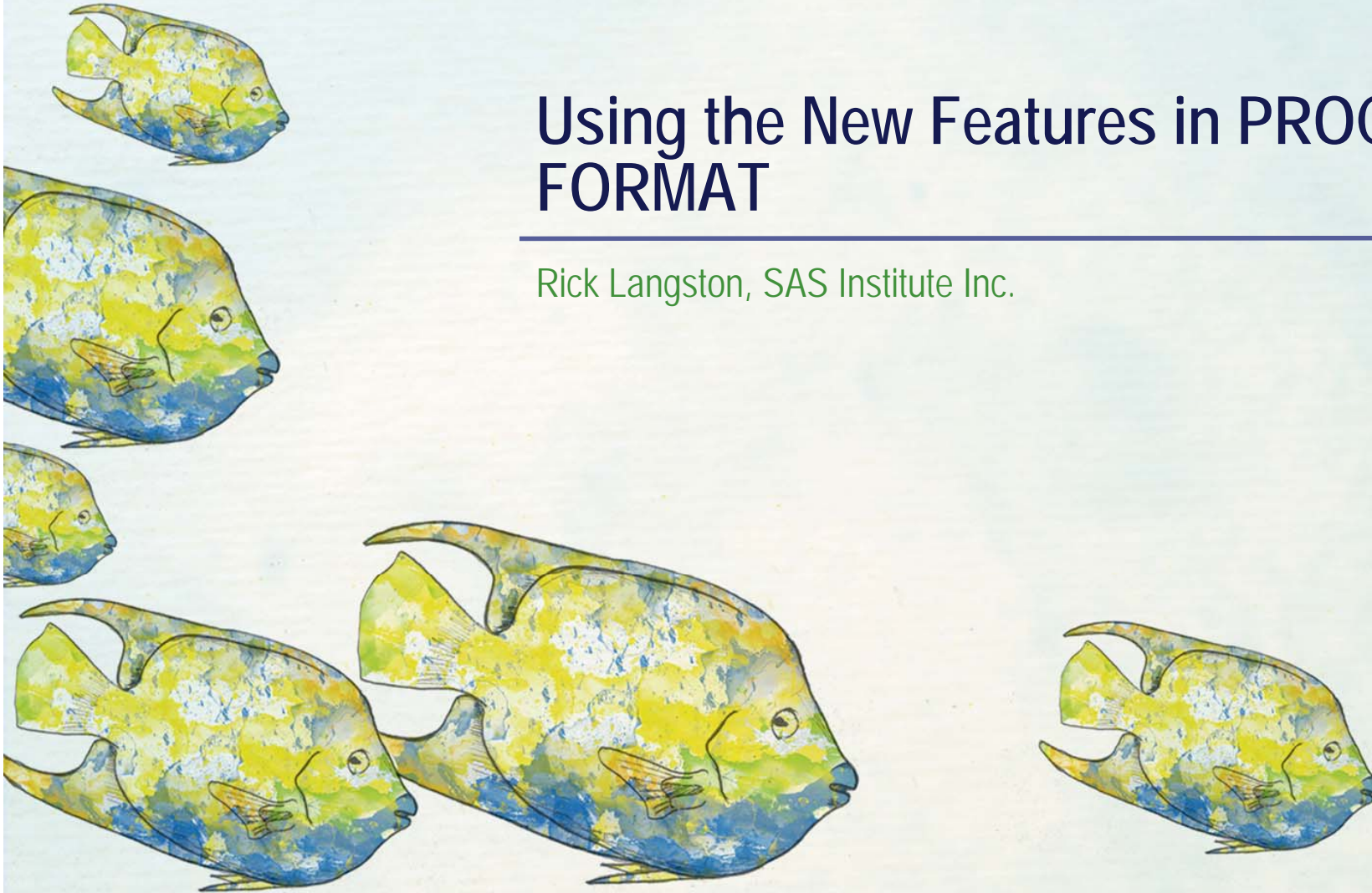


# Using the New Features in PROC FORMAT

---

Rick Langston, SAS Institute Inc.



