

Les bases du langage Python

Site Internet :
www.gecif.net

Type de document :
Fiche pratique

Intercalaire :

Date :

Python est un langage de programmation informatique, permettant d'écrire rapidement des petits programmes dans différents domaines [calcul scientifique, traitement de chaîne de caractères, tracé de graphiques, interface homme/machine, etc.]. La console Python permet de se familiariser avec les différentes instructions.

I - Les opérateurs arithmétiques

La console de Python peut s'utiliser comme une puissante calculatrice scientifique en utilisant les opérateurs suivants.

Les 7 opérateurs arithmétiques	
opérateur	opération effectuée
+	addition
-	soustraction
*	multiplication
/	division décimale [le résultat est un nombre décimal]
**	puissance
//	division entière [le résultat est un nombre entier]
%	reste de la division entière [le résultat est un nombre entier] : opérateur « modulo »

Remarque : l'affectation d'une valeur à une variable se fait avec l'opérateur =

Exemples :

```
a=23
b=a+4
c=17//3
d=17%3
e=2**3
f=(a+b)**2
g=4*9+3/7
h=(7+f)*8/3-((6-e)*4+1)-10**2
i=45.2389*10**-6
```

II - L'interface homme/machine

En Python la fonction permettant d'afficher un message à l'écran est **print**, et pour demander à l'utilisateur de saisir une valeur on utilise la fonction **input**. Le dialogue entre le programme et l'utilisateur se fait grâce à ces 2 fonctions.

Interface homme/machine	
fonction	rôle
print	affiche une chaîne de caractères à l'écran à destination de l'utilisateur
input	demande à l'utilisateur de saisir une chaîne de caractères au clavier

Remarque : la fonction **input** renvoie une chaîne de caractères. La fonction **int** convertit une chaîne en entier.

Exemples :

```
print("Bonjour")
print("a=",a, "et b=",b)
nom=input("entrez votre nom")
age=int(input("entrez votre age"))
print("Bonjour",nom,"vous avez",age,"ans")
```

III - Les boucles

En Python il existe 2 types de boucles, les voici :

Les boucles en Python	
boucle	principe
for	elle boucle tant que l'objet composé analysé renvoie des éléments
while	elle boucle tant que la condition testée est vraie

Exemples :

```
# affichage des nombres de 0 à 14 :
for i in range(15):
    print(i)
i=0
while i<15:
    print(i)
    i=i+1
# impose de saisir un nombre supérieur à 24 :
i=0
while int(i)<=24:
    i=input("entrez un nombre supérieur à 24")
print("Bravo ! Vous avez saisi",i)
```

IV - Le test if

En utilisant à la fois les opérateurs logiques **and**, **or** et **not** dans les formulations des conditions logiques complexes et les tests complets et imbriqués avec **if**, **elif** et **else** il est possible de tester finement toutes situations afin d'en détecter tous les cas particuliers recherchés.

Les conditions dans un if	
opérateur	test effectué
= =	égal à
! =	différent de
>	strictement supérieur à
> =	supérieur ou égal à
<	strictement inférieur à
< =	inférieur ou égal à
and	ET logique : le résultat est vrai si les 2 conditions sont vraies
or	OU logique : le résultat est vrai si au moins une des 2 conditions est vraie
not	NON logique : le résultat est vrai si la condition est fausse

Exemples :

```
m=12
if m==12:
    print("La variable m vaut douze")

if m%2==0:
    print("La variable m est paire")
else:
    print("La variable m est impaire")

if (m>=9) and (m<=35) or (m<0):
    print("m est compris entre 9 et 35, ou bien m est négatif")

m=53
if m>100:
    print("m est supérieur à 100")
elif m<10:
    print("m est inférieur à 10")
else:
    print("m est compris entre 10 et 100")
```

Prise de notes personnelles :