

Paramétrisation 101, 201 et 301

```
\\|///  
\\ - - //  
( @ @ )  
ooo ( ) ooo
```

**Donnez souplesse et versatilité
à vos traitements SAS®**

```
ooo  
ooo ( )  
( ) /  
( ( /  
\\ ) -  
-
```



Alain Voyer
Utilisateur SAS® curieux
Club des utilisateurs SAS de Québec, 2 juin 2016

Avertissement

Présentation qui s'adresse
aux « codeux », ces
personnes qui aiment
escalader des montagnes
de code et de données.



Le K2

Source: CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=14038>

Démarche

101 – Créer et utiliser des variables d'environnement

« *To %LET or not to %LET, that is the question.* »

Ham%let



201 – Transformer des variables d'environnement

Montage en LEGO® de variables d'environnement
avec et sans %LET

On ajoute une pincée de SQL



301 – Exporter des variables d'environnement

Utiliser ses variables d'environnement
en sortie



3

101 – Créer et utiliser des variables d'environnement...

%LET est une manière courante de paramétrer un pgm SAS.

Simple et efficace. « On est en voiture ».

Connaissance minimale du contexte %MACRO (macro facility).

Voir présentations de:

- François Beauchesne, mai 2015 (conférence #18)
- Mathieu Gaouette, 1er mai 2014 (conférence #16)
- François Bourbeau, Jean Hardy, oct. 2013 (conférence #15).

On regroupe généralement les paramètres au début d'un traitement.

On ne modifie que ces paramètres pour orienter notre traitement.

On peut mettre tous les %LET dans un petit fichier et importer son contenu avec un %INCLUDE

4

101 – Créer et utiliser des variables d'environnement...

Approche classique (« *To %LET...* »):

```
1  %let DATEP=2016-06-05;
2  %let PG=4;
3  %let TEST=OUI;

4  %put &=DATEP; /* Nouveau 9.3 */
DATEP=2016-06-05
5  %put &=PG;
PG=4
6  %put &=TEST;
TEST=OUI

7  %put _user_;
GLOBAL DATEP 2016-06-05
GLOBAL PG 4
GLOBAL TEST OUI
```



« On est en voiture »

101 – Créer et utiliser des variables d'environnement...

Une autre approche pour non familiers du contexte %MACRO propose de transformer automatiquement en variables d'environnement une séquence paramétrique de type:

"param1=valeur1,param2=valeur2,...,etc."

La séquence peut être passée par SYSPARM ou par fichier.


La transformation s'effectue avec une petite étape DATA invariante à mettre au début des traitements.

En la plaçant dans une bibliothèque de code source, on peut l'utiliser ensuite avec un %INCLUDE dans plusieurs traitements.

101 – Créer et utiliser des variables d'environnement...

Approche: «... or not to %LET... »:

```
1      ;options sysparm="DATEP=2016-06-05,PG=4,TEST=OUI";
2
3      ;data _null_;
4
5
6
7
8
9
10
11
12
13
14      ;run;
```



NOTE: The DATA statement used 0.01 CPU seconds and 20196K.

```
15      %put _user_ ;
GLOBAL DATEP 2016-06-05
GLOBAL PG 4
GLOBAL TEST OUI
```

7

101 – Créer et utiliser des variables d'environnement...

Avant d'ouvrir le capot de l'étape DATA regardons quelques éléments de la mécanique du moteur.

Rappel: Comment découper une chaîne de caractères?

Exemple avec la fonction SCAN():

```
1      ;data _null_;
2      Chaîne = "primo/secundo/tertio"; /* longueur=20 */
3      Un      =scan(Chaîne,1,"/");
4      Deux    =scan(Chaîne,2,"/");
5      Trois   =scan(Chaîne,3,"/");
6      NbParam =countw(Chaîne,"/");
7      putlog un= deux= trois= NbParam;
8      ;run;
```

Un=primo Deux=secundo Trois=tertio NbParam=3

La question qui tue: Quelle est la longueur des variables un, deux et trois? 20 ou 200.

8

101 – Créer et utiliser des variables d'environnement...

Avant d'ouvrir le capot de l'étape DATA regardons quelques éléments de la mécanique du moteur.

Rappel: Comment créer une variable d'environnement?