# New PROC FORMAT Features in 9.3

- Perl regular expressions in informats
- Locale-sensitive format catalogs
- Day count directive in picture formats
- DATETIME\_UTIL
- Functions in labels



# Perl Regular Expressions



# Perl regular expressions with informats

- Consider this example

```
proc format;
```

```
    invalue xxx (default=20)
```

```
        '/(\d+):(\d\d)(?:\.(?!\d+))?/'
```

```
    (REGEXP) = [time8.];
```

- `(\d+)`: is one or more digits followed by colon
- `(\d\d)` is exactly 2 digits
- `(?:\.(?!\d+))?` indicates optional dot with 1 or more digits





```

proc format;
  invalue xxx (default=20)
    '/(\d+):(\d\d)(?:\.\d\d)?/' (REGEXP) = [time8.]
    '/\d\d\D\D\D\d\d/' (REGEXP) = [date7.]
    '/\d\d\D\D\D\d\d\d\d/' (REGEXP) = [date9.]
    other=_error_;

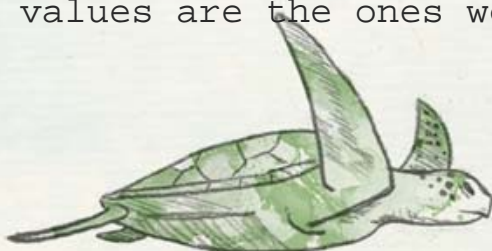
run;

data _null_; input x: xxx. @@; list; put x= x=date9. x=time10.1;
cards;
10:24 01jan02 10 abc 10:24:35 10:35:42 10:38 01jan2002
;

10:24      x=37440 x=04JUL2062 x=10:24:00.0
01jan02    x=15341 x=01JAN2002 x=4:15:41.0
10         invalid data
abc        invalid data
10:24:35   x=37475 x=08AUG2062 x=10:24:35.0
10:35:24.2 x=38124.2 x=18MAY2064 x=10:35:24.2
10:38      x=38280 x=21OCT2064 x=10:38:00.0
01jan2002  x=15341 x=01JAN2002 x=4:15:41.0

```

(red values are the ones we want)



```
proc format;  
  invalue $abc2def (default=20)  
    's/abc/def/' (REGEXPE) = _same_  
  other = _same_;  
run;  
  
data _null_; input x: $abc2def. @@; put x=; datalines;  
abc xyz  
  
x=def  
x=xyz
```

works like PRXCHANGE function



```
proc format;  
  invalue $abc2def (default=20) 's/abc/def/' (regexpe) = [def2ghi.];  
  invalue $def2ghi (default=20) 's/def/ghi/' (regexpe) = [ghi2jkl.];  
  invalue $ghi2jkl (default=20) 's/ghi/jkl/' (regexpe) = [jkl2mno.];  
  invalue $jkl2mno (default=20) 's/jkl/mno/' (regexpe) = _same_;  
run;
```

```
data _null_;  
  x=input('abc', $abc2def.);  
  put x=;  
run;
```

x=mno



# Locale-specific format catalogs



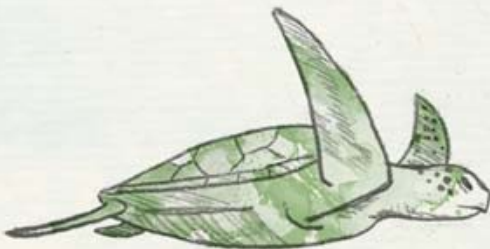


# Locale-Sensitive Format Catalogs

- New LOCALE option in PROC FORMAT
- New /LOCALE option in FMTSEARCH
- Catalogs have locale value concatenated in the name
- work.formats\_en\_us for English US
- work.formats\_fr\_fr for French France
- SAS code is the same, with a LOCALE= option change



```
proc format library=work.stdfmts;  
  value abc 1='ABC' 2='DEF';  
  run;  
  
options locale=en_us;  
  
/* creates work.myfmts_en_us */  
proc format locale library=work.myfmts;  
  value answer 1='yes' 0='no';  
  run;  
  
options locale=fr_fr;  
/* creates work.myfmts_fr_fr */  
proc format locale library=work.myfmts;  
  value answer 1='oui' 0='non';  
  run;
```



```
options fmtsearch=(work.stdfmts work.myfmts/locale);
```

```
options locale=en_us;
```

```
data _null_;  
  do x=0 to 2;  
    put x= x=abc. x=answer.;  
  end;  
run;
```

```
x=0 x=0 x=no  
x=1 x=ABC x=yes  
x=2 x=DEF x=2
```

