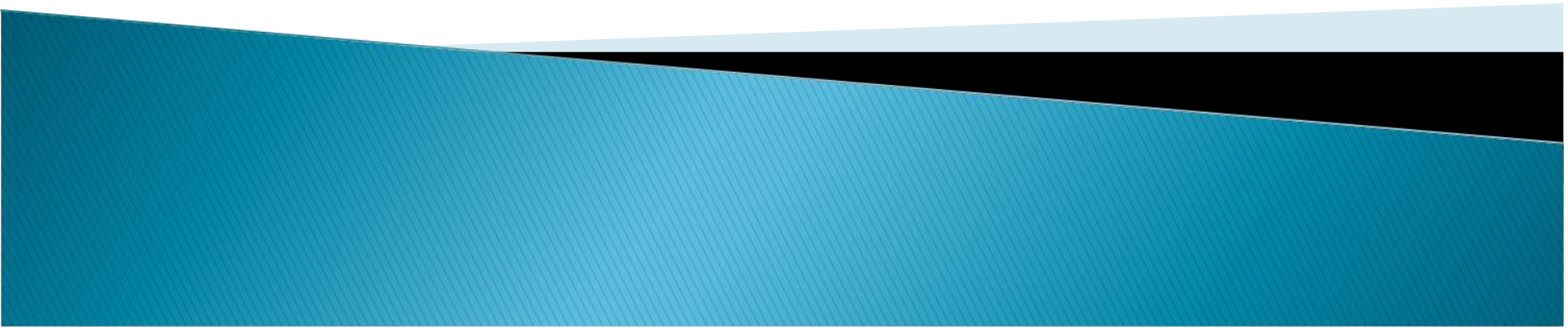
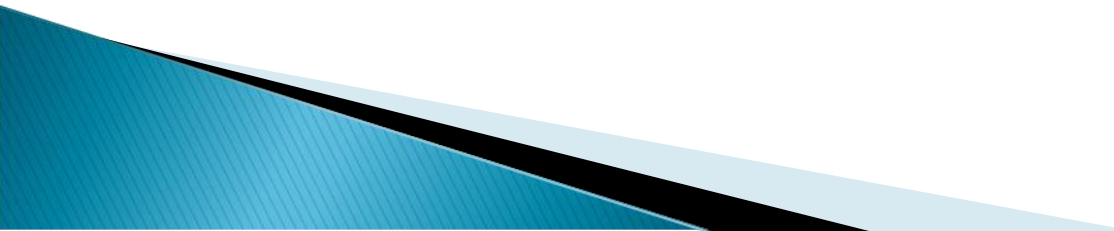


# Ce qu'on doit savoir sur le langage Macro de SAS

François Bourbeau, retraité  
Jean Hardy, Services Conseils Hardy Inc.  
Le 30 octobre 2013.

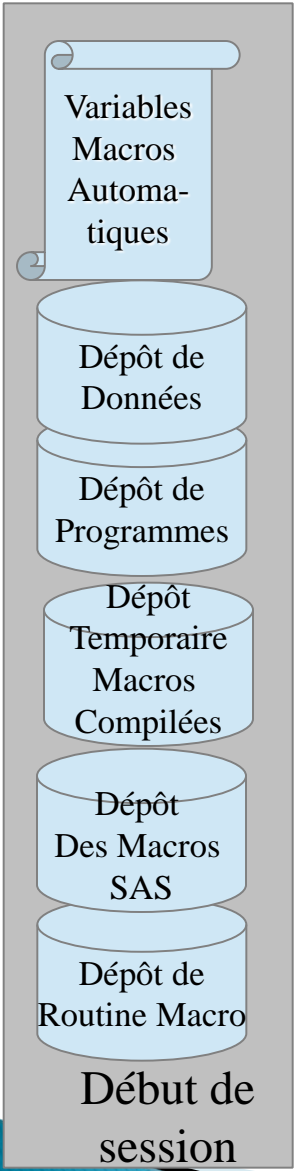


# Contenu de la présentation

1. Introduction
  2. Emploi des variables Macros et affectations simples
  3. Les Macros SAS
  4. Conclusion
- 

# 1. Introduction

- Le langage Macro sert à :
  - Passer du code SAS (texte) au logiciel sas.
  - Réduire la quantité de code SAS.
  - Se faire des utilitaires
- Carte de l'environnement du logiciel Macro Facility
  - Pour comprendre comment ça fonctionne.



Options macros

Éditeur SAS

Scanner de mots

Variables macros

M  
F

Compilation  
Exécution

S  
A  
S

Compilation  
Exécution

Ligne du temps

# 1.3 Explication de la carte

La carte est composé de 14 éléments :

Dont 5 sont connus :

- Dépôt de données
- Dépôt de programme
- Éditeur SAS
- Compilation de programmes SAS
- Exécution de programmes SAS.

