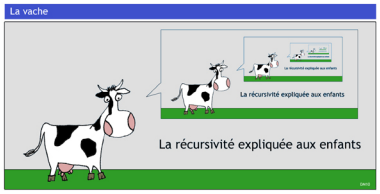
La récursivité

La récursivité est un phénomène qui se répète, qui s'applique à lui-même.



En informatique, le paradigme itératif avec les **boucles**, permet de répéter autant de fois que nécessaire une même instruction ou même bloc d'instructions.

Exercice

Écrire une fonction **somme** qui calcule la somme des n premiers nombres entiers positifs.

1. avec une boucle **for**.
2. avec une boucle **while**.

Fonction récursive

Une fonction **récursive** est une fonction qui s'appelle elle-même. Bien évidemment, elle ne peut s'appeler elle-même indéfiniment ! Elle contient donc une **condition d'arrêt**.

Remarque

1. Une fonction récursive est appelée une première fois avec ses arguments. Cette fonction, en s'exécutant, va donc s'appeler elle même avec d'autres arguments, qui va donc s'appeler elle même encore une troisième fois avec d'autres arguments, et ainsi de suite …
2. Une condition d'arrêt sur la valeur des arguments est vérifiée à chaque appel. Lorsque la condition est vraie, les appels s'arrêtent. Une pile d'appels s'est constituée mettant en attente toutes les fonctions appelantes.
3. La pile des fonctions en attente reprend son exécution jusqu'à l'appel initial qui renvoie le résultat attendu.

Exemple

Si on reprend la somme des premiers entiers, avec $n = 4$ soit $S = 1 + 2 + 3 + 4$.

- Calculer la somme des 4 premiers entiers revient à calculer la somme des 3 premiers à laquelle on ajoute 4 : **somme(3)+4**;
- Calculer la somme des 3 premiers entiers revient à calculer la somme des 2 premiers à laquelle on ajoute 3 : **somme(2)+3**;
- Calculer la somme des 2 premiers entiers revient à calculer la somme du premier entier à laquelle on ajoute 2 : **somme(2)+1**;
- Calculer la somme du premier entier revient à renvoyer sa valeur soit **somme(1)=1**.

Donc nous avons les appels successifs:

- somme(4) = 4 + somme(3) *se met en attente du résultat de l'appel de somme(3)*

- somme(3) = 3 + somme(2) *se met en attente du résultat de l'appel de somme(2)*
- somme(2) = 2 + somme(1) *se met en attente du résultat de l'appel de somme(1)*
- somme(1) = 1
- somme(2) = 2 + 1 =3 *reprise de l'appel de la fonction avec la valeur renvoyée 1*
- somme(3) = 3 + 3 = 6 *reprise de l'appel de la fonction avec la valeur renvoyée 3*
- somme(4) = 4 + 6 = 10 *reprise de l'appel de la fonction avec la valeur renvoyée 6*

Exercice

Écrire la fonction récursive somme en respectant la structure suivante:

```
def somme(n):
    if (condition d'arrêt):
        return valeur
    else:
        return (appel récursif)+valeur
```

[3]:
```
# fonction récursive somme
def somme(n):
    if n==1:
        print("somme(1)=1")
        return 1
    else:
        print("somme(%s)=somme(%s)+%s" % (n,n-1,n))
        return n+somme(n-1)

somme(4)
```

```
somme(3)=somme(2)+3
somme(2)=somme(1)+2
somme(1)=1
```

[3]: 10

Il est difficile d'imaginer la pile des appels et la remontée de cette pile avec les valeurs renvoyées. Il existe en ligne, une application qui permet de visualiser ces appels : **Python tutor**

Exercice

1. Rendez vous sur le site Python tutor.
2. Copier coller le code de votre fonction et l'appel **somme(4)** puis cliquer sur **visualiser**.
3. Procéder au déroulement du programme en cliquant sur **next** jusqu'aà la fin du programme.
4. Que se passe-t-il si on supprime le **return** dans les appels récursifs ?

0.1 Conclusion

- Une fonction récursive est une fonction qui s'appelle elle-même
- Une condition permet d'arrêter les appels récursifs de fonctions : c'est le **cas de base**.
- L'instruction **return** de la fonction permet de renvoyer une valeur qui sera utilisée par les autres appels. Si aucune valeur n'est à renvoyer, la fonction renvoie **None**.
- Les arguments passés dans les appels récursifs changent de valeurs jusqu'à arriver au cas de base.
- Attention, le nombre d'appels récursifs est limité par le système et le langage Python. Pour le connaitre, il faut importer le module **sys** puis exécuter la fonction `getrecursionlimit()` pour connaitre la valeur et utiliser la fonction `setrecursionlimit(valeur)` pour la modifier.

[4]:
```python
# Limite de récursion

import sys

print(sys.getrecursionlimit())
# sys.setrecursionlimit(3000)

def somme(n):
    if n==1:
        return 1
    else:
        return n+somme(n-1)

somme(2400)
```

```
3000
```

[4]: 2881200