

INFO-H-100 - Informatique

Séance d'exercices 12
Introduction à Python
La récursivité

Université Libre de Bruxelles
Faculté des Sciences Appliquées

2011-2012

Second semestre : informations pratiques

- Tous les TPs auront lieu sur machine, à l'exception du premier (celui-ci) et le dernier (TP 17). Un horaire sera publié sur notre site web ainsi que sur les valves à l'entrée de la salle Socrate.
- Le second projet sera publié début mars et défendu devant les assistants en mai. Il se fera obligatoirement par groupe de deux étudiants.

Récurtivité

La récursivité se base sur le principe de **réurrence**.

Principe de réurrence

Soit P_n , une proposition relative à l'entier n et n_0 un entier.

- si la proposition P_{n_0} est vraie (initialisation)
- et si pour tout $k > n_0$, la proposition P_k vraie implique que la proposition P_{k+1} soit vraie (hérédité),

alors la proposition P_n est vraie pour tout entier $n \geq n_0$.

Fonction récursive

Une **fonction récursive** est une fonction qui s'appelle elle-même.

Exemple : factorielle

- initialisation : $0! = 1$
- hérédité : $n! = n \times (n - 1)!$

En Python :

```
def factorial(n):  
    if n < 1:  
        return 1  
    else:  
        return n * factorial(n-1)
```

Exécution :

<http://people.csail.mit.edu/pgbovine/python/tutor.html>

Bonnes pratiques

Lors de l'écriture d'une fonction récursive, il faut :

- accepter qu'un appel récursif retourne la bonne valeur
- s'assurer que le cas de base (initialisation) est bien géré
- s'assurer que chaque appel récursif réduise la taille du problème jusqu'à atteindre le cas de base

Si une récursion n'atteint jamais le cas de base, le programme ne se termine (théoriquement) jamais : c'est une récursion infinie.

```
def recurse() :  
    recurse()
```

Exercices