Les deux composantes à découvrir vont nous permettre d'utiliser les 9 derniers éléments:

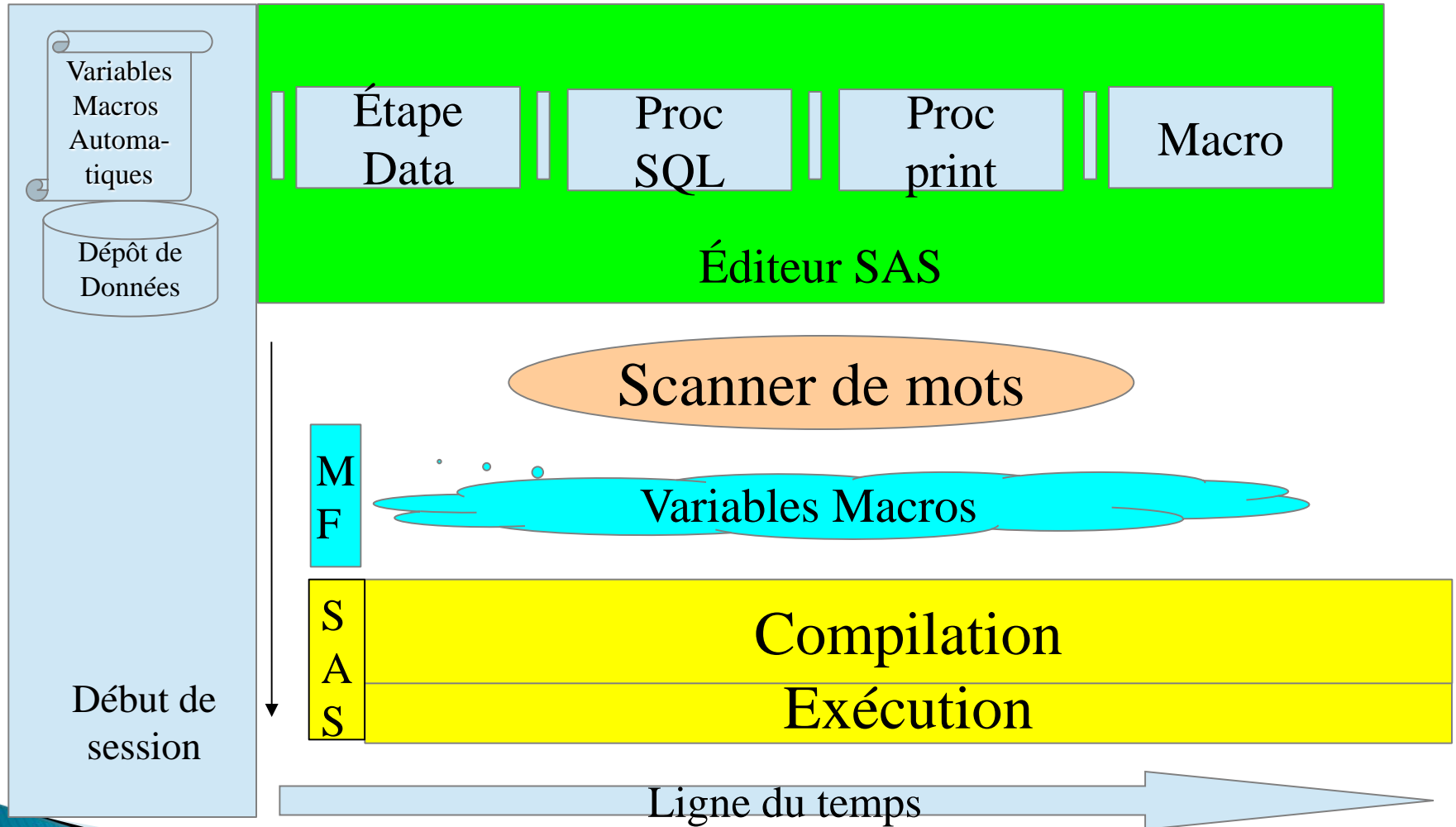
Les variables Macros

Le langage Macros ou macro Facility ( MF) qui fait parti de SAS de Base.

## 2. À quoi servent les variables MACRO?

Elles servent à fournir des chaînes de caractères.

# 2. Les variables Macros



## 2.1 Explication de la carte variables macros

Des 9 éléments de la carte de l'environnement  
MACRO

3 éléments sont utilisés par les variables  
MACRO.

