

# Quelques instructions d'AmiensPython



## Principes de base :

- Le nom du fichier ne doit pas comporter d'accents.
- Vos programmes doivent commencer par ces 2 lignes (automatiquement insérées)

```
__from future__ import division
from lycee import *
```

## Entrées/Sortie :

Afficher à l'écran :

```
print a           Affiche la valeur de a
print "a"         Affiche la lettre « a »
print "La valeur de a est :",a  Affichage mixte (texte et valeurs)
```

Demander un nombre et stocker la réponse dans une variable

```
x=demande("Nombre de côtés ?")  La question s'affiche et la réponse est attribuée à la variable x
```

Demander un texte :

```
reponse=texte_demande("Quelle est la couleur du cheval blanc d'Henry IV ?")
```

## Tests :

Programme demandé	Algorithme	Programme en Python	Remarques
Tester si un nombre entré est pair ou non : Nous allons regarder si le nombre est divisible par 2, c'est-à-dire si son reste vaut 0	<b>Demander</b> un nombre x <b>Si</b> le reste de $x \div 2$ vaut 0 <b>Alors</b> <b>Afficher</b> « OUI » <b>Sinon</b> <b>Afficher</b> « NON » <b>Fin du SI</b>	<pre>x=demande("nombre?"); if reste(x,2)==0 :     print "Nombre pair" else :     print "Nombre impair"</pre>	En Python, le ALORS se traduit par deux points et un décalage des instructions (alinéa). De même que pour le SINON qui se traduit par else :

## Opérateurs logiques :

== Egal à                      != Différent de                      >= Supérieur ou égal ....

## Boucles :

Comme dans la plupart des langages, il existe en Python principalement deux manières de réaliser une boucle, c'est à dire une répétition d'un bloc d'instructions. Comme pour la commande **si**, la partie à répéter sera indentée vers la droite, ce qui permet en plus une bonne visibilité de l'algorithme.

```
for v in ["a","e","i","o","u","y"] :
    ....BLOC D'INSTRUCTIONS...
Effectue la boucle, la variable l prenant à chaque tour
successivement les valeurs de la liste ( ici les voyelles )
```

```
for i in range(5) :
    ....BLOC D'INSTRUCTIONS...
( ici, tous les nombres de 0 à 4 (intervalle [ 0 ; 5 [ )
```

```
for i in range(3,10,2)
    ....BLOC D'INSTRUCTIONS...
Parcours l'intervalle [ 3 ; 10 [ avec un pas de 2 : 3, 5, 7,9
```

```
while i<10 :
    ....BLOC D'INSTRUCTIONS...
Exécute la suite d'instruction tant que i<10: le test sera effectué au
départ et à chaque fois que bloc d'instructions est fini, avant de
l'exécuter à nouveau
```