

Boucles *for*

◇ Une boucle *for* répète une liste d'instructions un certain nombre de fois. Une telle boucle est décrite par 4 éléments :

- Une *valeur de départ* a ;
- Une *valeur d'arrivée* b ;
- Un *compteur* k qui prend toutes les valeurs de a jusqu'à b ;
- Une *liste d'instructions* qui sont exécutées pour chaque valeur de k .

◇ Dans un algorithme, on écrit :

```
pour  $k$  allant de  $a$  à  $b$ 
    Liste d'instructions
fin pour
```

Exemple Python. Afficher les carrés des entiers 1,2,3,4,5 et calculer la somme de ces carrés :

```
s=0
for k in range(1, 6):
    print k**2 # Rappel : k**2 = k puissance 2
    s=s+k**2
print "Le total :", s
```

1 4 9 16 25 Le total : 55

□

⚠ **Remarques.**

- La commande **range(a, b)** désigne l'ensemble des entiers k tels que $a \leq k < b$ (on peut retenir que a est la première valeur prise par k et b est la première valeur qui n'est pas prise par k). Ici, **range(1, 6)** représente l'ensemble des entiers de 1 à 5.
- Les instructions à répéter sont repérées par leur indentation. On notera que la dernière instruction de l'exemple ci-dessus ne fait par conséquent pas partie de la boucle (elle n'est exécutée qu'une seule fois).
- Les boucles *for* peuvent prendre des formes plus générales qui seront en partie vues plus tard.

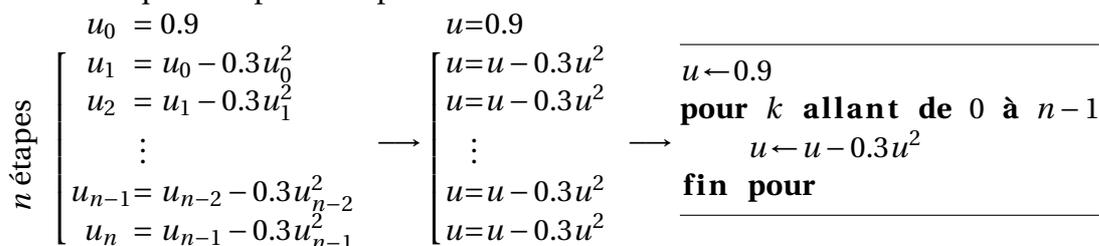
□

Application : calcul des termes d'une suite $u_{n+1} = f(u_n)$

◇ Considérons à titre d'exemple la suite (u_n) définie par :

$$u_0 = 0.9; \quad \forall n \in \mathbb{N}, u_{n+1} = u_n - 0.3u_n^2$$

En pratique pour calculer u_n , on calcule u_1 à partir de u_0 , u_2 à partir de u_1 , etc. jusqu'à u_n . Avec PYTHON, on va utiliser une même variable u qui va contenir successivement toutes les valeurs ce qui est représenté par le schéma suivant :



On doit donc répéter n fois l'instruction $u = u - 0.3u^2$, ce qui est fait au moyen d'une boucle *for* :

```
def u(n):
    U=0.9
    for i in range(0,n):
        U=U-0.3*U**2
    return U
```

On peut alors calculer u_5 par exemple :

```
print u(5)
```

0.340439309545

△ *Remarque.* On doit faire n étapes de calcul, donc il faut bien écrire **range(0, n)** et pas **range(0, n-1)**. On peut abrégé **range(0, n)** en **range(n)**. □

Remarque. On verra en TP des situations plus complexes ($u_{n+1} = f(n, u_n)$ ou $u_{n+1} = f(u_n, u_{n-1})$), on peut même imaginer que u_{n+1} s'écrit en fonction de u_0, \dots, u_n . □

◇ **À retenir.** Pour représenter graphiquement une suite, par exemple la suite (u_n) de l'exercice précédent pour $0 \leq n \leq 10$, on utilisera les commandes suivantes (dont la signification sera vue ultérieurement) :

```
import matplotlib.pyplot as plt
plt.clf()
plt.plot([u(n) for n in range(11)], '.')
```

```
plt.show()
```

