Programmation Fonctionnelle et Symbolique : feuille 1

Premiers pas - Expressions et fonctions

Exercice 1.1 *Symboles auto-évaluants*
Évaluer des entiers, des flottants, des rationnels, des complexes, des vecteurs, des caractères, des chaînes de caractères, les booléens.

Exercice 1.2 *Premières évaluations d'expressions*
Évaluer :

1. $3 + 4$.
2. $3.2 * 4.5$.
3. $|-5.3|/cos(4.3)$.
4. $3/4 - 4/5$.
5. $(1 + i3) * (2 + i4)$.

Exercice 1.3 *Arguments facultatifs et/ou nombre arbitraire d'arguments*
Évaluer, et cela après en avoir vérifié la définition par :

- la documentation "on-line" associée aux symboles
  exemple : `(documentation '+ 'function)` ou `C-c C-f` en donnant `+` comme argument)
- par la documentation `Hyperspec` :

1. $1 + 2 + 3 + 4$.
2. $maximum(1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 7, 6, 5, 4, 3, 2, 1)$.
3. $log_3(4)$.

Exercice 1.4 *Typage dynamique*
Pour le vérifier, utiliser les prédicats prédéfinis de la forme <*type*>`p` (exemple : `stringp` pour une chaîne).

Exercice 1.5 *Définition de nouvelles fonctions*
Implémenter :

1. $x + 3$.
2. x^2.
3. La valeur absolue.
4. La moyenne de deux nombres.
5. $\sum_{i=n}^{p} i$.
6. Factorielle (version naïve récursive).
7. Fibonacci.

Provoquer des erreurs en transmettant des arguments non valides aux fonctions.

Exercice 1.6 *Récursivité terminale*
Écrire une version récursive terminale de la fonction factorielle. Comparer ses limites avec celles de la version naïve.

Exercice 1.7 *Documentation des fonctions*
Ajouter de la documentation aux fonctions.