- Variables MACRO
  - Scanner de mots
  - Variables MACRO automatiques.
- 



# 2.2 Emploi d'une variable Macro

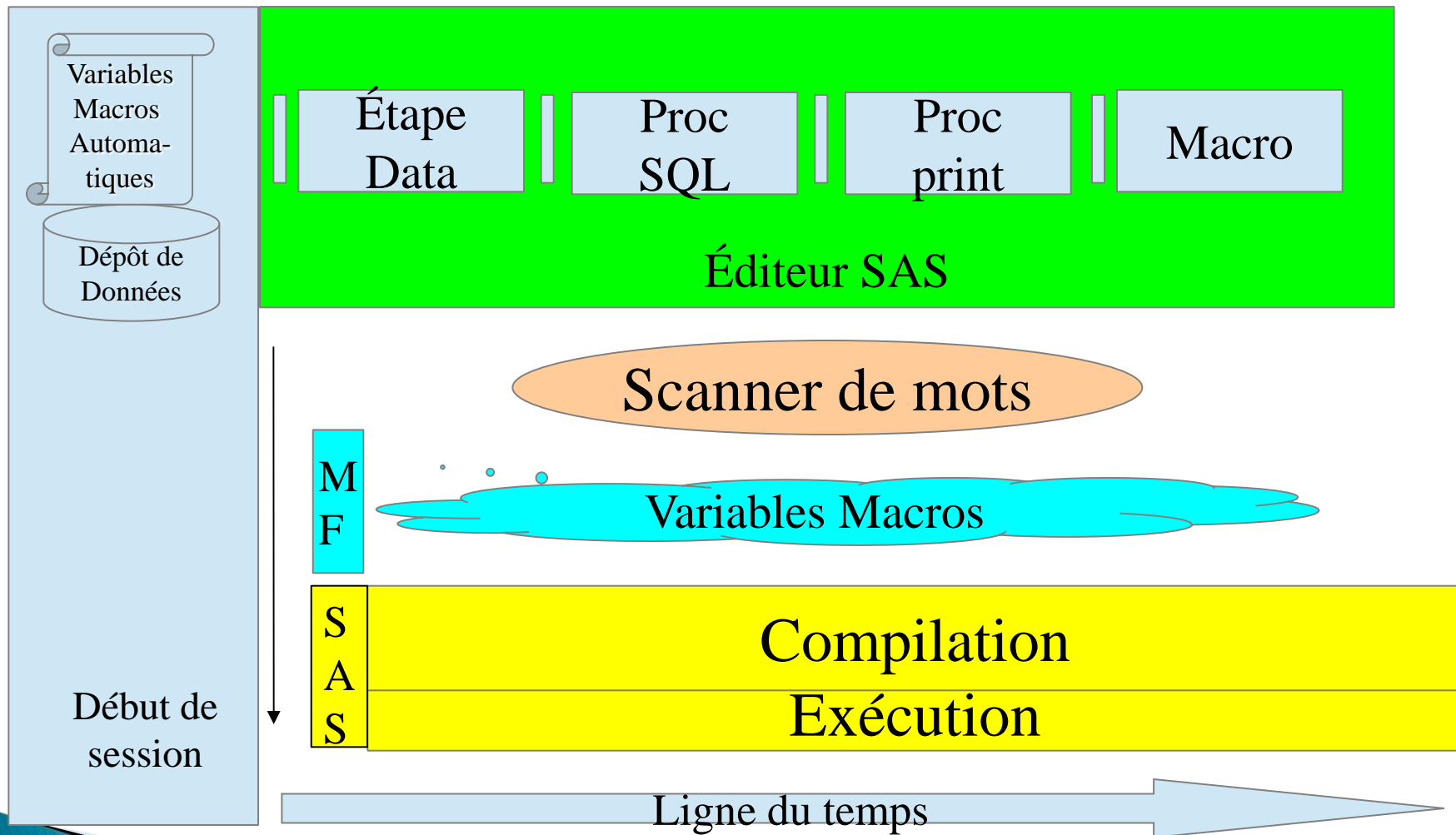
```
Proc PRINT DATA=educ.pers;  
  FOOTNOTE "Document produit par &sysuserid";  
RUN;
```

---

34	340	DELMON	84	VENTE	7	6,880.87	4,692.21
35	350	MATHIEU	84	ADMIN	5	7,967.33	.

