

```

/* CHARACTER FUNCTIONS */

/* PART-1 */

/* REMOVEING SPACES/BLANKS/CHARCTERS/NUMBERS FROM A STRING:---> */

/* TYPE OF BLANKS: */

/* 1. TRAILING BLANKS: */

/* Blanks at the end of the string. */
/* Eg: */
/* Name:-  thoamas */

/* 2. LEADING BLANKS: */

/* Blanks at the beginning of the string */
/* Eg: */
/* Name:  Thomas */

/* 3. BLANKS IN BETWEEN: */
/* Blanks between two strings. */
/* Eg: */
/* Name:  Thomas  masker */

/* FUNCTIONS TO REMOVE THE SPACES/BLANKS: */

/* A. TRIM */
/* B. TRIMN */
/* C. STRIP */

/* D. COMPRESS:  Removes all the blanks/spaces */
/* Eg: */
/* Name:  Thomas  Cruis  Mapother */
/* Result:  ThomasCruisiMapother */

/* E. COMBBL:  This translates/compress more than one consecutive spaces/blanks into one. */
/* Eg: */
/* Name:  Thomas  Cruis  Mapother */
/* Result:  Thomas Cruis Mapother */

/* It shows the length of the name. */

.....
data sample;
name="Thomas Cruise Mapothe";
len = length(name);
run;

/* We give some spaces between the name. */

.....
data sample;
name="Thomas  Cruise  Mapothe";

```

```
len = length(name);  
run;
```

```
/* COMPRESS */
```

```
/* I remove all the spaces of the string. */
```

```
data sample;  
name="Thomas Cruise Mapothe";  
len = length(name);  
compp=compress(name);  
len_2=length(compp);  
run;
```

```
/* SYNTAX:--> COMPRESS(SOURCE,<CHARACTER>,<MODIFIER>) */
```

```
/* It also remove all the special characters from the strings. */
```

```
data sample2;  
value="asdjlAkfbAas12345;%";  
new_value=compress(value,';%');  
run;
```

```
/* It also remove characters from the strings. */
```

```
data sample2;  
value="asdjlkbfas12345;%";  
new_value=compress(value,'a');  
run;
```

```
/* It removes all the "a" alphabets in the strings. */
```

```
data sample2;  
value="asdjlkbfas12345;%";  
new_value=compress(value,'a','i');  
run;
```

```
/* It removes all the number from the strings. */
```

```
data sample2;  
value="asdjlkbfas12345;%";  
new_value=compress(value,'','d');  
run;
```

```
/* COMPBL */
```

```
/* It give the original string with removeing all the spaces. */
```

```
data sample;  
name="Thomas Cruise Mapothe";  
len = length(name);  
compp=compress(name);  
len_2=length(compp);  
cobm=compbl(name);  
len_3=length(cobm);  
run;
```

```
/* PART-2 */
```

```
/* CONCATENATION:---> */
```

```
/* HOW TO JOINING THE STRINGS: */
```

```
/* A. MANUALLY ---> USING PIPE SYMBOL IN THE KEYBOARD */
```

```
DATA SAMPLE;  
FIRST_NAME="THOMAS";  
MIDDLE_NAME="CRUIS";  
LAST_NAME="MAPOTHER";  
FULL_NAME=FIRST_NAME||MIDDLE_NAME||LAST_NAME;  
RUN;
```

```
/* SPACE SEPARATED STRINGS */
```

```
DATA SAMPLE;  
FIRST_NAME="THOMAS";  
MIDDLE_NAME="CRUIS";  
LAST_NAME="MAPOTHER";  
FULL_NAME=FIRST_NAME||' '||MIDDLE_NAME||' '||LAST_NAME;  
RUN;
```

```
/* B. CAT FUNCTION: */
```

```
/* IT JOIN ALL THE STRING WITHOUT SPACES SEPARATED. */
```

```
DATA SAMPLE;  
FIRST_NAME="THOMAS";  
MIDDLE_NAME="CRUIS";  
LAST_NAME="MAPOTHER";  
FULL_NAME=FIRST_NAME||' '||MIDDLE_NAME||' '||LAST_NAME;  
FULL_NAME2=CAT(FIRST_NAME,MIDDLE_NAME,LAST_NAME);  
RUN;
```

```
/* C. CATX FUNCTION: */
```

```
/* IT JOIN ALL THE STRING WITH SPACE SEPARATED. */  
/* YOU ONLY GIVE THE SEPARATORS. */
```

```
DATA SAMPLE;  
FIRST_NAME="THOMAS";  
MIDDLE_NAME="CRUIS";  
LAST_NAME="MAPOTHER";  
FULL_NAME=FIRST_NAME||' '||MIDDLE_NAME||' '||LAST_NAME;  
FULL_NAME2=CAT(FIRST_NAME,MIDDLE_NAME,LAST_NAME);  
FULL_NAME3=CATX(' ',FIRST_NAME,MIDDLE_NAME,LAST_NAME);  
RUN;
```

```
/* D. CATS FUNCTION: */
```

```
/* E. CATT FUNCTION: */
```

```
/* PART-3:-----> */
```

```
/* SPLITING OF THE STRINGS: */
```

```
.....  