Exemple avec %LET ...

```
1      %let MaVarEnv=primo;
```

...et la routine CALL SYMPUT():

```
1      ;data _null_;  
2          call symput("MaVarEnv","primo");  
3      ;run;
```

```
4      %put _user_;
```

```
GLOBAL MAVARENV primo
```

101 – Créer et utiliser des variables d'environnement...

Avant d'ouvrir le capot de l'étape DATA regardons quelques éléments de la mécanique du moteur.

Rappel: Comment créer une variable d'environnement?

Exemple 2 avec la routine CALL SYMPUTX():

```
1      ;data _null_;  
2          Chaîne="primo";  
3          NomVar="MaVarEnv";  
4          call symputx(NomVar,Chaîne);  
5      ;run;
```

```
6      %put _user_;
```

```
GLOBAL MAVARENV primo
```

101 – Créer et utiliser des variables d'environnement...

Regardons sous le capot la mécanique dans son ensemble.

Approche: «... or not to %LET... »:

```
1  options sysparm="DATEP=2016-06-05,PG=4,TEST=OUI";
2
3  ;data _null_;
4      length SysParm Expression NomParam Valeur $ 200;
5      SysParm = symget("sysparm");
6
7      do i=1 to countw(SysParm, ",");
8          Expression = scan(SysParm,i,"");
9          NomParam   = scan(Expression,1,"=");
10         Valeur      = scan(Expression,2,"=");
11         call symputx(NomParam,Valeur);
12     end;
13
14 ;run;
15 %put _user_;
GLOBAL DATEP 2016-06-05
GLOBAL PG 4
GLOBAL TEST OUI
```

Source: Professional SAS® User Interfaces, Rick Aster, 1992, Windcrest/McGraw-Hill



11

101 – Créer et utiliser des variables d'environnement - fin

Extra: Une petite validation est ajoutée (fonction nvalid)

```
1  options sysparm="DATEP=2016-06-05,PG=4,TEST=OUI";
2
3  ;data _null_;
4      length Sysparm Expression NomParam Valeur $ 200;
5      Sysparm = symget("sysparm");
6
7      do i=1 to countw(Sysparm, ",");
8          Expression = scan(Sysparm,i,"");
9          NomParam   = scan(Expression,1,"=");
10         Valeur      = scan(Expression,2,"=");
11         if nvalid(NomParam,"V7")
12             then call symputx(NomParam,Valeur,"g");
13             else putlog "WARNING: (PROG) PARAM: Nom Invalide: " NomParam;
14     end;
15
16 ;run;
```

WARNING: (PROG) PARAM: Nom invalide: TES?

ERROR: Symbolic variable name TES? must contain only letters, digits, and underscores.



12

Sagesse ancestrale

Thus spake the master programmer:

***« Though a program be but
three lines long, someday it
will have to be maintained. »***

© The Tao of Programming, Geoffrey James

13



201 – Transformer des variables d'environnement...

Montage en LEGO® de variables d'environnement avec %LET

```
1  %let A=30;  
2  %let B=%eval(&A*2);  
3  %let C=&A ET &B;  
4  %let D=CLUB SAS &C;  
5  %put _user_;
```

```
GLOBAL A 30  
GLOBAL B 60  
GLOBAL C 30 ET 60  
GLOBAL D CLUB SAS 30 ET 60
```



14

201 – Transformer des variables d'environnement...

Montage en LEGO® de variables d'environnement sans %LET

```

1  options sysparm='A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C';
2
3  ;data _null_;
4      length SysParm Expression NomParam Valeur $ 200;
5      SysParm = symget("sysparm");
6
7      do i=1          to          countw(SysParm, ",");
8          Expression =          scan(SysParm,i,"");
9          NomParam   =          scan(Expression,1,"=");
10         Valeur      = resolve(scan(Expression,2,"=")) ;
11         call symputx(NomParam,Valeur);
12     end;
13
14 ;run;
15 %put _user_;

```

```

GLOBAL A 30
GLOBAL B 60
GLOBAL C 30 ET 60
GLOBAL D CLUB SAS 30 ET 60

```



15

201 – Transformer des variables d'environnement...

Montage en LEGO® de variables d'environnement sans %LET

On voit la transformation à chaque itération de la boucle DO grâce à la fonction RESOLVE()

Enlève les traces de & et de %



Liste des paramètres

i	SysParm	Expression	NomParam	Valeur
1	A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C	A=30	A	30
2	A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C	B=%eval(&A*2)	B	60
3	A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C	C=&A ET &B	C	30 ET 60
4	A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C	D=CLUB SAS &C	D	CLUB SAS 30 ET 60

16

201 – Transformer des variables d'environnement...

