DATA Cars;

INPUT Make $;

DATALINES;

Porsche\_Cayenne

Audi

BMW

;

RUN;

PROC CONTENTS;

RUN;

Data Cars;

Length Make $ 15. Default=4;

Input Make $ Year;

Datalines;

Porsche\_Cayenne 2018

Audi 2016

BMW 2014

;

Run;

PROC CONTENTS;

RUN;

PROC PRINT;

RUN;

Data Cars;

Length Make $ 5.;

Input Make $;

Datalines;

Audi

;

Data Length;

Set Cars;

Length\_Trimmed=Length(Make);

Length\_Non\_Trimmed=Lengthc(Make);

Run;

PROC CONTENTS;

RUN;

PROC PRINT;

RUN;

Data Case;

Set Cars;

Upper=UpCase(Make);

Proper=PropCase(Make);

Lower=LowCase(Upper);

Proper\_second\_argument=PropCase(Make, "\_");

Run;

Data Case\_Test;

Set Cars;

Upper\_Pos = AnyUpper(Make);

Lower\_Pos = AnyLower(Make);

Tupper\_Pos = NoTupper(Make);

Run;

PROC PRINT;

RUN;

Data Align;

Set Cars;

Char\_right=Right(Make);

Num\_left=Left(Year);

Run;

PROC CONTENTS;

RUN;

PROC PRINT;

RUN;

Data Cities;

Input City $50.;

First=Scan(City, +1);

Last=Scan(City,-1);

Fourth=Scan(City,4);

Datalines;

Chicago Paris London Geneva Dublin

;

PROC CONTENTS;

RUN;

PROC PRINT;

RUN;

Data Cities;

Input City $50.;

First=Scan(City, +1);

Last=Scan(City,-1);

Fourth=Scan(City,4);

Fourth\_alt=Scan(Compress(City,"##"),4);

Datalines;

,Chicago,Paris,~,##London,Geneva,Dublin,

;

PROC CONTENTS;

RUN;

PROC PRINT;

RUN;

Data Two\_Year\_Payment;

Initial = 26000;

Balance = Initial - 2000;

Do i = 1 to 12;

Balance = Balance - 2000;

Output;

End;

Run;

PROC PRINT;

RUN;

Data Scan\_in\_loop;

Length thedebate $50;

thedebate = "Is Pluto a Planet, well yes, and no";

delim = ',';

modif = 'oq';

nwords = CountW(thedebate, delim, modif);

Do count = 1 to nwords;

words = scan(thedebate, count, delim, modif);

Output;

End;

Run;

PROC PRINT;

RUN;

Data Compare;

Length String $20.;

Format String $20.;

String = " 3 fn comparison ";

Compress="#"||Compress(String)||"#";

Trim="#"||Trim(String)||"#";

Strip="#"||Strip(String)||"#";

Run;

PROC PRINT Noobs;

RUN;

data test;

string\_x = 'Indexit(FINDIT)';

string\_y = 'findit';

Not\_Found = index(string\_x,string\_y);

Found=index(string\_x,upcase(string\_y));

run;

PROC PRINT;

RUN;

Data Indexc;

String="It's confusing";

Answer\_A=indexc(String,'sortit','how?');

Answer\_B=indexc(String,'sortit',"i can't");

Answer\_C=indexc(String,'sortit','I can');

run;

PROC PRINT;

RUN;

Data Test;

String="It's confusing now #";

Del=IndexW(String, "#", "#");

put del;

Run;

Data Test1;

String="It's confusing #now #";

Del=IndexW(String, "#now", "#");

put del;

Run;

Data Test1;

String="It's confusing now #";

Del=IndexW(String, "now", "#");

put del;

Run;

Data Test2;

String="It's confusing $ $now$ #";

Del=IndexW(String, "$now$", "$");

put del;

Run;

Data Test2;

String="It's confusing $now$ #";

Del=IndexW(String, "$now$", "$");

put del;

Run;

Data Test2;

String="It's confusing $ $now$ #";

Del=IndexW(String, "$now$", "$");

put del;

Run;

Data Blank;

String= "Dealing with Spaces";

Resolve=IndexW(String, " Spaces");

Run;

Data Blank;

String= "Dealing with Spaces";

Resolve=IndexW(String, "Spaces");

Run;

Data Blanksearch;

String = " Blank ";

Resolve = IndexW(String, Trimn(" "));

Run;

Data Blanksearch1;

String = " ";

Resolve = IndexW(String, Trimn(" "));

Run;

Data Find;

String = "We will explore the FIND function. Won't we?";

String\_Length = Length(String);

Answer = Find(String, "we");

Run;

Data Non\_Case;

String = "We will explore the FIND function. Won't we?";

String\_Length = Length(String);

Answer = Find(String, "we", "i");

Run;

Data Startpos;

String = "We will explore the FIND function. Won't we?";

String\_Length = Length(String);

Startposvar = 2;

Answer = Find(String, "we", "i", Startposvar);

Run;

Data Negativestart;

String = "We will explore the FIND function. Won't we?";

String\_Length = Length(String);

