

## LES MACRO-COMMANDES %DO , %TO et %END

Les instructions comprises entre une macro-commande %DO et une macro-commande %END peuvent contenir une ou plusieurs étapes DATA ou/et PROC. Mais attention, il ne faut pas confondre les macro-commandes %DO et %END avec les expressions DO et END contenues dans une étape DATA. **Si le bloc d'instructions compris entre %DO et %END peut contenir une ou plusieurs DATA, le bloc commençant par DO et se terminant par END ne peut contenir ni une étape DATA, ni une étape PROC.**

### Exemple

Avec la macro-expression *calcul* ci-dessous qui calcule le carré (nb2) puis le cube (nb3) d'un chiffre quelconque (nb ou i),

```
%MACRO calcul(i);  
DATA data&i;  
nb=&i;  
nb2=&i*&i;  
nb3=&i*&i*&i;  
PROC PRINT;  
title " carré et cube de &i";  
%MEND;
```

vous pouvez, par exemple, calculer le carré et le cube de 5 en insérant ce chiffre dans l'argument suivant la macro-commande *calcul* suivante.

```
%calcul(5) ;  
RUN ;
```

Ces calculs peuvent être réalisés pour un très grand nombre de chiffres en définissant des bornes de calcul inférieure (debut) et supérieure (fin) dans la boucle définie à l'aide des macro-commandes %DO, %TO et %END contenues dans la macro-expression *devel*.

```
%MACRO devel(debut,fin);  
%DO indice = &debut %TO &fin ;  
%calcul (&indice);  
%END;  
%MEND;
```

Les valeurs *debut* et *fin*, définies dans l'argument associé à la macro-commande %devel,

```
%devel(5,10);  
RUN ;
```

rendent équivalentes la macro-commande *devel* et les macro-commandes ci-dessous.

```
%calcul(5);  
%calcul(6);  
%calcul(7);  
%calcul(8);  
%calcul(9);  
%calcul(10);  
RUN;
```