PROGRAMMATION FONCTIONNELLE : SYNTHÈSE DE COURS

1) Les principes clés de la programmation fonctionnelle :

- Une fonction est une expression comme une autre, qui peut donc être passée en paramètre à une autre fonction, ou être renvoyée comme résultat.
- Une fonctions doit être « pure » : Le retour d'une fonction ne doit dépendre que de ses arguments, sans effets de bord. Une fonction pure doit donc toujours retourner le même résultat si ses arguments n'ont pas changé.
- Transparence Référentielle : Comme il n'y a pas d'effets de bord, on peut interchanger une expression avec sa valeur sans modifier le comportement du programme.
- Immutabilité : On s'interdit de modifier une entité après son instanciation. Il n'y a pas de réaffectation de variable (les variables sont en fait des constantes).

2) Conséquences de ces principes :

- Pour ne pas avoir à réaffecter des variables, on est amené à remplacer les boucles par la récursivité ou par des fonctions d'ordre supérieur comme map, filter ou reduce ou encore par des compréhensions de listes.
- Quand on aimerait malgré tout modifier une variable, on en fait plutôt une copie.
- D'habitude, un programme est constitué de briques logicielles que l'on exécute l'une après l'autre (en séquence), alors qu'en programmation fonctionnelle les briques logicielles s'exécutent en général l'une à l'intérieur de l'autre (composition de fonction).
- On remplace souvent l'instruction if par l'expression « valeur_si_vrai if condition else valeur_si_faux ».

3) Avantages :

- Permet d'éviter beaucoup de bugs en limitant l'implicite grâce à l'absence d'effets de bord.
- Facilite la parallélisation des calculs et permet dans certains cas un gain de rapidité.
- Facilite les tests et pousse à écrire des briques de programmes toujours plus simples et à les empiler pour former le programme final plus robuste.

4) Inconvénients :

- Moins intuitif (avis subjectif assumé ;-)
- Le fait de refuser les variables mutables peut obliger à dupliquer des données ce qui peut ralentir le programme et consommer plus de mémoire.

5) Et dans la vraie vie ?

Une fonction pure n'interagit pas avec l'extérieur, pourtant la plupart des programmes ont besoin de communiquer avec l'extérieur. Comment faire ?
L'idée n'est pas d'interdire les fonctions impures, mais de les séparer du reste du programme. En programmation fonctionnelle, on va séparer le "cœur pur" du programme, qui sera sans effet de bord et qui contiendra la logique du programme, d'une fine couche extérieure composée de fonctions qui se chargeront spécifiquement des effets de bord. Ces derniers seront alors davantage isolés et maîtrisés.

Il n'est pas simple pour quelqu'un qui a appris à développer de manière traditionnelle de changer sa manière de raisonner pour adopter une écriture plus fonctionnelle. Cela dit, les effets de bord et la mutabilité faisant partie des problèmes principaux auxquels font face les développeurs, les éliminer rendra les programmes plus concis, faciles à tester et à maintenir.