2 - Syntaxe de base

1 Input et Ouput

On entre des *lignes de saisie (Input)* au clavier, et Maple y répond (*Output*).

Pour que Maple comprenne que cette *ligne de saisie* est complète, qu'elle doit être prise en compte et qu'une réponse doit lui être apportée, deux conditions doivent être réunies :

- Elle doit se terminer par l'un des deux caractères **:** (deux-points) ou **;** (point-virgule).
- L'utilisateur doit appuyer sur la touche *Entrée*.
  Pour évaluer cette ligne de saisie, on peut aussi la *double-cliquer* (deux clics rapprochés avec le bouton gauche de la souris) en un point quelconque.

2 Lignes de saisie et lignes physiques

Une ligne de saisie débute par le caractère > appelé le *prompt*.

Cette ligne *logique* peut cependant occuper plusieurs *lignes physiques* successives à l'écran.

La touche *Entrée* permet de continuer sur la ligne physique suivante.

La ligne de saisie ne sera pas analysée et évaluée par Maple tant qu'elle ne se terminera pas par un des deux caractères terminaux **:** ou **;**. Tout au plus Maple vous préviendra par un message que vous êtes peut-être en train d'oublier ce caractère.

On peut encore continuer la ligne de saisie sur la ligne physique suivante même si le dernier caractère significatif est **:** ou **;**
Dans ce cas on utilisera *Shift+Entrée*, pour prévenir une évaluation prématurée.

Arrivé au bord droit de la feuille, la saisie continue automatiquement à la ligne suivante, et dans ce cas le saut de ligne est non significatif.

On peut enfin forcer le passage à la ligne suivante à l'intérieur même d'un mot, par exemple un entier très long. On utilisera dans ce cas le caractère \ (*anti-slash*) juste avant le passage à la ligne (par *Entrée* ou mieux par *Shift+Entrée*). Ce saut de ligne est alors non significatif.

Plus généralement, \ peut servir à améliorer la lisibilité d'un mot très long, comme par exemple découper un entier en blocs de trois chiffres. Un tel caractère \ ne sera pas interprété lors de l' évaluation de la ligne de saisie. Enfin, si on tape \\ , le premier \ est ignoré et le deuxième est un caractère comme un autre.

3 Commentaires

Dans une ligne de saisie, on peut placer un commentaire.

On utilise pour cela le caractère # : tout ce qui suivra #, dans la ligne physique où il se trouve, sera ignoré lors de l'évaluation de la ligne de saisie.

Il y a une exception : à l'intérieur d'une chaine de caractères délimitée par des ", le caractère # sera considéré comme faisant partie de la chaine.

4 Majuscules et minuscules

Dans l'analyse d'une ligne de saisie, Maple distingue les majuscules des minuscules.

L'exemple classique est la constante Pi ($= 3.1415\ldots$) qu'il ne faudra pas écrire pi ou PI.

De même diff et Diff sont deux instructions différentes de Maple.

5 Les séparateurs d'instructions : et ;

Une même *ligne de saisie* est formée d'une ou de plusieurs *instructions* successives, séparées par l'un des deux caractères **:** ou **;**

Lors de l'évaluation de la ligne de commande (par *Entrée*, la dernière instruction se terminant par **:** ou **;**), ces instructions sont *interprétées* et *évaluées* par Maple dans l'ordre où elles se trouvent. Seules celles qui se terminent par le caractère **;** (point-virgule) voient leur résultat affiché à l'écran.

6 Géographie et historique des calculs

En mode interactif, alors que Maple attend vos ordres au clavier, la ligne logique courante est celle où se trouve le curseur.

En principe, une session de travail est constituée d'une suite de lignes de saisie auxquelles Maple a apporté des réponses, selon un schéma *saisie1-réponse1-saisie2-réponse2*-etc. qui correspond à l'historique des calculs.

Il est possible de déplacer le curseur d'un point à un autre de la feuille de travail (au clavier avec les touches de déplacement, ou avec la souris).

On peut ainsi revenir sur une ligne de commande précédente (et l'évaluer de nouveau par *Entrée*, après l'avoir en général modifiée), ce qui rompt le parallélisme entre l'*historique* des calculs et ce qui apparait affiché à l'écran (la *géographie*).

7 La fonction %

Dans une nouvelle instruction, il est souvent utile de se référer au dernier résultat calculé par Maple. Plutôt que de recopier ce résultat au clavier, on peut se servir de la fonction %.
Le résultat renvoyé par % est celui de la dernière instruction *exécutée* par Maple, que ce résultat ait été *affiché* ou non, c'est-à-dire que cette dernière instruction se termine par **;** ou **:**.