Startposvar = 3-44;

Answer = Find(String, "we", "i", Startposvar);

Run;

Data Leadingandtrailing;

String = "We will explore the FIND function. Won't we?";

String\_Length = Length(String);

Startposvar = 1;

Answer = Find(String, " explore ", "i", Startposvar);

Answer\_1 = Find(String, "explore ", "t", Startposvar);

Answer\_2 = Find(String, " explore ", "t", Startposvar);

Run;

Data Select;

A = Coalesce(1,2,3,4,5,6,7,8,9,10,1);

B = Coalesce(1,2,.,.,5,6,7,8,9,10,1);

C = Coalesce(1.,2.,.,.,5,6,7,8,9,10,1);

D = Coalesce(.,.,.,.,.,.,.,.,.,.,.);

E = Coalescec(0,2,.,.,5,6,7,8,9,10,1.);

F = Coalescec(0,2,.,.,5,6,7,8,9,10,'1#', 'Choose me');

G = Coalescec('Choose me', 'No!');

Run;

Proc Print noobs;

Run;

Proc sql;

select \* from dictionary.tables;

Quit;

proc sql;

describe table dictionary.indexes;

Data Emissions;

Input Year Month $3. Coal Gas Petrol Diesel Nuclear;

Datalines;

2018 Jan 110 112 113 114 112

2018 Feb 110 113 114 116 112

2018 Mar 112 114 114 116 110

2018 Apr 114 115 113 115 111

2018 May 116 114 112 114 110

;

Data Add\_Month;

Set Emissions;

Format Date\_next Date9.;

Date\_next = INTNX ('Month', '1Jan2018'd, \_n\_);

Format Date\_current Date9.;

Date\_current = INTNX ('Month', '1Jan2018'd, \_n\_ - 1);

Format Date\_plus\_one Date9.;

Date\_plus\_one = INTNX ('Month', '1Jan2018'd, \_n\_ + 1);

Run;

Proc Print Noobs;

Run;

Data Interval\_Days;

Set Add\_Month (Drop = Date\_next Date\_plus\_one);

Interval\_Days

= INTNX('Month', Date\_Current, 1) - INTNX('Month', Date\_Current, 0);

Interval\_Days1

= INTNX('Month', Date\_Current, 2) - INTNX('Month', Date\_Current, 0);

Interval\_Days2

= INTNX('Month', Date\_Current, -1) - INTNX('Month', Date\_Current, 0);

Interval\_Days3

= INTNX('Month', Date\_Current, 3) - INTNX('Month', Date\_Current, 0);

Run;

Data Cieling\_Years;

Set Add\_Month (Drop = Date\_next Date\_plus\_one);

Format OldYear Year4.;

Format NewYear Year4.;

OldYear = INTNX('Year', Date\_Current + 1, -1);

CurrentYear = Year(Date\_Current);

NewYear = INTNX('Year', Date\_Current, 1);

Run;

Data Alignment;

Format Beginning Date9.;

Beginning = INTNX('Month', '31Jan2019'd, 7, 'Beginning');

Format Middle Date9.;

Middle = INTNX('Month', '31Jan2019'd, 7, 'Middle');

Format End\_ Date9.;

End\_ = INTNX('Month', '14Jan2019'd, 7, 'End');

Format SameDay\_1 Date9.;

Sameday\_1 = INTNX('Month', '31Jan2019'd, 7, 'Sameday');

Format SameDay\_2 Date9.;

Sameday\_2 = Intnx ('Month', '09Jul2019'd, 1, 'S');

Run;

Data INTCK;

Years\_1 = INTCK ('Year', '01Jan2011'd, '01Aug2019'd);

Years\_2 = INTCK ('Days365', '01Jan2011'd, '01Aug2019'd);

Years\_3 = INTCK ('Year', '31Dec2018'd, '01Jan2019'd);

Years\_4 = INTCK ('Days365', '31Dec2018'd, '01Jan2019'd);

Months = INTCK ('Month', '01Jan2011'd, '01Aug2019'd);

Days = INTCK ('Days', '01Jan2011'd, '01Aug2019'd);

SemiYear = INTCK ('SemiYear', '01Jan2011'd, '01Aug2019'd);

Quarters = INTCK ('Qtr', '01Jan2011'd, '01Aug2019'd);

Hours = INTCK ('Hour', '14:00:56't, '23:45:54't);

Minutes = INTCK ('Minute', '14:00:56't, '23:45:54't);

Seconds = INTCK ('Second', '14:00:56't, '23:45:54't);

Run;

Data Weeks;

Format Week\_Start Date9.;

Format Week1-Week5 Date9.;

Week\_Start = '18Aug2019'd;

Week1 = INTNX('Week.1', '18Aug2019'd, 1);

Week2 = INTNX('Week.2', '18Aug2019'd -1, 3);

Week3 = INTNX('Week.3', '18Aug2019'd -1, 4);

Week4 = INTNX('Week.4', '18Aug2019'd -1, 5);

Week5 = INTNX('Week.5', '18Aug2019'd -1, 6);

Run;