Montage en LEGO® de variables d'environnement avec %LET

Exemple extrait d'un pgm réel de 563 lignes:

```
%let dtf      =%sysfunc(left(%sysfunc(date()),ymmdd10.));
```

```
%let NomDoc1  =tgl0215e;
```

```
%let TitreDoc =_Admin_CMI_TSM;
```

```
%let ftpPDF   =&NomDoc1.&TitreDoc..pdf;
```

```
title3 "Production de &sysjobid (pgm %upcase(&NomDoc1))";
```

```
filename pdfout1 ftp "&dtf._&ftpPDF" etc...;
```

```
filename pdfout2 ftp      "&ftpPDF" etc...;
```

```
GLOBAL NOMDOC1 tgl0215e
```

```
GLOBAL TITREDOC _Admin_CMI_TSM
```

```
GLOBAL FTPPDF tgl0215e_Admin_CMI_TSM.pdf
```

Et &dtf._&ftpPDF se résout comme: 2016-04-14_tgl0215e_Admin_CMI_TSM.pdf



17

201 – Transformer des variables d'environnement...

On ajoute une pincée de SQL... à la prochaine acétate.

D'abord lire un fichier avec des numéros de dossiers

```
%let lVar=6; /* longueur convenue de variables */
```

```
data tNoDoss;
```

```
  input  NoDossNum &lVar.; /* Lire la variable en numérique */
```

```
datalines;
```

```
123
```

```
4567
```

```
89012
```

```
444444
```

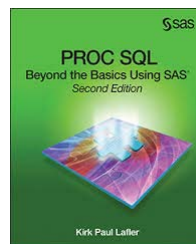
```
123
```

```
4567
```

```
89012
```

```
;
```

```
run;
```



18

201 – Transformer des variables d'environnement...

On ajoute une pincée de SQL... pour « empiler » les valeurs numériques lues dans une variable d'environnement.

```
;proc sql; *noprint;
  select distinct(NoDossNum)
    into :LstNoDossNum separated by ","
    from tNoDoss;
;quit;
```

NoDossNum

123

4567

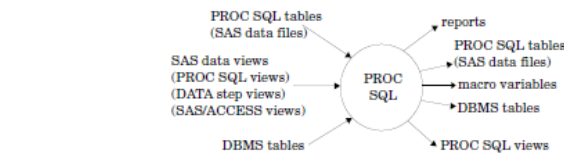
89012

444444

;

```
%put _user_;
```

GLOBAL LSTNODOSSNUM 123,4567,89012,444444



19

201 – Transformer des variables d'environnement...

On ajoute une pincée de SQL... pour « emplir » les valeurs en mode caractère dans une variable d'environnement.

```
;proc sql; *noprint;
  select distinct(quote(put(NoDossNum,z&lVar..))
    as NoDossCar length=%eval(&lVar+2)
    into :LstNoDossCar separated by ","
    from tNoDoss;
;quit;
```

NoDossCar

"000123"

"004567"

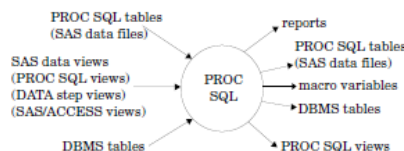
"089012"

"444444"

;

```
%put _user_;
```

GLOBAL LSTNODOSSCAR "000123","004567","089012","444444"



20

201 – Transformer des variables d'environnement...

On ajoute une pincée de SQL... pour lister les variables d'environnement avec le dictionnaire.

```
2      ;options sysparm='A=30,B=%eval(&A*2),C=&A ET &B,D=CLUB SAS &C';
3
4      %include sasmac(paramr);
17
18      ;title1 "Liste de variables d'environnement";
19
20      ;proc sql;
21          select name           label="Nom de variable"
22                 ,length(value) label="Longueur"
23                 ,value        label="Valeur de la variable"
24      from    dictionary.macros
25      where   (scope="GLOBAL" and
26              (name not like 'SQL%' and
27               name not like 'SYS%' )
28              );
29      ;quit;
```

Préalable avec SAS Studio:
%let pathf=/folders/myshortcuts/MyFolder/;
filename sasmac "&pathf";

21

201 – Transformer des variables d'environnement...

