



# Programming is Done By Human Beings

Ray McKinnis

# Computer Language And Human Being Language

Computer Language and Human Being Language

SAS Understands Both

# Some Basic Principles Incorporated into SAS

1. Simple and self-documenting. That means understandable to human beings.
2. Let computer do what it does best and humans do best what they do best.
3. Errors that the computer doesn't understand-explained in language that a human being can understand.
4. And continue scanning code (with obs=0) to see if there are any other mistakes.
5. Everything a human being needs to know to understand the logic and purpose of the file is in the file itself.

# Self-Documenting

```
libname sdtm  
'\\server124\fileabc\sdtm';  
  
data ae1;  
    set sdtm.adae;  
    keep usubjid aeterm aecod  
aesae;  
run;
```

Good Computer Language  
Missing Human Language

**%include;**

**data ae1;**

**set sdtm.adae;**

**keep usubjid aeterm**

**aecod aesae;**

**run**

Good Computer Language  
Good Human Language

```
%includeit;  
libname sdtm  
'\\server124\\fileabc\\sdtm';  
  
data ae1;  
    set sdtm.adae;  
    keep usubjid aeterm aecod aesae;  
run;
```

WHY Would Anyone Do This?  
This can get you into trouble.

```
*Get Data;  
proc sort data=sdtm.lb out=lb;  
run;
```

```
*Create dataset;  
data lb1;  
    set lb;  
run;
```

# Brilliant SAS Code Speaking Human Language

**if inae and inadsl;**

What Happened?  
The Human Language has Disappeared  
This is Meaningless  
To a Human Being

**if a and b;**

Not only is this meaningless to a human being, its meaning changes throughout the program.

Good  
Computer Language  
and Human Language

**data ae;**

# Computer Language ONLY

**Data data1;**

**or**

**Data &data1;**

# Brilliant SAS Feature

Self-Documenting at its  
**BEST**

**Title1 ‘Listing 14.3.10’;**

**Title2 ‘Laboratory Results from the First  
Two Periods’;**

**Title3 ‘By Gender and Age’;**

**proc print data=lab;**

**var usubjid partam sex lbresult;**

**run;**

# Human Language is Absent and So is Understanding

```
%titles;  
proc print data=lab;  
  var usubjid partam sex lbresult;  
run;
```

\*Human beings need some  
help in understanding this;

**%dates(indate=,outdate=);**

\*Macro for converting a date  
from ddMONyyyy into dd Month  
yyyy format;

**%dates(indate=,outdate=);**

\*Macro for converting a date from  
ddMONyyyy into dd Month yyyy format;  
And too much more!

**%dates(indate=,outdate=,  
type=, source=, missing=,  
use=, format=, error=, level=);**

GOOD SAS CODE

**aedur = aeendt – aestdt;**

# CONFUSING SAS CODE

## Why Do This?

```
if aeendt>. and aestdt>. then  
    aedur = aeendt – aestdt;
```

GOOD, CLEAR, CLEAN SAS CODE

**trtn = 1\*trtc;**

SO WHY MAKE IT CONFUSING  
Like This?

**trtn = input(trtc,best.)**

Which of These is  
Self Documenting?

**PROC PRINT**

**PROC SORT**

**PROC MEANS**

**PROC FREQ**

**PROC UNIVARIATE**

PROC SQL?

# SQL CODE

```
proc sql;  
create table case2 as select a.*, b.visitnum  
as num  
from case1(where=(visitnum=.) as a  
left join  
sv as b  
on a.usubjid=b.usubjid and  
substr(a.&date.,1,10)=b.svstdtc;  
quit;
```

# SAS Code

```
data unsched; set case1; run;
```

```
proc sort data=unsched; by usubjid; run;
```

```
data sv1; set sv; run;
```

```
proc sort data=sv1; by usubjid; run;
```

```
data unsched1;  
  merge unsched(in=inunsc) sv1(in=insv);  
by usubjid;  
run;
```

# The Point of This Presentation is Simply:

Whenever you use SAS programming, put as much effort into making sure that it is understandable by another human being as into making sure the computer will understand it.

Otherwise you are dancing on only one leg.

SAS was developed so that you could easily use both legs:

**a computer and a human being.**

# The Kicker:

If done simply, I believe it could save hundreds of hours and thousands of dollars in the business of drug development.