstname);

length lastname firstname $12 divsn $3;

set sashelp.baseball;

lastname=scan(name,1);

firstname=scan(name,2);

divsn=substr(division,1,1);  run;
```

AND

```
data sale (keep=year qtr quantity profit);  
  
set sashelp.orsales;  
  
length qtr $4;  
  
qtr=substr(quarter,3,4);  
  
run;
```

Replacing text with SUBSTR function;

```
data project;  
  
set sashelp.failure;  
  
substr(process,4,4)='ject';  
  
run;
```

OR

```
data hr1 (drop=exchange);  
  
set newlib.hrd;  
  
length exchange $3;  
  
exchange=substr(phone,1,3);  
  
if exchange='622' then substr(phone,1,3)='433';  
  
run;
```

TRIM function;

```
data newlib.hrdept;

input address $19. @21 city $11. @33 state $ zip month day year phone $ rate days hours;

datalines;

65 Elm Drive      CARY      NC 27513 01 12 2000 6224549 10 11 88

11 Sun Drive      CARY      NC 27513 11 17 1999 6223251 8 25 200

712 Hardwick Street CHAPEL HILL NC 27514 12 30 1999 6774749 40 26 208

5372 Whitebud Road RALEIGH    NC 27612 10 10 1999 6970450 15 10 80

;

run;
```

```
data hr2 (drop=address city state zip);

set newlib.hrdept;

newaddress=trim(address)||', '||trim(city)||', '||zip;  run;
```

CATX function;

```
data hr3 (drop=address city state zip);

set newlib.hrdept;

newadd=catx(' ', address, city, zip);

run;
```

INDEX function;

```
data datapool;  
  
set sashelp.cars;  
  
if index(drivetrain,'All')>0;  (we can also use find instead of index)  
  
run;
```

UPCASE, LOWCASE, PROPCASE;

```
data dataset;  
  
set newlib.hrdept;  
  
address=upcase(address);  
  
city=propcase(city);  
  
state=lowcase(state);  
  
run;
```

TRANWRD function;

```
data carchange;  
  
set sashelp.cars;  
  
type=tranwrd(type,'Truck','PickUp');  
  
run;
```

INT and ROUND function;

```
data vote;  
  
set sashelp.vote1980;  
  
pop=int(pop);  
  
edu=round(edu,.2);  
  
run;
```

Nesting functions;

```
data base;  
  
set sashelp.baseball;  
  
nestfunc=substr(scan(name,2),1,1);  
  
run;
```

INTCK function;

```
DATA massaged (drop= weight) ;  
  
    input subj 1-4 |_name $ 18-23 weight 30-32  
           +1 wt_date mmddyy8. @43 b_date mmddyy8.;  
  
ageinyrs=intck('year',b_date,wt_date);  
ageinmonths=intck('month',b_date,wt_date);  
ageindays=intck('day',b_date,wt_date);  
ageinweeks=intck('week',b_date,wt_date);  
  
format b_date wt_date mmddyy10.;  
  
DATALINES;  
  
1024 Alice    Smith 1 65 125 12/1/05 01/01/60  
  
1167 Maryann  White 1 68 140 12/01/05 01/01/59  
  
1168 Thomas   Jones 2  190 12/2/05 06/15/60  
  
1201 Benedictine Arnold 2 68 190 11/30/05 12/31/60  
  
1302 Felicia  Ho    1 63 115 1/1/06 06/15/58  
  
;  
  
RUN;
```

```

data timeintervals;

first=intck('day','03jan2001'd,'02jul2002'd);

second=intck('weekday','03jan2001'd,'02jul2002'd);

third=intck('week','03jan2001'd,'02jul2002'd);

fourth=intck('month','03jan2001'd,'02jul2002'd);

fifth=intck('qtr','03jan2001'd,'02jul2002'd);

sixth=intck('year','03jan2001'd,'02jul2002'd);

run;

```

INTNX function;