PROC IMPORT DATAFILE="/home/u63730693/my_sas/CHART.xlsx"
```

```
    out=chars  
    dbms=xlsx replace;  
    sheet="Sheet1";
```

```
run;
```

```
.....  
data test;
```

```
set work.chars;
```

```
/* SCAN FUNCTION: */
```

```
/* SYNTAX--> SCAN(CHAR,COUNT,MODIFIER) */
```

```
.....  
DATA SAMPLE;
```

```
SET WORK.CHARS;
```

```
FIRST_NAME=SCAN(NAME,1,'_');
```

```
MIDDLE_NAME=SCAN(NAME,2,'_');
```

```
LAST_NAME=SCAN(NAME,3,'_');
```

```
RUN;
```

```
/* PART-4:---> */
```

```
.....  
data card_info;
```

```
INPUT CARD_NUMBER $ 1-50;
```

```
DATALINES;
```

```
4444333322221110
```

```
9876435209873450
```

```
5674321487907680
```

```
8765432199002340
```

```
3456789012340000
```

```
6578329234562220
```

```
7890654324564440
```

```
7659010983245210
```

```
;
```

```
RUN;
```

```
/* EXTRCATING A PART OF STRING */
```

```
/* SUBSTR FUNCTION: */
```

```
/* SYNTAX-- SUBSTR(SOURCE,STRATING POSITION,NUMBER OF CHARACTERS) */
```

```
/* EG-1: */
```

```
/* IT GIVE LAST 4 DIGITS OF ALL THE NUMBERS */
```

```
DATA SAMPLE;  
SET CARD_INFO;  
LAST_DIGITS=SUBSTR(CARD_NUMBER,12,4);  
RUN;
```

```
/* EG-2: */
```

```
/* IT PUT * TO THE MIDDLE NUMBERS AS BANKING CARD NUMBER. */
```

```
DATA SAMPLE;  
SET CARD_INFO;  
LAST_DIGITS=SUBSTR(CARD_NUMBER,12,4);  
SUBSTR(CARD_NUMBER,5,8)="*****";  
RUN;
```

```
/* CHANGE CASE FUNCTION: */
```

```
data S_NAME;  
INPUT named $ 1-50;  
DATALINES;  
thomas  
adison  
rakesh  
samesh  
rukesh  
SABESH  
RABESH  
;  
RUN;
```

```
/* a. LOWCASE: */  
/* IT COVERT THE STRINGS TO SMALL LETTERS. */  
/* EG:-- thomas adison */
```

```
DATA CHANGE_CASED;  
SET S_NAME;  
CAPITALS=UPCASE(NAMED);  
RUN;
```

```
/* b. UPCASE: */  
/* IT CONVERT THE STRINGS TO UPPER LETTERS. */  
/* EG:-- THOMAS ADISON ` */
```

```
DATA CHANGE_CASED;  
SET S_NAME;  
SMALLL=LOWCASE(NAMED);  
RUN;
```

```
/* c. PROPCASE: */  
/* IT CONVERT THE ONLY 1ST LETTER OF THE STRING TO UPPER LETTER. */  
/* EG:--- Thomas adison */
```

```
DATA CHANGE_CASED;  
SET S_NAME;  
SENTEC=PROPCASE(NAMED);  
RUN;
```

```
/* PART-5:---> */
```

```
/* REPLACE PART OF STRING: */
```

```
/* a. TRANWRD: */
```

```
/* IT REPLACES ALL OCCURRENCES A SPECIFIC WORD IN A GIVEN STRING. */