Document produit par **Serv. Conseils Hardy**

## 2.3 Les variables Macros



## 2.3 Processus suivi par l'ordinateur

Lorsque l'utilisateur soumet le code, celui-ci est analysé par « scanner de mots » qui :

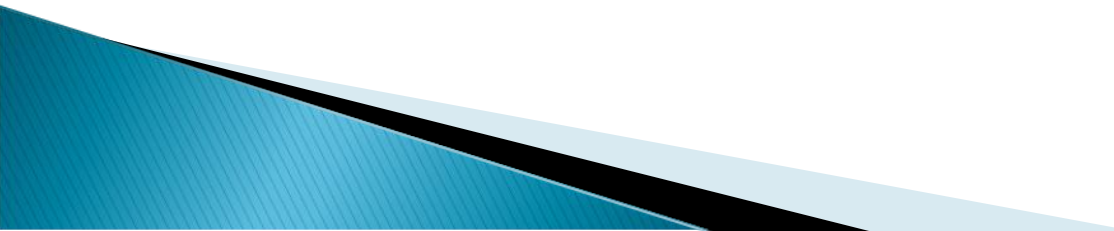
- Envoie les mots commençant par & suivi d'une lettre ou \_, à Macro Facility pour remplacement.
- Envoie les autres mots à SAS pour compilation.

Lorsque SAS a reçu les mots jusqu'à l'énoncé « run; » il compile le programme ou la procédure.

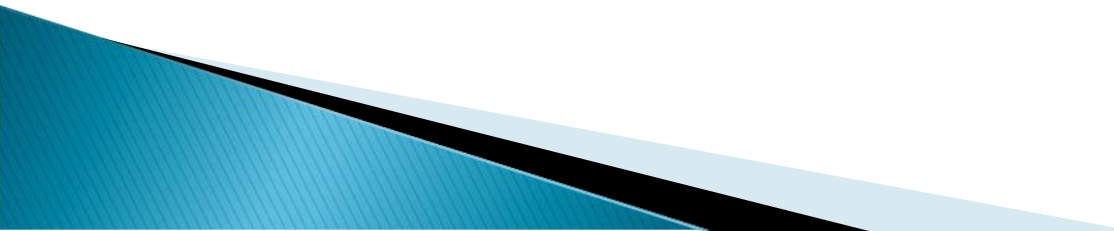
Lorsque la compilation est valide SAS l'exécute.



## 2.4 Caractéristiques générales

- Temporaires
  - Contiennent du texte.
  - Toutes les composantes SAS peuvent utiliser des variables Macros.
  - Elles sont disponibles à la fin de l'étape créatrice.
- 

## 2.4 Autres caractéristiques des variables Macro

- Le scanner de mots reconnaît une variable Macro par un & suivi d'un caractère autre que l'espace, selon les règles de nomenclature des variables SAS.
  - Cette référence à une variable Macro peut se trouver n'importe où dans un programme SAS.
  - Si la référence se trouve dans une chaîne de caractères, la chaîne doit être encadrée de guillemets (") et non d'apostrophes (').
  - Toujours résolue avant l'exécution de l'énoncé où elle se trouve.
- 

## 2.4 Autres caractéristiques des variables macro

- ▶ Utilisables dans les étapes DATA ou Proc et routines Macro
- ▶ Existent pour la durée de la session SAS
- ▶ Ne peuvent pas être sauvegardées dans des fichiers, pour usage ultérieur (version 9.4 Proc presenv Proc Strait)
- ▶ Toutes de type caractère, même si leur valeur ne contient que des chiffres
- ▶ Longueur effective = celle de sa valeur courante (pas de longueur fixe comme variables de table)
- ▶ Longueur maximale : 65534 caractères
- ▶ Variables Macro automatiques sont définies et affectées de leur valeur par le système SAS.

## 2.5 Les variables Macro automatiques

- Quelques variables Macro automatiques:

<b>SYSDATE9</b>	Date du début de la session ou du travail en lot (Ex. <i>22AUG2013</i> )
<b>SYSTIME</b>	Heure du début de la session ou du travail en lot (Ex. <i>10:53</i> )
<b>SYSLAST</b>	Nom de la dernière table créée ou modifiée (Ex. <i>EDUC.TEST</i> )
<b>SYSUSERID</b>	Identification de l'utilisateur (dans tous les environnements, en lot ou en interactif)

# 2.6 Affecter et utiliser une variable Macro

Affecter une variable macro :

- En Code ouvert ou Macro on utilise : %LET
- Étape Data : Call Symput
- Proc SQL : INTO

Utiliser la variable macro :

- En code ouvert, dans une définition de macro, dans une procédure ou dans une étape DATA; **&nom\_de\_la\_variable\_macro.**
- Aussi dans l'étape DATA; les énoncés SYMGET et RESOLVE.



## 2.7 Affecter une valeur à une var. Macro

- ▶ Utiliser l'énoncé `%LET`.

- ▶ Syntaxe :

`%LET varMacro = valeur;`

- ▶ Exemples :

```
%LET DEP=20;
```

```
%LET NOM_FICH=PERS;
```

- ▶ Pas de & lors de la définition d'une variable Macro.