```

data dateintervals;

one=intnx('day','02jan2003'd,5);

two=intnx('week','02jan2003'd,4);

three=intnx('month','02jan2003'd,3,'b');

four=intnx('qtr','02jan2003'd,2,'m');

five=intnx('year','02jan2003'd,1,'e');

format one two three four five weekdate20.;

run;

```

```

DATA diet (drop=weight subj b_date);

input subj 1-4 l_name $ 18-23 weight 30-32

      +1 wt_date mmddyy8. @43 b_date mmddyy8.;

nxt_yr=intnx('year',wt_date,1,'b');

nxt_qtr=intnx('qtr',wt_date,2,'m');

nxt_mnth=intnx('month',wt_date,3,'e');

```

```

nxt_week=intnx('week',wt_date,4);
nxt_day=intnx('day',wt_date,5);
format nxt_yr nxt_qtr nxt_mnth nxt_week nxt_day wt_date weekdate20.;
DATA LINES;
1024 Alice    Smith 1 65 125 12/1/05 01/01/60
1167 Maryann  White 1 68 140 12/01/05 01/01/59
1168 Thomas   Jones 2  190 12/2/05 06/15/60
1201 Benedictine Arnold 2 68 190 11/30/05 12/31/60
1302 Felicia  Ho   1 63 115 1/1/06 06/15/58
;
RUN;

```

ODS HTML output;

```

libname newlib '/folders/myfolders/output';
ods html file (or body)='/folders/myfolders/output/car.html';
proc print data=sashelp.cars;
title 'Car data in web file';
id make;
var type origin drivetrain;
run;
ods html close;

ods html body='/folders/myfolders/output/car.html'
      contents='/folders/myfolders/output/cartoc.html'
      frame='/folders/myfolders/output/carframe.html';

```

```
proc print data=sashelp.cars;  
title 'Car data in web file';  
id make;  
var msrp invoice type origin drivetrain;  
run;  
ods html close;
```

```
ods html body='/folders/myfolders/output/car.html' (url='car.html')  
      contents='/folders/myfolders/output/cartoc.html' (url='cartoc.html')  
      frame='/folders/myfolders/output/carframe.html';  
proc print data=sashelp.cars;  
title 'Car data in web file';  
id make;  
var msrp invoice type origin drivetrain;  
run;  ods html close;
```

```
ods html path='/folders/myfolders/output' (url='none')  
      body='car.html'  
      contents='cartoc.html'  
      frame='carframe.html';  
proc print data=sashelp.cars;  
title 'Car data in web file';  
id make;  
var msrp invoice type origin drivetrain;  
run;  ods html close;
```


Word Output (ods rtf);

```
ods rtf file='/folders/myfolders/output/fish.rtf';  
  
proc print data=sashelp.fish;  
  
run;  
  
ods rtf close;
```

Ods Pdf;

```
ods pdf file='/folders/myfolders/output/fish.pdf' style=htmlblue;  (style= is optional)  
  
proc print data=sashelp.fish;  
  
run;  
  
ods pdf close;
```

```
data test;  
  
input obs name$ level;  
  
if level=. then expertise='Unknown';  
  
else if level=1 then expertise='Low';  
  
else if level=2 or 3 then expertise='Medium';  
  
else expertise='High';  
  
cards;  
  
1 Frank 1  
  
2 Joan 2  
  
3 Sui 5  
  
4 Jose 3  
  
5 Bert 4  
  
6 Kelly .
```

7 Juan 6

;

run;

data files;

infile cards;

input height 2. @4 weight 2;

cards;

72 95

;

run;

proc sort data=sashelp.cars out=cars;

by make descending cylinders;

run;

data new;

set cars;

retain temp;

by make descending cylinders;

if first.make then do;

temp=cylinders;

output;

end;

if last.make then do;

```
new=horsepower-cylinders;
```

```
output;
```

```
end;
```

```
run;
```

```
proc report data = sashelp.fish;
```

```
column species weight;
```

```
define species / group width = 12 'FishType';
```

```
define weight / mean format = 5.2 width = 8 'AvgWt';
```

```
run;
```

```
title2 'Fish';
```

```
title3 'Types';
```

```
proc tabulate data=sashelp.fish;
```

```
class species;
```

```
var weight;
```

```
table species, weight * mean;
```

```
run;
```

```
proc means data = sashelp.fish mean;
```

```
class species height width;
```

```
var weight;
```

```
output out = fish mean = weight;
```

```
run;
```