```

```
/* eg:- */
```

```
/*      thomas cruis mapother */  
/*  RESULT-- thomas cruis enough */
```

```
/* eg-2:-- */
```

```
/*  john is good boy. that boy is naughty. */  
/*  RESULT--- john is good girl. that girl is naughty. */
```

```
/* SYNTAX--- TRANWRD(VARIABLE, REPLACE WHAT, REPLACE WITH) */
```

```
DATA SAMPLE;  
NAME="THOMAS CRUSI MAPOTHE";  
RUN;
```

```
DATA SAMPLE;  
NAME="THOMAS CRUSI MAPOTHE";  
CHANGED=TRANWRD(NAME, "MAPOTHE", "ENOUGH");  
RUN;
```

```
/* IT SHOW IN TABLUAR FORM. */
```

```
DATA SAMPLE;  
NAME="THOMAS CRUSI MAPOTHE";  
CHANGED=TRANWRD(NAME, "MAPOTHE", "ENOUGH");  
RUN;  
PROC PRINT DATA=SAMPLE;  
RUN;
```

```
DATA SAMPLE;  
NAME="THOMAS CRUSI MAPOTHE";  
CHANGED=TRANWRD(NAME, "MAPOTHE", "ENOUGH");  
SEC_EXP="JOHN IS GOOD BOY. THAT BOY IS NAUGHTY";  
CHN_SEC=TRANWRD(SEC_EXP, "BOY", "GIRL");  
RUN;
```

```
/* IMPORATANT EXAMPLE: */
```

```
DATA N_D_2;  
INPUT NAME $10. GENDER $1.;  
CARDS;  
MS.THOMAS      M
```

```
MR.RAVI           M
MS.SAKHSA        M
MR.RASHI         F
MS.LAXSMI        F
MR.RAVINA        F
;
RUN;
```

```
DATA SAMPLE2;
SET n_d_2;
```

```
IF UPCASE(GENDER) = "M" THEN NAME2 = TRANWRD(NAME,"MS.","MR.");
ELSE IF UPCASE(GENDER) = "F" THEN NAME2 = TRANWRD(NAME,"MR.","MS.");
ELSE NAME2=NAME;
RUN;
```

```
/* b. TRANSLATE FUNCTION: */
```

```
/* REPLACES SPECIFIC CHARACTERS IN A GIVEN CHARACTER EXPRESSION. */
```

```
/* SYNTAX--> TRANSLATE(VARIABLE, REPLACE WITH, REPLACE WHAT) */
```

```
/* EG-1: */
```

```
/* IT CONVERT THE CHARACTER "A" AND "M" TO "I" AND "X" */
```

```
DATA SAMPLE3;
NAME="SAMEERS";
NAME2= TRANSLATE(NAME,'IX','AM');
RUN;
```

```
/* EG-2: */
```

```
/* IT ONLY CONVERT THE CHARCTER "A" TO "I" */
```

```
DATA SAMPLE3;
NAME="SAMEERS";
NAME2= TRANSLATE(NAME,'IX','AZ');
RUN;
```

```
/* PART-6:--> */
```

```
/* FINDING CHARACTER/WORD IN A STRING:-- */
```

```
/* a. INDEX FUNCTION: */
```

```
/* IT RETURNS THE POSITION OF SPECIFIED STRING VALUE. */
```

```
/* SYNTAX--> INDEX(SOURCE, FIND WHAT) */
```

```
DATA SAMPLES;
NAME="AMAZON.COM";
EXIST=INDEX(NAME,".");
RUN;
```

```
DATA SAMPLES;  
NAME="AMAZON.COM";  
EXIST=INDEX(NAME,".");
```

```
IF EXIST > 0 THEN OUTPUT;  
RUN;
```

```
