3 - Noms et variables

1 Les noms dans Maple

Les *noms*, au sens où nous l'entendrons ici, servent à former des mots, des messages, ou à nommer des objets. On rappelle que Maple différencie les majuscules des minuscules.

Un nom débute en général par une lettre suivie éventuellement par des lettres ou des chiffres.

Le caractère de soulignement _ peut également figurer dans un nom, à n'importe quelle position.

Mais on évitera de le placer au début car Maple utilise de tels noms pour son usage interne.

2 Utilisation des caractères ‘

On peut en fait former un nom avec des caractères quelconques notamment des espaces, ou des caractères particuliers comme *, ?, +, etc.

Il suffit pour celà de débuter et de terminer le nom par le caractère ‘ (accent grave).

Il est même possible d'utiliser ce caractère ‘ dans le nom, à condition de le doubler en “.

Les accents graves ‘ n'apparaissent que lors de la création du nom. Ils ne sont ensuite plus affichés, et ils ne comptent pas dans la longueur du nom. Ils sont indispensables pour créer des noms contenant des caractères ni alphabétiques ni numériques (par exemple des caractères accentués !), ou débutant par un chiffre.

Dans le cas où les accents graves ne sont pas à priori indispensables, ils permettent de s'assurer que le nom sera bien interprété comme tel par Maple. Par exemple ‘expand‘ est un nom qui ne risque pas d'être interprété comme un appel à la fonction intégrée expand.

Pour obtenir un accent grave au clavier, on pourra utiliser la combinaison de touches AltGr+7 suivie d'un appui sur la barre d'espacement. Cet appui sur espace n'est vraiment indispensable que lorsque l'accent grave précède une lettre comme *a, e, i, o, u*, pour éviter que ne soit créée une lettre accentuée.

Dans tous les cas la taille maximum d'un nom est de 499 caractères.

3 Passage à la ligne dans un nom

Même si ce n'est pas très courant on peut placer un saut de ligne (par Entrée ou Shift+Entrée) dans un nom nécessairement débutant par ‘.

On n'omettra pas cependant de clore ce nom par un autre accent grave, sans quoi toute nouvelle ligne serait encore considérée comme une continuation du nom (un message de Maple permet de s'en rendre compte).

©EduKlub S.A.
Tous droits de l'auteur des œuvres réservés. Sauf autorisation, la reproduction ainsi que toute utilisation des œuvres autre que la consultation individuelle et privée sont interdites.

Extrait gratuit de document, le document original comporte 8 pages.

Si on arrive en fin de ligne à l'écran et si le nom n'est pas encore terminé, il y a saut à la ligne automatique, qui ne sera pas considéré comme faisant partie du nom.

On peut également forcer le passage à la ligne en utilisant le caractère \ (qui sera ignoré dans la formation du nom). Enfin pour qu'un caractère \ apparaisse effectivement dans un nom, il faut le doubler en \\.

4 Noms de variable

Il est très souvent utile de mémoriser un résultat en lui donnant un nom.

N'importe quelle chaine de caractères peut alors être utilisée pour ce nom, mais il est recommandé, pour des raisons de lisibilité, de se limiter à des noms débutant par un caractère alphabétique (lettre) suivi éventuellement par des lettres, des chiffres ou des caractères de soulignement.

On ne peut donner un nom à un résultat si ce nom est également celui d'une instruction intégrée de Maple.

5 Assignation et désassignation

L'opération qui consiste à associer un résultat à un nom est l'*assignation* (ou encore *affectation*), qui utilise le double caractère :=.

La syntaxe est donc name := expr.

Ici *expr* peut être une expression quelconque, qui est d'abord *évaluée* avant de faire l'objet de cette assignation. Dans toute la suite de la session (sauf évidemment si on affecte un nouveau résultat à *name*), le nom *name* sera toujours remplacé par cette valeur de *expr*.

Quand une instruction name := expr ; (bien noter le point-virgule) est exécutée, Maple affiche quelque chose comme *name := valeur_de_expr*.

Que l'instruction se termine par ; (point-virgule) ou par : (deux-points), le contenu de la fonction % est *valeur_de_expr*.

Par exemple :

```
> x :=2^5 ; %+1 ;
```

$$x := 32$$

$$33$$

Tant qu'un nom comme *name* n'a pas subi une affectation particulière, il continue à ne désigner que lui-même. On peut alors considérer *name* comme une variable formelle (symbolique), c'est-à-dire une variable au sens mathématique habituel.

©EduKlub S.A.
Tous droits de l'auteur des œuvres réservés. Sauf autorisation, la reproduction ainsi que toute utilisation des œuvres autre que la consultation individuelle et privée sont interdites.

Extrait gratuit de document, le document original comporte 8 pages.

Une des grandes forces de Maple est de pouvoir traiter, manipuler et simplifier des expressions contenant de tels noms formels.

Pour désassigner un nom comme *name*, on exécutera l'instruction name := 'name' (on notera l'utilisation d'*apostrophes*), ce qui aura pour conséquence que le nom *name* ne pointera (jusqu'à nouvel ordre) que sur lui-même : il aura ainsi retrouvé son statut de variable formelle.

On rappelle que l'instruction restart permet de redémarrer complètement la session de travail et en particulier de désassigner tous les noms de variables qu'aurait créés l'utilisateur.

6 Noms obtenus par concaténation

On peut former des noms à partir d'autres noms, par *concaténation*.
L'opérateur utilisé est la double barre vertical (——).

Par exemple, l'expression x——1 sera évaluée en x1, et A——i——j sera évaluée en le nom Aij. Dans cette dernière construction on suppose que les variables i et j ne sont pas assignées, car sinon les noms i et j seraient d'abord évalués avant la concaténation.

Dans la création d'un nom par concaténation, le nom le plus à gauche n'est jamais évalué.

Exemple :

```
> restart : A :=1 : i :=2 : j :=3 : A||i||j :=10 ; A||k :=100 ;
```

$$A23 := 10$$

$$Ak := 100$$

7 Lettres grecques

Les lettres grecques sont saisies sous la forme *alpha*, *beta*, *gamma*, *lambda*, *mu*, *epsilon*, *pi*, etc., mais Maple les affiche avec le caractère correspondant de l'alphabet grec.

Par exemple :

```
> alpha,beta,delta,gamma,lambda,epsilon,mu,nu,pi,phi,rho,sigma,theta ;
```

$$\alpha,\ \beta,\ \delta,\ \gamma,\ \lambda,\ \varepsilon,\ \mu,\ \nu,\ \pi,\ \phi,\ \rho,\ \sigma,\ \theta$$

8 Evaluation récursive d'une expression

Quand une expression expr est évaluée (par exemple lors d'une assignation name := expr), les noms assignés qu'elle contient sont évalués. Il se peut que le contenu d'un de ces noms soit lui-même une expression contenant des noms assignés. Ceux-ci sont à leur tout évalués.

On voit donc se mettre en place une évaluation récursive, jusqu'à ce que les noms qui apparaissent ne soient plus assignés (c'est-à-dire ne désignent qu'eux-mêmes).

Par exemple :

©EduKlub S.A.
Tous droits de l'auteur des œuvres réservés. Sauf autorisation, la reproduction ainsi que toute utilisation des œuvres autre que la consultation individuelle et privée sont interdites.

Extrait gratuit de document, le document original comporte 8 pages.