On ajoute une pincée de SQL... pour lister les variables d'environnement.

Résultat:

Liste de variables d'environnement

Nom de variable	Longueur	Valeur de la variable
A	2	30
B	2	60
C	8	30 ET 60
D	17	CLUB SAS 30 ET 60

22

201 – Transformer des variables d'environnement - fin

On ajoute une pincée de SQL. Heu non, c'est assez pour le SQL.
Les variables d'environnement ainsi créées peuvent ensuite être utilisées dans une étape DATA comme ceci:

```
;data TableResultat;  
    /* Sélection avec un WHERE et un IN... avec                */  
    set TablePrincipale (where=(NoDeDossierCar in(&LstNoDossCar)));  
    /* ou                                                        */  
    set TablePrincipale (where=(NoDeDossierNum in(&LstNoDossNum)));  
;run;  
  
Rappel concernant les variables d'environnement créées:  
GLOBAL LSTNODOSSCAR "000123","004567","089012","444444"  
GLOBAL LSTNODOSSNUM 123,4567,89012,444444
```

23

Sagesse ancestrale

Thus spake the master programmer:

**« *A well-written program is
its own heaven; ☺
a poorly-written program is
its own hell. ☹* »**

© The Tao of Programming, Geoffrey James

24



301 – Exporter des variables d'environnement...

Étape 1: Lecture de ce qu'on veut transformer: Des énoncés quelconques sous forme de gabarit (pattern)

Exemple: Commande DOS d'appel à un script PowerShell

```
;data tENONinp;
  length LigneInp $ 80;

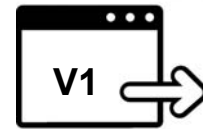
  infile datalines;

  input @001 LigneInp $char080.;

  if LigneInp in: ( "(", ")" ) then delete;

;datalines;

CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\&env\PS\TSM_BK.PS1
-HOPT dsm_nas_&opt._&v1&v2..opt
-HCHaine TGN&ch
-HNASBK Y
;
;run;
```



25

301 – Exporter des variables d'environnement...

Étape 2: Étape de transposition des variables d'environnement dans le gabarit (pattern)

Exemple: Commande DOS d'appel à un script PowerShell

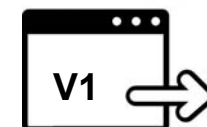
```
;data _null_;
  length Expression NomParam Valeur LignePrm LigneOut $ 80;

  infile datalines;
  input @001 LigneLue $080.; /* Ligne lue et justifiée à gauche */

  LignePrm=scan(LigneLue,1," "); /* Ligne avec paramètres seulement */
  if LignePrm in: ( "(", ")" " " " " ) then delete;

  do i=1 to countw(LignePrm, " ");
    Expression = scan(LignePrm,i," ");
    NomParam = scan(Expression,1,"=");
    Valeur = scan(Expression,2,"=");
    call symputx(NomParam,Valeur);
  end;

suite...
```

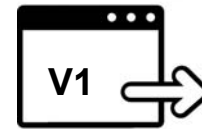


26

301 – Exporter des variables d'environnement...

Étape 2: Suite... Les ensembles de paramètres bornées par le délimiteur « /fin/ »

```
if find(LigneLue,"/fin/","i") then
    do i=1 to NbEnon;
        set tEnonInp point=i nobs=NbEnon;
        LigneOut = resolve(LigneInp);
        file fENONout noprint notitles mod;
        put @001 LigneOut $char080.;
    end;
;datalines;
) Parametres a transformer en variables d'environnement
env=PROD
  ch=S17,v1=2P,v2=B5,hno=02,opt=SQL      /fin/
  ch=S14,v1=2P,v2=B6,hno=02,opt=ORACLE   /fin/
env=TEST
  ch=R01,v1=1M,v2=01,hno=01,opt=FIC      /fin/
;
;run;
```



27

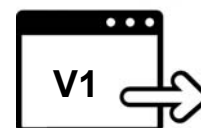
301 – Exporter des variables d'environnement...