/* CONDITIONAL FINDING OF CHARACTER/WORD IN A STRING */
```

```
/* b. FIND FUNCTION: */
```

```
/* IT WORKS LIKE INDEX HOWEVER PROVIDES OPTIONAL  
FINDING FEATURES LIKE :-- START POSISTION ARGUMENT, MODIFIERS ETC. */
```

```
/* SYNTAX--> FIND(STRING,SUBSTRING,<MODIFIER>,<STRAT-POSITION>) */
```

```
/* i--> IGNORE CASES */
```

```
DATA SAMPLES;  
NAME="AMAZON.COM";  
EXIST=FIND(NAME,".", "i");  
RUN;
```

```
/* EG-2: */
```

```
/* IT WILL GIVE THE POSITION OF FIRST "COUNTRY" */
```

```
DATA SAMP;  
STRINGS="INDIA IS A GREAT COUNTRY TO LIVE. THIS IS THE COUNTRY OF FESTIVALS";  
EXITST=FIND(STRINGS,"COUNTRY");  
RUN;
```

```
/* IT WILL GIVE THE POSITION OF SECOND "COUNTRY" */
```

```
DATA SAMP;  
STRINGS="INDIA IS A GREAT COUNTRY TO LIVE. THIS IS THE COUNTRY OF FESTIVALS";  
EXITST=FIND(STRINGS,"COUNTRY",25);  
RUN;
```

```
/* PART-7:---> */
```

```
/* SEARCH FOR PATTERN IN A STRING: */
```

```
/* PRXMATCH FUNCTION: */
```

```
/* IT STANDS FOR PERL ECPRESSION MATCH. */
```

```
/* IT GIVE THE OUTPUT AS THE POSITION OF THE STRING */
```

```
/* SYNTAX--> PRXMATCH(PERL-REGULAR-EXPRESSION, SOURCE) */
```

```
/* EG-1: */
```

```
/* IT GIVE THE OUTPUT POSITION OF LETTER WHICH STARTS WITH LETTER "S". */
```

```
DATA TEST;  
SET WORK.CHARS;  
FLAG=PRXMATCH("/^S/",NAME);  
RUN;
```



```
/* EG-2: */  
/* IT GIVE THE OUTPUT POSITION OF THE LETTER WHICH STARTS WITH LETTERS "S" & "M" */
```

```
DATA TEST;  
SET WORK.CHARS;  
FLAG=PRXMATCH("/^S|^M/",NAME);  
RUN;
```

```
/* EG-3: */
```

```
DATA TEST1;  
INPUT COMMENTS $ 1-100;  
DATALINES  
;
```

```
THE CUSTOMER CALLED FROM HIS NO. 9765543331 AND MADE THE PMT OF RS. 989782  
REGISTERED THE CUSTOMER'S NUMBER AS 8844993366 RESIDING IN THE ZIP CODE AREAS 998855  
DATE OF BIRTH IN DDMYYYY FORMAT WAS 08081995 AND THE NUMBER UPDATED WAS 9900675421  
;  
RUN;
```

```
/* IT GIVE THE POSITION NUMBER OF MOBILE_NUMBER IN THE COMMENTS */
```

```
DATA MOBILE_NUM;  
SET TEST1;  
MOB_NUM_POSITION=PRXMATCH("/\d{10}/",COMMENTS);  
RUN;
```

```
/* TO EXTRACT THE NUMBER FROM THE COMMENTS USE SUBSTR FUNCTION. */
```

```
DATA MOBILE;  
SET TEST1;  
MOB_NUM_POSITION=PRXMATCH("/\d{10}/",COMMENTS);  
MOB_NUMM=SUBSTR(COMMENTS,PRXMATCH("/\d{10}/",COMMENTS),10);  
RUN;
```