Traducteur Caml - Python

Opération	**Caml**	**Python**
Égalité, différence en valeur	`=`, `<>`	`==`, `!=`
Égalité, différence en mémoire (rarement utile)	`==`, `!=`	`is`, `is not`
Reste, quotient de `a` par `b` (algo. d'Euclide en $O(\log(a))$)	`a mod b`, `a / b`	`a % b`, `a // b`
Division classique flottante	`a /. b`	`a / b`
Puissance (par exponentiation rapide)	`a**b` (si `a` et `b` sont des flottants)	`a**b`
Opérateurs logiques: et, ou, non	`&&`, `||`, `not`	`and`, `or`, `not`
Vrai, faux	`true`, `false`	`True`, `False`
minimum, maximum de `x` et `y`	`min x y`, `max x y`	`min(x, y)`, `max(x, y)`
Définir une fonction `f(x, y)`	`let f x y = ... in`	`def f(x, y):`
Créer un tableau contenant 1, 2 et 3	`[|1; 2; 3|]`	`[1, 2, 3]` (tableau dynamique)
Créer un tableau de taille `n` rempli de `e`	`Array.make n e`	`[e] * n` (tableau dynamique)
Créer une matrice de taille `n`×`p` remplie de `e`	`Array.make_matrix n p e`	`[[e] * p for i in range(n)]`
Accéder à l'élément d'indice `i` d'un tableau `t`	`t.(i)`	`t[i]`
Modifier l'élément d'indice `i` d'un tableau `t`	`t.(i) <- ...`	`t[i] = ...`
Définir une liste contenant 1, 2 et 3	`[1; 2; 3]`	n'existe pas
Taille d'un tableau `t`	`Array.length t`	`len(t)`
Répéter ... pour `i` variant de 0 à `n-1`	`for i = 0 to n-1 do ... done`	`for i in range(n): ...`
Répéter ... tant que `condition` est vraie	`while condition do ... done`	`while condition: ...`

Remarque: les « listes » Python sont en fait implémentés comme des tableaux (car ils permettent d'accéder à un élément `L[i]` en une seule opération) redimensionnables (on peut leur ajouter un élément, contrairement aux tableaux OCaml - on verra plus tard comment implémenter un tableau redimmensionnable en OCaml).