Résultat de la conversion:

```
CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\PROD\PS\TSM_BK.PS1
-HOPT dsm_nas_SQL_2PB5.opt
-HCHaine TGNS17
-HNASBK Y

CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\PROD\PS\TSM_BK.PS1
-HOPT dsm_nas_ORACLE_2PB6.opt
-HCHaine TGNS14
-HNASBK Y

CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\TEST\PS\TSM_BK.PS1
-HOPT dsm_nas_FIC_1M01.opt
-HCHaine TGNR01
-HNASBK Y
```



28

301 – Exporter des variables d'environnement - fin

En résumé:

Variables et valeurs:

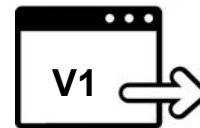
```
env=PROD  
ch=S17,v1=2P,v2=B5,hno=02,opt=SQL      /fin/
```

Gabarit de code:

```
CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED  
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\&env\PS\TSM_BK.PS1  
-HOPT dsm_nas_&opt._&v1&v2..opt  
-HCHaine TGN&ch  
-HNASBK Y
```

Résultat:

```
CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED  
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\PROD\PS\TSM_BK.PS1  
-HOPT dsm_nas_SQL_2PB5.opt  
-HCHaine TGNST7  
-HNASBK Y
```



29

301 – Exporter des variables ... Version 2...

Étape 1: Lecture de ce qu'on veut transformer: Des énoncés quelconques sous forme de gabarit (pattern)

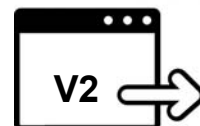
Exemple: commande DOS d'appel à un script PowerShell

```
;data tENONinp;  
length LigneInp $ 80;  
  
infile datalines;  
  
input @001 LigneInp $char080.;  
  
if LigneInp in:( "(", ")" ) then delete;
```

datalines;

```
CMD /C POWERSHELL -EXECUTIONPOLICY REMOTESIGNED  
-FILE \\AGEMNASPROD\GEM\SANDEF\PS_GEM\&env\PS\TSM_BK.PS1  
-HOPT dsm_nas_&opt._&v1&v2..opt  
-HCHaine TGN&ch  
-HNASBK Y
```

```
;  
;run;
```



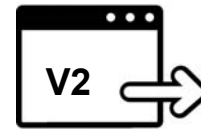
30

301 – Exporter des variables ... Version 2...

Étape 2: Macro qui effectue la transposition des variables d'environnement dans le gabarit (pattern)

Exemple: commande DOS d'appel à un script PowerShell

```
%macro convertir;  
;data _null_;  
  length LigneInp LigneOut $ 80 ;  
  
  set tENONinp;  
  
  if find(LigneInp, '&')  
    then LigneOut = resolve(LigneInp);  
    else LigneOut = LigneInp;  
  
  file fENONout noprint notitles mod;  
  put @001 LigneOut $char080.;  
;run;  
%mend;
```

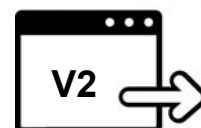


31

301 – Exporter des variables ... Version 2...

Étape 3.1: Étape DATA qui appelle la macro pour chaque ensemble de paramètres.

```
;data _null_;  
  length Expression NomParam NomValeur LignePrm LigneLue $ 80;  
  
  infile datalines;  
  input @001 LigneLue $080.;  
  
  LignePrm=scan(LigneLue,1," "); /* Ligne avec paramètres seulement */  
  if LignePrm in:( "(", ")" " " " " ) then delete;  
  
  do i=1 to count(LignePrm,"")+1;  
    Expression = scan(LignePrm,i,"");  
    NomParam = scan(Expression,1,"=");  
    NomValeur = scan(Expression,2,"=");  
    call symputx(NomParam,NomValeur);  
  end;  
  
  if find(LigneLue,"/fin/","i") then do;  
    rc = dosubl('%convertir');  
    putlog "INFO: (PROG) dosubl rc: " rc;  
  end;
```