- ▶ Dans le deuxième exemple, même si `NOM_FICH` est affecté d'une valeur caractère, celle-ci n'est pas placée entre guillemets ou apostrophes.

# 2.7 Définir puis utiliser des var. Macro

## ► Exemple :

```
%LET DEP=20;  
%LET NOM_FICH=PERS;  
LIBNAME EDUC "C:\EDUCSAS";  
Proc PRINT DATA=EDUC.&nom_fich;  
  WHERE DEPT=&dep;  
  TITLE1 "Liste des employés du département &dep";  
  TITLE2 "(fichier &nom_fich)";  
  FOOTNOTE "Document produit par &sysuserid";  
RUN;
```

## 2.7 Définir puis utiliser des var.... (suite)

Liste des employés du département 20  
(fichier PERS)

Obs	MATRICUL	NOM	DEPT	POSTE	ANNEES	SALAIRE	COMM
1	10	TANGUY	20	CADRE	7	12050.09	.
2	20	GATTAU	20	VENTE	8	8084.18	6213.22
8	80	HENRY	20	ADMIN	.	9210.41	.
19	190	JANISSON	20	ADMIN	8	9935.40	.

Document produit par Jean Hardy

## 2.8 Affectation de var. Macro par Data

- ▶ **CALL SYMPUT**, une fonction de l'étape DATA, est la seule solution :
- ▶ Syntaxe :
  - CALL SYMPUT(" *varMacro*", *var-caractère*)
  - CALL SYMPUT(" *varMacro*", put(*var-numérique*, *format*))
- ▶ Le deuxième argument doit être une chaîne ou une valeur numérique convertie en chaîne par la fonction **PUT** de l'étape DATA.
- ▶ Exemple – trouver la date de la plus ancienne transaction dans un fichier, qui doit demeurer trié par *NOM*, et l'utiliser dans un titre :

## 2.9 Exemple de CALL SYMPUT

```
Proc SORT DATA=EDUC.ENTREVUE (KEEP=DATE MATR_P NOM
    PRENOM)
        OUT=TRI;
    BY NOM PRENOM;
RUN;
DATA _NULL_;
    SET TRI END=FIN;
    RETAIN X 9999999999;
    X=MIN(X, DATE);
    IF FIN THEN CALL SYMPUT ("PREM_ENTR", PUT (X, YMMDD10.));
RUN;
Proc PRINT DATA=TRI NOOBS;
    TITLE "CANDIDATS INTERVIEWÉS DEPUIS LE &PREM_ENTR";
RUN;
```

## 2.10 Affectation de var. Macro par Proc SQL

- ▶ Syntaxe :

```
SELECT expression INTO : varMacro  
FROM table ...;
```

- ▶ Clause **SELECT** peut retourner plus d'une valeur – si oui, utiliser la syntaxe suivante:

```
SELECT expression INTO : varMacro  
SEPARATED BY "car"  
FROM table ...;
```

## 2.11 Exemple d'affectation par Proc SQL

```
Proc SQL;  
    SELECT MIN (DATE)  FORMAT=YYMMDD10 .  
           INTO : PREM_ENTR  
    FROM EDUC.ENTREVUE;
```

- ▶ La variable Macro contient une valeur, la même que dans l'exemple précédent.

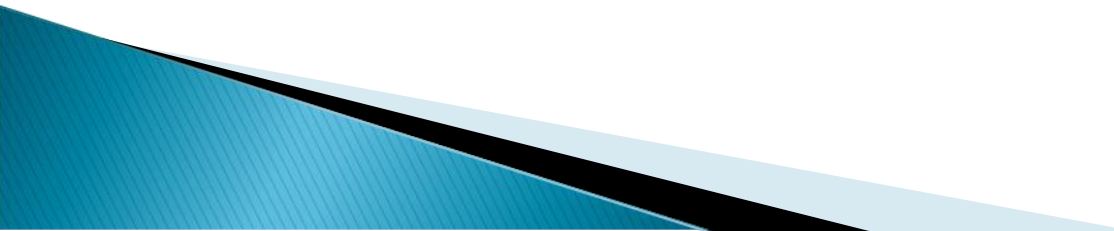
# 2.12 Afficher le contenu d'une var. Macro

