15 - Les Types de Maple

1. Types de base

Pour savoir quel est le type d'un objet *obj*, on évaluera l'instruction `whattype`(*obj*).

Le résultat est un nom.

On pourra vérifier qu'un objet *obj* est d'un type particulier en évaluant `type`(*obj,typname*), où *typname* est l'un des noms prédéfinis par Maple pour représenter tel ou tel type.

Le résultat est `true` ou `false`, suivant que l'objet *obj* est ou n'est pas de type *typname*.

Les types de base de Maple sont :

- `symbol`, `string` : noms symboliques, chaines de caractères.
- `integer`, `fraction`, `float` : entiers, nombres rationnels, nombres réels.
- `‘+‘`, `‘*‘`,`‘^‘` : sommes, produits, puissances.
- `‘=‘`, `‘<‘`, `‘<=‘`,`‘<>‘` : égalités et inégalités.
- `‘and‘`, `‘or‘`, `‘not‘` : expressions booléennes.
- `function` : appels à une fonction, ou composition de fonctions.
- `exprseq`, `set`, `list` : séquences, ensembles, listes.
  Remarque : le type `exprseq` ne peut pas être testé par la fonction `type`.
- `indexed`, `table`, `array`, `‘..‘` : noms indicés, tables, tableaux, intervalles.
- `procedure` : programmes.
- `series` : objets obtenus par l'instruction `series` (développements limités.)
- `uneval` : objets non évalué car encadrés d'apostrophes ('x', '2+3').
- `‘.‘` : concaténations non évaluées, par exemple $a.(2/3)$.

2. Réunions ou synonymes de types de base

Certains types sont des regroupements de types de base de Maple.

Ils peuvent encore être utilisés comme deuxième argument dans l'instruction `type` (la syntaxe est `type`(*obj,typname*).

On rappelle cependant que cette instruction ne renvoie que des noms de type de base.

©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 4 pages.