



## TESTS

```
if condition:  
    #traitement  
elif condition:  
    #traitement  
else:  
    #traitement
```

Opérateurs :  
and, or, et not

## BOUCLES

```
while condition:  
    #traitement  
  
for elt in [1, 2, 3]:  
    #traitement sur elt  
  
De 0 à n-1 :  
for i in xrange(n):  
    # traitement  
  
xrange(début, fin, pas)
```

## MODULES

```
import module  
import module as nom  
from module import *  
from module import f1, ...
```

Exécution si chargement direct :  
if `__name__` == `"__main__"`:

## ENTREE SORTIE

```
Python 2.7 :  
print "texte", var  
n = input("Entier :")  
c = raw_input("Chaîne")  
  
Python 3.2 :  
print("texte", sep=" ",  
end="\n")  
n = input("Entier ou  
Chaîne")
```

## CONSOLE

Arguments :  
import sys  
sys.argv[0] nom script  
sys.argv[1] 1er argument  
len(sys.argv) nb arguments

En-tête :  
# -\*- coding:utf-8 -\*-

## CHAINES

Chaîne c :  
c[début:fin:pas] slicing  
c[::-1] inverse

Concaténation : c + ".."

Conversion : str(i)

Non modifiables !

## LISTES

Liste l :  
l[début:fin:pas] slicing  
l.append(valeur) ajout  
l2=l[:] copie  
l2=copy.deepcopy(l) copie  
récursive  
valeur in l test

## DICTIONNAIRE

Définition :  
d={clé:valeur, ...}  
clé in d existence clé  
d.keys() liste clés  
d.values() liste valeurs  
d.get[clé,val] d[clé] ou val  
si d[clé] n'existe pas

## FONCTIONS

Définition :  
def fct(arg1, arg2, ...):  
 fct(arg2=..., ..) ordre  
\*args liste arguments  
\*\*args dico arguments  
Paramètres par référence

Documentation : <http://docs.python.org>