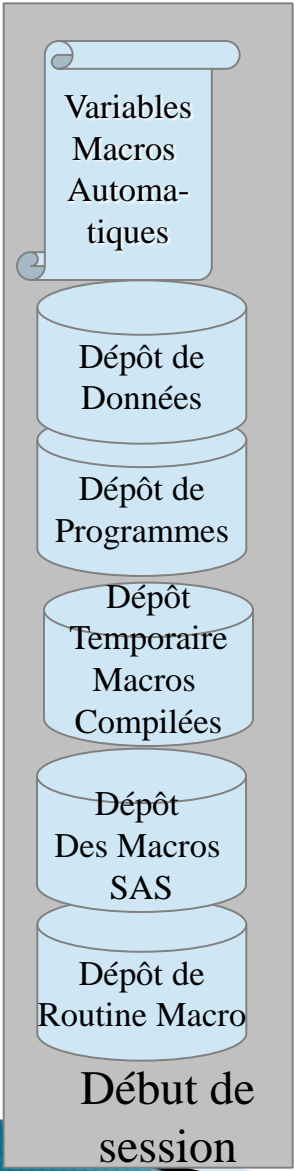
- ▶ Utiliser l'énoncé ou la commande %PUT.
- ▶ Syntaxe :  
%PUT *texte / &varMacro ou \_all\_, \_automatic\_,  
\_user\_, \_global\_ ou \_local\_;*
- ▶ Pas de guillemets autour du texte.
- ▶ Un & devant la variable Macro, comme lors de toute utilisation d'une variable Macro existante.
- ▶ Exemple :

```
      %PUT Variable DEP contient &DEP;  
40  
44  %PUT Variable DEP contient &dep;  
Variable DEP contient 10,15,20
```



# 3. Le langage Macro

- Les Macro servent à écrire du code SAS.
  - Langage de programmation propre aux Macros.
  - Moduler le code SAS.
- 



Options macros

Éditeur SAS

Scanner de mots

Variables macros

M  
F

Compilation  
Exécution

S  
A  
S

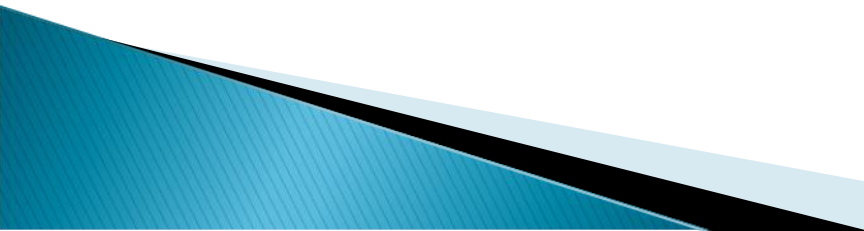
Compilation  
Exécution

Ligne du temps

# 3 Explication de la carte

La carte est composé de 14 éléments.

Il reste 6 éléments à découvrir :

- Les options macros.
  - Le compilateur MF
  - L'exécuteur MF
  - Le dépôt de routine macro.
  - Le dépôt de macros SAS.
  - Le dépôt temporaire de macros compilées.
- 

# 3.1 L'environnement du logiciel Macro Facility

À l'ouverture de la session il y a :

- Création des variables Macros automatiques.
- Allocation des Macros SAS.
- Allocation de la bibliothèque Work où sera placé le catalogue SASMACR des Macros compilées.
- Les options Macros sont mises en place

Optionnellement:

- Allocation de dépôts de données.
  - Allocation du répertoire des programmes.
  - Allocation du répertoire de routine Macros.
- 

## 3.1 L'environnement du logiciel Macro Facility (MF)(suite)

Les options Macro vont moduler le fonctionnement de MF.

