

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 si vous utilisez AmiensPython 1.9b ou supérieur) en sélectionnant « nouveau »)

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

Entrées/Sortie :

Afficher à l'écran :

<code>print a</code>	Affiche la valeur de a
<code>print "a"</code>	Affiche la lettre « a »
<code>print "La valeur de a est :",a</code>	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 :

```
x=demande("nombre?");
if reste(x,2)==0 :
    print "Nombre pair"
else:
    print "Nombre impair"
```

Affiche si le nombre entré est pair ou non :

Le symbole == indique un test d'égalité

La déclaration d'un bloc de commande se fait grâce à « : »

La fin du bloc est détectée par le retour d'indentation (alinéa)

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.

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

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

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

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