On peut même utiliser %% ou %%% qui contiennent respectivement l'avant-dernier résultat calculé ou l'avant-avant-dernier (l'antépénultième).

Attention : quand on parle du dernier calcul (ou de l'avant-dernier, etc.) on se réfère à l'historique qui peut différer de la succession physique des lignes à l'écran.

Si l'instruction qui utilise % est la première d'une ligne de saisie, le contenu de % désigne le résultat calculé de la dernière instruction de la ligne de saisie précédente au sens de l'historique, qui peut être différente de la ligne de saisie située au-dessus à l'écran.

Si l'instruction qui utilise % n'est pas la première de sa ligne de saisie, % désigne le dernier résultat calculé dans cette même ligne de saisie. Dans ce cas le risque d'erreur sur l'historique est moins important.
Pour compliquer un peu la situation, certaines instructions, assez rares heureusement, n'ont pas d'influence sur l'historique : le résultat qu'elles produisent ne remplace en effet pas celui qui est contenu dans %.

Il s'agit par exemple des commandes d'affichage print et lprint (*linear print*), cette dernière affichant un résultat en mode linéaire, c'est-à-dire sans le *pretty-print* habituel.

L'intérêt de %, %% ou %%% est d'éviter de stocker dans des variables des résultats très récents et qui doivent être immédiatement réutilisés mais qui deviennent inutiles ensuite.

Attention : avant la version 5.1 du logiciel, le caractère utilisé pour désigner le résultat précédent était " et non %. Le caractère " désigne maintenant les délimiteurs de chaines de caractères (avant on utilisait l'accent grave `).

8 Fonctions et arguments

Dans l'expression expand((x+1)^2,x), on dit que expand est une *fonction* et que ses *arguments* sont (x+1)^2 et x. Le résultat *renvoyé* (on dit aussi *retourné*) par cette fonction, avec ces arguments, est x^2+2x+1.

La liste (éventuellement vide) des arguments d'une fonction devra toujours être placée entre parenthèses.

Une expression comme $x(y+1)$ peut prêter à confusion : est-ce la fonction nommée x appelée avec l'argument $y+1$, ou le produit de la variable x par l'expression $y+1$?

Pour lever ce genre d'ambiguité, le signe de multiplication * est toujours obligatoire (même si le risque n'existe pas : on écrira par exemple 2*x et non pas 2x).

Pour des raisons d'esthétique, ces caractères * ne seront pas affichés dans les résultats.

9 Les signes := et =

L'une des principales instructions de Maple est l'affectation, qui consiste à stocker un résultat dans une variable (c'est-à-dire à donner un nom à ce résultat) et qui utilise le symbole := (comme en *Pascal*).

La forme de cette instruction est donc : *nom* := *expression*

Le signe = sert à former des équations commme solve(exp(x)=3*x,x)) ou des égalités qui pourront être testées dans un contexte *booléen* (par exemple dans un test : if a = b then...).

10 Faire le ménage

Au cours d'une session de travail, on est amené à stocker des résultats dans des *variables*.

Certains noms sont alors affectés et d'autres ne le sont pas, et il n'est pas toujours facile de s'en souvenir, ce qui peut occasionner des problèmes.

On peut alors repartir à zéro de deux manières, en excluant l'idée saugrenue de carrément quitter le logiciel pour y revenir.

- On peut fermer la feuille de calcul en cours et en ouvrir une autre.
- On peut utiliser l'instruction restart : c'est préférable.

11 Un petit test

Imaginez les réponses (et précisez quand celles-ci sont des messages d'erreur) à chacune des lignes de saisie suivantes. Comparer ensuite avec les réponses effectivement données par Maple, et justifiez les réactions du logiciel.

Dans certaines des lignes de saisie ci-dessous, on trouvera l'instruction restart, qui permet de s'assurer que toutes les variables sont préalablement *désassignées*, c'est-à-dire n'ont plus de *contenu*.

Question 1 [Réponse]

```
> 2 : %+1 ; %+2 : %%+% : %%%+%%+% ;
```

Question 2 [Réponse]

```
> print(2^3) :
```

Question 3 [Réponse]

```
> 2 : print(2^3) : %+10 ;
```

Question 4 [Réponse]

```
> x :=2 : y :=3 : x(y+1) ;
```

Question 5 [Réponse]

```
> x :=5 : print(2x) :
```

Question 6 [Réponse]

```
> restart : x :=2 : y :=5 : X*(Y+1) ;
```

Question 7 [Réponse]

```
> x :=7 : y :=3 : x+y+xy ;
```

Question 8 [Réponse]

```
> restart : x=2 : y :=x : x+3*y ;
```

Question 9 [Réponse]

```
> x :=5 # Ici on appuye sur Shift+Enter
  2^x ;
```

Question 10 [Réponse]

```
> x :=10 : %*% ;# y :=x+3 ; x*y ;
```

Question 11 [Réponse]

```
> restart : x=7\ # Ici on appuye sur Shift+Enter
  4+y ;
```