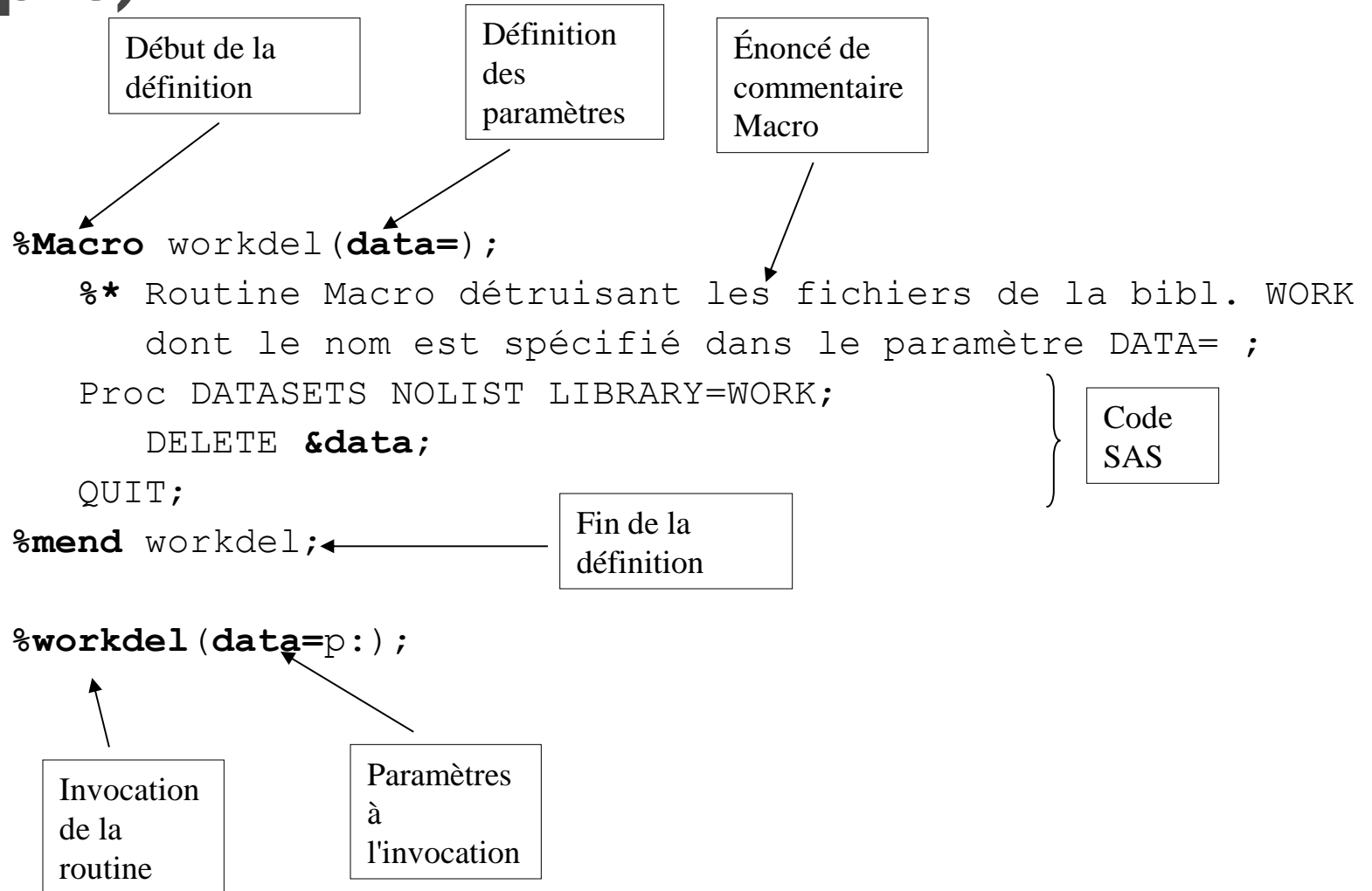
Donc à partir du moment où l'on soumet un traitement il y a intervention du scanner de mots pour déterminer si le traitement doit-être exécuter par MF ou SAS.

MF fait la gestion entre le compilateur et exécuter.

Une fois exécuté par MF, le code SAS généré est passé au compilateur SAS et de là en exécution.



# 3.2 Les routine Macro (exemple simple)



## 3.2 Les routines Macro (exemple simple suite)

```
%Macro workdel (data=) ;  
    %* Routine Macro détruisant les fichiers de la bibl. WORK  
    dont le nom est spécifié dans le paramètre DATA= ;  
    Proc DATASETS NOLIST LIBRARY=WORK;  
        DELETE &data;  
    QUIT;  
%mend workdel;  
  
%workdel (data=pe:);
```

► Le texte ne débutant pas par % est transmis à MF pour compilation, après l'exécution de la macro tout le code est transmis à SAS pour compilation et exécution.

## 3.2 Les routines Macro (exemple simple suite)

- ▶ L'option **MPRINT** affiche au journal le code SAS généré par une routine Macro :

```
203 option mprint;
204 %workdel (data=pe:);
MPRINT (WORKDEL) : Proc DATASETS NOLIST LIBRARY=WORK;
MPRINT (WORKDEL) : DELETE pe:;
MPRINT (WORKDEL) : RUN;

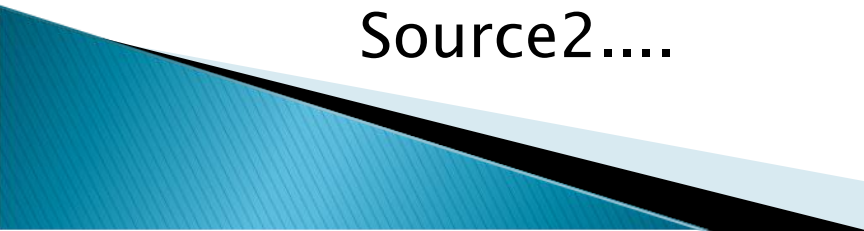
NOTE: Deleting WORK.PERS1 (memtype=DATA) .
NOTE: Deleting WORK.PERS2 (memtype=DATA) .
MPRINT (WORKDEL) : QUIT;
```