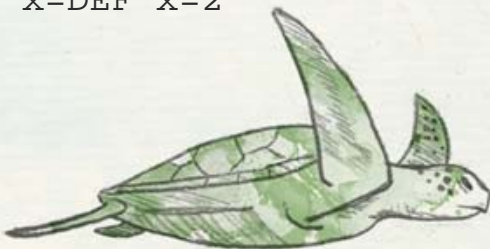
```
options locale=fr_fr;
```

```
data _null_;  
  do x=0 to 2;  
    put x= x=abc. x=answer.;  
  end;  
run;
```

```
x=0 x=0 x=non  
x=1 x=ABC x=oui  
x=2 x=DEF x=2
```



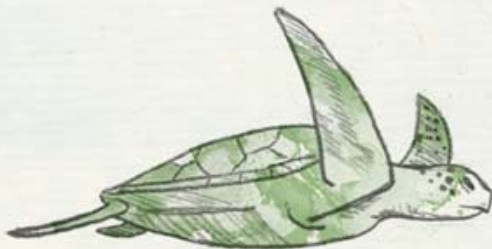
#SASGF12



# Day Count Directive



#SASGF12





```

/* Shows the %n directive for day count */
proc format;
  picture durtest(default=27)
    other='%n days %H hours %M minutes' (datatype=time);
  picture durtestx(default=27)
    other='%d days %H hours %M minutes' (datatype=time);
  picture durtesty(default=27)
    other='%d days %H hours %M minutes' (datatype=datetime);
run;

data _null_;
  start = '01jan2010:12:34'dt;
  end = '15feb2010:18:36'dt;
  diff = end - start;
  put diff=durtest. / diff=durtestx. / diff=durtesty.;
run;

```

produces

```

diff=45 days 6 hours 2 minutes (what we want)
diff=0 days 6 hours 2 minutes (ignores days)
diff=15 days 6 hours 2 minutes (thinks it's 15feb1960)

```



# DATETIME\_UTIL



#SASGF12



```
/* The utility industry often wants to reference a midnight date
   to be 24:00:00 instead of 00:00:00. The new DATATYPE= value
   DATETIME_UTIL allows this. */
```

```
proc format;
  picture abc (default=19)
    other='%Y-%0m-%0d %0H:%0M:%0S' (datatype=datatype_util);
run;
```

```
data _null_;
  x = '01nov2008:00:00:00'dt; put x=abc.;
  x = '01nov2008:00:00:01'dt; put x=abc.;
run;
```

produces

```
x=2008-10-31 24:00:00
x=2008-11-01 00:00:01
```



# Functions as labels

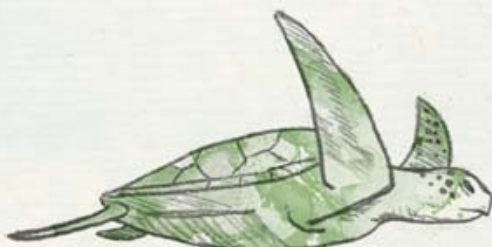




# Functions-as-labels

```
proc format;  
    value abc (default=10)  
        other=[myfunc( )];  
run;
```

- Bracketed name with parentheses indicates function
- Function myfunc takes one argument (numeric/character) and returns numeric/character
- Use DEFAULT= to ensure proper widths
- Functions can be SAS-supplied or ones written with PROC FCMP



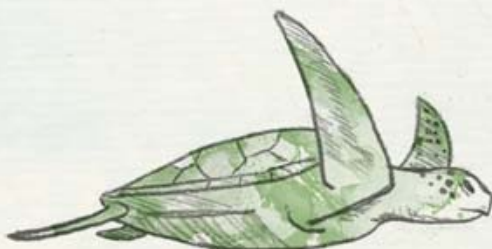
# Using the JULDATE7 function

```
proc format;  
value myexmp '01nov2008'd - '30nov2008'd = [juldate7()]  
other=[date9.];  
run;  
data _null_;  
x='15nov2008'd; put x=myexmp.;  
x='01dec2008'd; put x=myexmp.;  
run;
```