Proc print;

Run;

Data Method;

Input Type $ Production Jul :Date9. Aug :Date9.;

Format Jul :Date9. Aug :Date9.;

Datalines;

W/e 131 07Jul2019 06Oct2019

W/e 234 14Jul2019 13Oct2019

W/e 232 21Jul2019 20Oct2019

W/e 212 28Jul2019 27Oct2019

M/e 203 31Jul2019 31Oct2019

;

Data Comparison;

Set Method;

Month\_D = INTCK ('Month', Jul, Aug);

Month\_C = INTCK ('Month', Jul, Aug, 'C');

Run;

Proc print;

Run;

Data \_Null\_;

A = "This ";

B = " is";

C = " a test ";

D = " of CAT function";

Out\_Symbol = A||B||C||D;

Out\_CAT = CAT (A, B, C, D);

Put Out\_Symbol;

Put Out\_CAT;

Run;

Data Joins;

A = "This ";

B = " is";

C = " a test ";

D = " of CAT";

Out\_Symbol = Compress (A||B||C||D);

Out\_CAT = Compress (CAT (A, B, C, D));

Out\_CATS = CATS (A, B, C, D);

Out\_CATT = CATT (A, B, C, D);

SP = '$';

Out\_CATX = CATX (SP, A, B, C, D);

Run;

Proc Print Noobs;

Var Out\_Symbol Out\_CAT Out\_CATS Out\_CATT Out\_CATX;

Run;

Data Emissions\_City;

Input UK $ US $ China $;

Datalines;

. Coal .

Gas Nuclear Petrol

Coal Gas .

. . Petrol

;

Proc Print;

Run;

Data String\_Missing;

Set Emissions\_City;

SP = '"';

Delimiter = CATX (SP, UK, US, China);

Delimiter\_Space = CATX ("", UK, US, China);

No\_Delimiter = CATT (UK, US, China);

Run;

Proc Print Noobs;

Var Delimiter Delimiter\_Space No\_Delimiter;

Run;

Data \_Null\_;

Format Week Date9. Week\_Sun Date9. Week\_Mon Date9.;

Week = INTNX ('Week', '01Jan2019'd+1, 3);

Week\_Sun = INTNX ('Week2', '01Jan2019'd+1, 3);

Week\_Mon = INTNX ('Week.2', '01Jan2019'd+1, 3);

PUT 'Week= ' Week;

PUT 'Week\_Sun = ' Week\_Sun;

PUT 'Week\_Mon = ' Week\_Mon;

Run;

Data Decimal;

Input Score;

Ceil = Ceil (Score);

Floor = Floor (Score);

Int = Int (Score);

Round = Round (Score);

Datalines;

67.454

53.34

45.23

60.80

80.4

68.5

68.9

;

Data Negative (Drop = Score);

Set Decimal;

ScoreNeg = Score\*-1;

Ceil = Ceil (ScoreNeg);

Floor = Floor (ScoreNeg);

Int = Int (ScoreNeg);

Round = Round (ScoreNeg);

Run;

Proc Print;

Var ScoreNeg Ceil Floor Int Round;

Run;

Data Fource\_Round;

Thousand = Round (1564.46, 1000);

Hundreds = Round (1564.46, 100);

Tens = Round (1564.46, 10);

Unit = Round (1564.46, 1);

Tenth = Round (1564.46, .1);

Hundredth = Round (1564.46, .01);

Run;

Data Lag;

Set Decimal (Keep = Round);

Lag1 = Lag1 (Round);

Lag2 = Lag2 (Round);

Run;

Data Class\_Scores;

Input Class $1. Score;

Datalines;

A 21

A 23

A 25

A 27

B 15

B 20

B 25

B 30

;

Data Lag\_ByGroup;

Set Class\_Scores;

By Class;

Lag1 = Lag1 (Score);

IF FIRST.Class THEN DO;

Lag1 = .;

END;

ELSE DO;

Lag1 = Lag1;

END;

RUN;

Data Logical;

Input A $ B $ C $ X Y Z;

Char = COALESCEC (A, B, C);

Num = COALESCE (X, Y, Z);

IFC = IFC (A=Char, "A", "B or C");

IFN = IFN (X=Num, "X", IFN(Y=Num, "Y", "Z"));

IFN\_alt = IFN (X=Num, 9, IFN(Y=Num, 99, 999));

Datalines;

FromA FromB FromC . 2 3

;

Data Similar;

Set Logical;

IF Char = "FromA" THEN LongWayC = "A";

ELSE LongWayC = "B or C";

IF Num = . THEN LongWayN = .;

ELSE IF Num = 2 then LongWayN = 99;

ELSE LongWayN = 999;

Run;

Proc Print Noobs;

Var Char Num IFC IFN IFN\_alt LongWayC LongWayN;

Run;

Data \_NULL\_;

Char = WhichC ("FromA", "FromB", "FromC", "FromA");

Num = WhichN (100/25, 34, 4, 40, 10);

Zero = WhichC ("FromA", "FromB", "FromC", "From A");

Put Char= / Num= / Zero=;

Run;