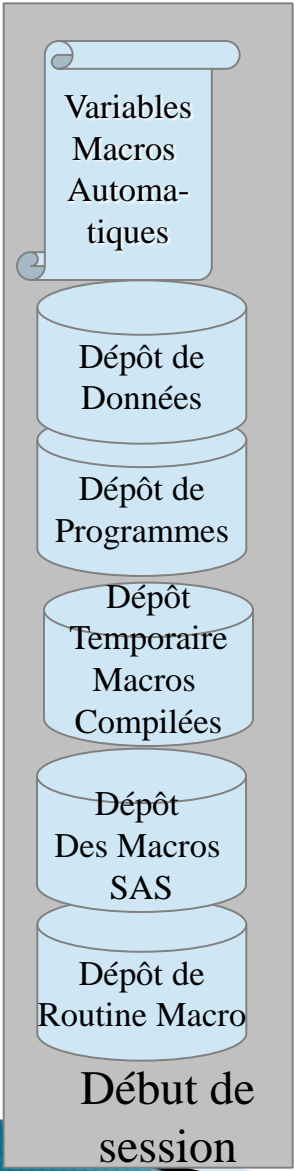


# 3.3 Les Options SAS Macro

Les options SAS Macro servent à gérer le comportement de MF.

2 groupes d'options Macro :

- Débogage
  - Comportement ou les autres.
  - Débogage
- Symbolgen, Mlogic, Mprint, Mlogicnest, Mprintnest.
- Comportement
    - Sasautos, Mfile, Macro, Merror, Source, Source2.....
- 



Options macros

Éditeur SAS

Scanner de mots

Variables macros

M  
F

Compilation  
Exécution

S  
A  
S

Compilation  
Exécution

Ligne du temps

## 3.4 Explication de la compilation de Macros

Lorsque « scanner de mots » rencontre %Macro il passe le contrôle des opérations au logiciel Macro Facility.

- Celui-ci prépare la compilation en stockant le texte qui suit jusqu'à l'énoncé %mend;.
- Il compile cette routine Macro.
- Qui retourne le résultat à SAS pour le mettre au journal.
- Sas poursuit le traitement.
- Les variables Macros sont irrésolues en attente de l'exécution de la Macro.

# 3.5 L'exécution de la routine Macro

Lorsque le logiciel MF rencontre l'appel de Macro  
%Workdel(data=PE:);. Il effectue la recherche de la référence  
« workdel » dans:

- Catalogue MACR de la bibliothèque WORK.
- Dépôt de Macro SAS.
- Dépôt de Macros permanentes.

Il arrête la recherche dès qu'il trouve.

La compile si elle n'est compilée.

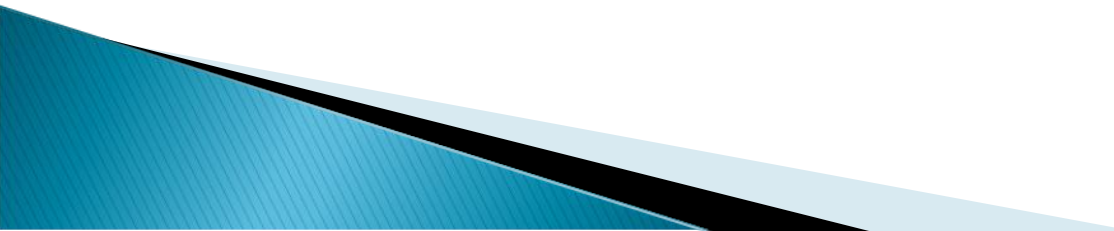
L'exécute.

Il envoie à SAS le résultat :

```
Proc DATASETS NOLIST LIBRARY=WORK;  
    DELETE PE ;  
QUIT;
```

# 3.6 Constituants des routines Macro

Une routine Macro peut être constituée d'un ou plusieurs des éléments suivants :

- ▶ Énoncés Macro (%LET, %DO, %IF, %PUT, etc.)
  - ▶ Fonctions Macro (%SUBSTR, %SCAN, %TRIM, etc.)
  - ▶ Références à une variable Macro
  - ▶ Code SAS
- 

# 4. Conclusion

Le langage Macro de SAS comprend deux composantes :

- Les variables Macros.
- Les routines Macros.

Permettent de générer du code SAS.

Sas 9.3 macro language reference.

J'espère avoir démystifié ce logiciel qu'il est bon d'avoir dans sa trousse à outils de programmeur SAS.

%FIN