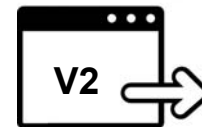


32

301 – Exporter des variables ... Version 2 - fin

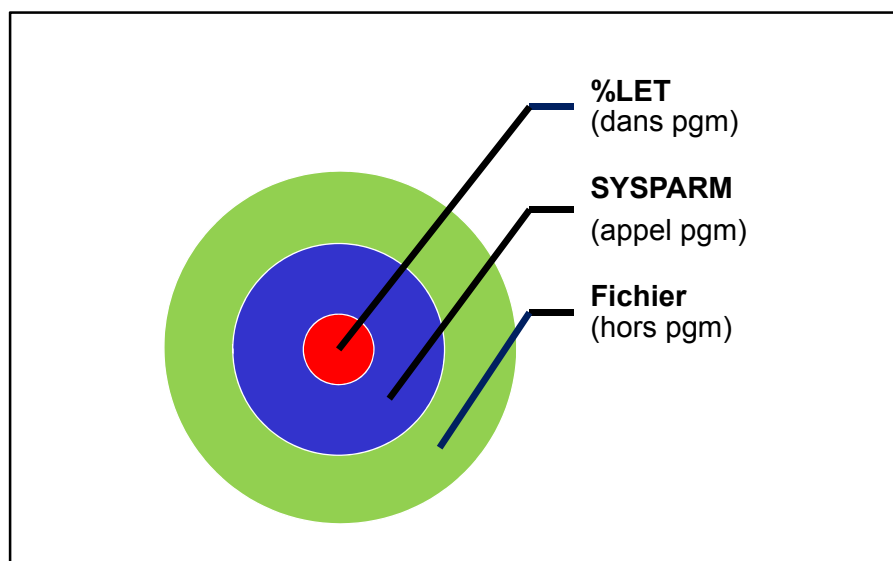
Étape 3.2: Les ensembles de paramètres bornés par le délimiteur « /fin/ »

```
;datalines;  
  
) Paramètres à transformer en variables d'environnement  
env=PROD  
  
    ch=S17,v1=2P,v2=B5,hno=02,opt=SQL           /fin/  
    ch=S14,v1=2P,v2=B6,hno=02,opt=ORACLE        /fin/  
    ch=R01,v1=1M,v2=01,hno=01,opt=FIC           /fin/  
;  
;run;
```



33

Hiérarchie des paramètres



34

Quelques sources d'inspiration

Professional SAS® User Interfaces

Rick Aster, 1992, Windcrest/McGraw-Hill (Page 103)

Document d'où provient la petite étape DATA invariante que j'ai adaptée.

The RESOLVE Function – What is It Good For?

Ian Whithlock, Westat (NESUG 98, Paper P088)

Document qui m'a donné plus tard l'idée d'utiliser la fonction RESOLVE.

A Better SYSIN Than SYSIN: Instream Files on Any Platform

Ted Conway, Ted Conway Consulting, Inc., Chicago, IL (SUGI 30, Paper 034-30)

Document dont j'aurais aimé prendre connaissance avant de créer mon approche d'exportation de variables d'environnement avec la fonction RESOLVE.

Undocumented and Hard-to-find PROC SQL® Features

Kirk Paul Lafler, Software Intelligence Corp., Spring Valley, CA (NESUG 2007, Paper bb07)

Document qui explique l'utilisation de la clause INTO de PROC SQL.

The TAO of Programming

Geoffrey James, 1987, Info Books, Santa Monica, CA

Et la documentation SAS, les groupes d'utilisateurs SAS (SUGI, SGF, Club SAS), etc.

35

Sagesse ancestrale

Thus spake the master programmer:

« After three days without programming, life become meaningless. »

© The Tao of Programming, Geoffrey James

36

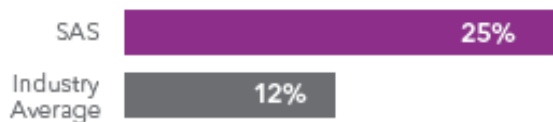


Sagesse moderne ?

Pour s'améliorer en SAS peut-être est-il opportun de consacrer une partie de son temps en apprentissage...

Comme SAS Institute consacre 25% de ses revenus en R/D, serait-il approprié de consacrer entre 12% et 25% de son temps de développement SAS en formation, lecture, participation à des clubs SAS, etc. ?

Revenue Reinvested in R&D



Source: Company Overview 2015 Annual report, p. 13

37



Merci!

*Questions ?
Commentaires ?
Suggestions ?*

Contact:

```
1 ;data _null_;  
2   courriel=lowercase(reverse("Alain")||reverse("Voyer"))||"@gmail.com";  
3   putlog courriel=;  
4 ;run;
```

courriel=nialareyov@gmail.com

Marques de commerce

SAS et tous les autres noms de produits et services de SAS Institute Inc. sont des marques de commerce de SAS Institute Inc. aux États-Unis et dans les autres pays. ® indique l'inscription au États-Unis. Les autres noms de produits ou marques sont des marques déposées ou des marques de commerce de leurs compagnies respectives.

38