12 Réponses au test

Réponse à la question 1 [Retour au test]

La réponse de Maple est

$$3$$

$$16$$

L'entier 2 est évalué mais n'est pas affiché car il est suivi de : et non de ;
Le résultat 2 va dans %, l'expression $\% + 1$ s'évalue en 3, qui est affiché.
L'expression suivante, $\% + 2$, s'évalue en $3 + 2 = 5$. Le résultat n'est pas affiché.
A ce moment $\% = 5$ et $\%\% = 3$. L'expression $\%\% + \%$ s'evalue donc en $3 + 5 = 8$.
Après ce résultat non affiché, $\% = 8$, $\%\% = 5$, et $\%\%\% = 3$.
La dernière expression $\%\%\% + \%\% + \%$ s'évalue donc en $3 + 5 + 8 = 16$.

Réponse à la question 2 [Retour au test]

La réponse de Maple est

$$8$$

S'il y a affichage, malgré le caractère :, c'est parce qu'il s'agit de l'instruction `print`.

Réponse à la question 3 [Retour au test]

La réponse de Maple est

$$8$$

$$12$$

L'entier 2 est évalué, mais non affiché. Il devient le contenu de %.
Ensuite l'instruction print affiche la valeur de 2^3, c'est-à-dire 8.
Mais, parce que `print` n'agit pas sur l'historique, ce résultat ne vas pas dans %.
Il est donc normal que la dernière instruction s'évalue en $2 + 10 = 12$.

Réponse à la question 4 [Retour au test]

La réponse de Maple est

$$2$$

Les deux premières instructions placent 2 dans x, puis 3 dans y.
Ces valeurs ne sont pas affichées à cause des caractères : terminaux.
Dans la troisième instruction, Maple pense que $x(y + 1)$ désigne la valeur en $y + 1$, donc en 4, de la fonction nommée x.
Mais cette fonction x est en fait la fonction constante 2, d'où le résultat.
Pour afficher la valeur du produit $x(y + 1)$, il faut valider $x * (y + 1)$.

Réponse à la question 5 [Retour au test]

La réponse de Maple est

```
missing operator or ‘;‘
```

La première instruction place 5 dans la variable x. Le résultat n'est pas affiché.
La deuxième contient une erreur de syntaxe : il aurait fallu écrire `print`$(2 * x)$.

Réponse à la question 6 [Retour au test]

La réponse de Maple est

$$X\,(Y + 1)$$

Maple place 2 dans x, puis 5 dans y, sans rien afficher.
Comme `restart` a tout réinitialisé, X et Y ne contiennent rien.
Il est donc normal que l'évaluation de $X * (Y + 1)$ renvoie cette expression inchangée.
On retiendra que Maple différencie les majuscules et les minuscules.

Réponse à la question 7 [Retour au test]

La réponse de Maple est

$$10 + xy$$

Maple place 7 dans x, puis 3 dans y, sans rien afficher.
Pour lui, xy désigne un nom de variable, et non pas le produit de x par y.
Or il n'y a rien dans cette variable.
$x + y + xy$ s'évalue donc en $7 + 3 + xy = 10 + xy$.

Réponse à la question 8 [Retour au test]

La réponse de Maple est

$$4\,x$$

L'instruction `restart` vide le contenu de toutes les variables.
L'instruction $x = 2$ désigne une simple égalité et non pas l'affectation $x := 2$.
Il n'y a donc toujours rien dans x, et $y := x$ place le nom x dans la variable y.
L'expression $x + 3 * y$ s'évalue donc en $x + 3x$, donc en $4x$.

Réponse à la question 9 [Retour au test]

La réponse de Maple est

```
unexpected number
```

En effet le saut de ligne compte comme un séparateur, comme une espace.
Or Maple n'accepte pas que deux chiffres soient séparés pas une espace.

Réponse à la question 10 [Retour au test]

La réponse de Maple est

$$100$$

A l'issue de la première instruction, x et % contiennent 10.
La deuxième instruction évalue et affiche $\% * \% = 10 * 10 = 100$.
A cause de #, tout ce qui suit est ignoré car considéré comme un commentaire.

Réponse à la question 11 [Retour au test]

La réponse de Maple est

$$x = 74 + y$$

Le caractère " lie les deux lignes.
Tout se passe donc comme si on avait entré l'unique ligne $x = 74 + y$.