produces

x=2008320

x=01DEC2008



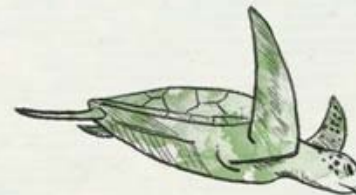
# Creating a function using FCMP

```
proc fcmp outlib=work.functions.smd;  
  function ctof(c) $;  
    return(cats(((9*c)/5)+32, '° F'));  
  endsub;  
  
  function ftoc(f) $;  
    return(cats((f-32)*5/9, '° C'));  
  endsub;  
run;
```

```
options cmplib=(work.functions);  
data _null_;  
  f=ctof(100); put f=;  
  c=ftoc(212); put c=;  
run;
```

produces

```
f=212° F  
c=100° C
```



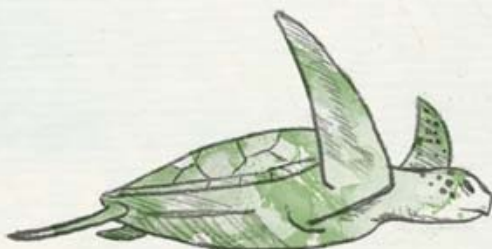
# Using the function as a label

```
proc format;  
  value ctof (default=10) other=[ctof()];  
  value ftoc (default=10) other=[ftoc()];  
run;
```

```
data _null_;  
  c=100; put c=ctof.;  
  f=212; put f=ftoc.;  
run;
```

produces

```
c=212° F  
f=100° C
```





# TSGN function defined

```
proc fcmp outlib=work.functions.smd;  
  
  function tsgn(text $);  
    x = input(text, trailsgn10.);  
    x = x/100;  
    return(x);  
  endsub;  
run;
```

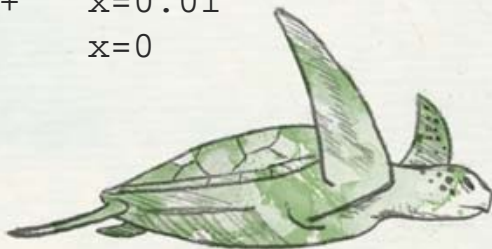


# Using as an informat

```
options cmplib=(work.functions);  
proc format; invalue tsgn(default=10) other=[tsgn()];  
  
data _null_;  
    input x: tsgn.; put x=; list; cards;  
1  
1-  
12-  
123-  
123+  
1+  
0  
run;
```

produces

```
1      x=.01  
1-     x=-0.01  
12-    x=-0.12  
123-   x=-1.23  
123+   x=1.23  
1+     x=0.01  
0      x=0
```



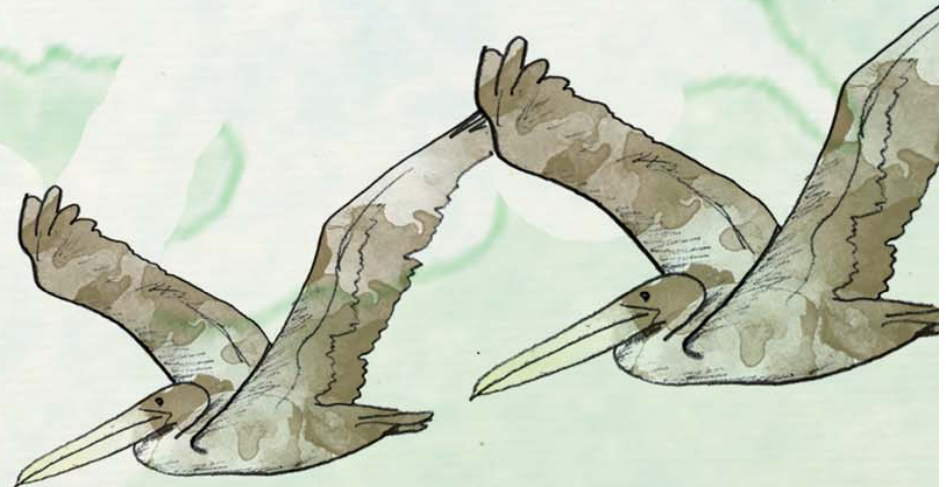
# Contact Info

- Rick Langston, SAS Institute Inc.
- [Rick.Langston@sas.com](mailto:Rick.Langston@sas.com)





# 2012 Orlando Florida



#SASGF12