

Le langage Python

Notre premier langage

Prenons Python. C'est un langage moderne, complet et dans l'air du temps. Son cœur est petit prend peu de place par contre il demande souvent d'y associer un ou plusieurs compléments nommés "module".

Qu'est-ce qu'un module c'est une suite de routines (de fonctions) qui traitent un domaine. Le cœur de Python ne s'embarrasse pas de formules mathématiques qui servent à rien dans la programmation de base. Par contre nous avons le module "math" qui contient tout le nécessaire si l'on veut faire de la mathématique. Attention le cœur cependant connaît les quatre opérations.

Installation

Il faut installer Python et Notepad++ qui est un traitement de texte dédié à la programmation. Il opère une coloration correspondant à la syntaxe du langage employé. Cela permet de relever plus facilement les erreurs d'écriture.

Notre premier programme

```
# premier programme
print("Notre première ligne d'écriture")
input("Terminé")
```

- Une instruction par ligne c'est obligatoire.
- Une ligne débutant par dièse est ignorée du programme c'est un commentaire.
- On doit insérer input en dernière ligne sinon la fenêtre d'affichage se referme et nous ne voyons rien. input("texte") est une demande faite à l'opérateur tant que l'on a pas appuyé sur ENTREE la fenêtre reste ouverte.

Une addition

```
# une addition
a=3+9
print(a)
input("Terminé")
```

Pour afficher la variable, il ne faut pas la mettre entre double ou simple guillemet.

Une instruction conditionnelle

```
# Instruction conditionnelle
a=10
if(a<10):
    print('La valeur a est inférieure à 10')
else:
    print("La valeur vaut 10 ou est plus grande")
input("Terminé")
```

Remarquons que la condition se termine par deux points. La ligne suivant est indentée soit par une tabulation soit par des blancs. (L'un ou l'autre mais pas l'un et l'autre) else (sinon) est aligné avec if et se termine par deux points. La ligne suivante est encore indentée. Les lignes indentées qui se suivent forment un bloc qui sera exécuté en une seule fois. Dans les langages c'est habituel d'indenter les lignes pour rendre plus clair le raisonnement mais cela ne représente pas une obligation. Ici c'est une nécessité.

Un choix

Il semble qu'il n'existe pas d'équivalence dans ce langage.

La boucle for

Nous allons verrons les boucles for après l'étude des listes.

La boucle while

La boucle sera exécuter tant que la condition reste vraie.

```
# la boucle while
i=0
while i<5:
    print( str(i)+" ! ")
    i+=1;
input("Terminé")
```

Il ne faut pas oublier d'incrémenter la variable de boucle sinon la boucle est infinie.
Remarquez `i+=1` c'est un raccourcis de `i=i+1` par extension `i*=3 => i=i*3`

Les particularités de Python

Le langage est sensible à la casse.

`#` est le signe du commentaire

`\` est le signe d'échappement

`aa='C'` est le signe d'échappement'

`\n` retour ligne

`\` permet en fin de ligne de continuer l'instruction sur la suivante

Triple guillemets permet de donner la signification d'une fonction

Une chaîne est non modifiable, une liste l'est.

Fonction ou procédure défini par `def()` : (ne pas oublier l'indentation)

les variables

Les variables ne se déclarent pas mais doivent être initialisées avant emploi.

Le type de la variable peut changer au cours du programme.

Le type de variable doit correspondre au type de paramètre demandé par la fonction.

Types de variable : integer, float, string, boolean

les fonctions internes

`print()` est une fonction et les parenthèses sont obligatoires.

`type()` renvoie le type de la variable

`len(B)` donne les nombres de caractères d'une chaîne, donne le nombre d'éléments d'une liste.

`int(a)` `a` est une string et la fonction renvoie un integer

`Float(v)` convertit la string en réel

`a=input("Txt")` Demande de saisie au clavier de la part de l'opérateur. `a` est une chaîne. Elle doit être convertie si on veut un nombre.

`range(a,b,c)` employé dans une boucle for `a` valeur départ `b` valeur arrivée `c` incrémentation.

`chaîne.upper()` met la chaîne en majuscules

les affectations

`x=y=7` est affectation correcte multiple

`a,b=b,a` # *échange de valeur entre a et b sans passer par une 3ème variable.*

`a,b,c=1,2,3` reste valable

`10%3 => donne le modulo 1`

les chaînes de caractères

Une chaîne de caractères peut toujours être considérée comme une liste de caractères

`aa[1]` renvoie la seconde lettre de la chaîne. Une liste débute à l'indice zéro

`A="le temps"`

`B="est beau"`

`A+" "+B => le temps est beau` (concaténation tous les termes doivent être des strings)

les listes

Une liste est une suite d'éléments de toutes sortes y compris de liste.

`maliste=list()` => crée une liste vide.

`maliste=[]` => crée une liste vide.

`maliste=('Lundi','mardi','mercredi','jeudi','vendredi','samedi','dimanche')`

`maliste[1]` renvoie "mardi" une liste commence à l'indice zéro

`maliste[0]='Dimanche'` ici on remplace "Lundi" par "Dimanche"

`maliste.append('7')` => ajoute '7' à la fin de la liste

maliste.insert(2,'dimanche') => insert en position 2 'dimanche' (comme 3ème mot et repousse les suivants.)

del maliste(2) => on supprime le 3ème mot de la liste

maliste.remove('jeudi') => supprime 'jeudi' de la liste

Parcourir une liste peut être fait avec while mais préférable de le faire avec for.

```
for elm in maliste: # elm va prendre les valeurs successives des éléments de ma_liste
    print(elm)
maliste = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
for i, elm in enumerate(maliste):
    print("À l'indice {} se trouve {}".format(i, elm))
```

Avec enumerate() nous avons l'indice et la valeur

Remarquez le format {} endroit où sera inséré la variable de la liste format dans l'ordre indiqué.

Notez le point liant le texte au format.

i n'a pas été initialisé et vaut au départ zéro.

```
autre_liste = [
    [1, 'a'],
    [4, 'd'],
    [7, 'g'],
    [26, 'z'],
] # la liste sur plusieurs lignes (liste de listes)
for nb, lettre in autre_liste:
    print("La lettre {} est la {}e de l'alphabet.".format(lettre, nb))
```

Les tuples

Ce sont des listes non modifiables

```
ma_chaine = "Bonjour à tous"
ma_chaine.split(" ")
['Bonjour', 'à', 'tous']
```

split renvoie une liste de la chaîne découpée selon le séparateur fourni. (ici un blanc)