

Activités sur la récursivité :

Activité 1 : soit la fonction ci-dessous écrite en pseudo-code et en python qui demande une note tant que celle-ci n'est pas comprise entre 0 et 20

Fonction quelnote Demander (note) Tant que note<0 ou note>20 faire Demander (note) Fin tant que Retourner note	<pre>def quellenote(): note = float(input("Entrer note :")) while note<0 or note>20 : note = float(input("Entrer note :")) return note</pre>
--	--

Comment écrire cette fonction sans utiliser une boucle ?

	<pre>def quelnote(): note=float(input("entrer une note ")) if: else: </pre>
--	--

Activité 2 : un très mauvais exemple

C'est déjà une première chose à comprendre : un programme **peut** être appelé par lui-même, à l'intérieur de sa propre définition.

```
1 def prems():  
2     print("un très mauvais exemple")  
3     prems()
```

Si on appelle cette commande par la commande :

```
>>> prems()
```

Que se passe-t-il ? pourquoi ?

Activité 3 : Enfin un bon exemple :

```
1 def mystere(n):  
2     if n == 0 :  
3         return 0  
4     else :  
5         return n + mystere(n-1)
```

Trois choses sont essentielles et doivent se retrouver dans tout programme récursif :

- lignes 2 et 3 : le cas de base (si n vaut 0 on renvoie *vraiment* une valeur, en l'occurrence 0)
- ligne 5 : l'appel récursif
- ligne 5 : la décrémentation du paramètre d'appel

Que se passe-t-il lorsqu'on appelle mystere(4) ?

mystere(4) =

=

Que fait cette fonction ?