les types: type()

nombres entiers relatifs: *int()*
Nombres flottant(décimaux): *float()*
Booléens: *bool()*. (Seulement *True* et *False*)
Les chaines de caracteres: *str()*
Tableau dynamique: *list()*

Exemples: *>>> type(2*) renvoie *int*
>>>type(2.) renvoie *float*
>>>type(2<3) renvoie *bool*
>>>type("") renvoie *str*

Les variables

l'affectation de variable se fait avec =
exemples:
age=5 *>>> type(age)* renvoie *int*
poids=7.2 *>>>type(poids)* renvoie *float*

nom="khofi" *>>>type(nom)* renvoie *str*

YeuxBleu=False *>>>type(YeuxBleu)* renvoie *bool*

opérations sur les chaines de caractères

- *bin(a)* retourne une chaine de **caractère** qui représente a en binaire
- *chaine[n:p]* retourne la chaine contenue dans "chaine" entre le nième exclus et le pième caractère inclus
- *chaine[p:]* enlève les p premiers caractères
- *chaine[:p]* enlève les p derniers caractères
- *len(chaine):* nombres de caractères *chaine*

Opérateurs mathématiques

- +: addition et concaténation
- -:soustraction
- *:multiplication
- /:division décimal
- //: division euclidienne
- %: reste de la division euclidienne

remarque sur la concaténation:
exemple:

```
>>>A="Chapi"
>>>B="Chapo"
>>>C=A+" "+B
>>>print C
```

Chapi Chapo

Affichage

Ces trois méthodes donnent le même résultat

```
age=2
print "LouLou a "+str(age)+"ans"
print "LouLou a ",age,"ans"
print "LouLou a %d ans" %age
```

Affichage par chaine de formatage

```
print "Message à trous" % (val1,val2,val3)
```

les formats sont:

- *%d*: entier relatif
- *%f*: flottant
- *%s*: chaine de caracteres
- *%r*: booléen

exemple

```
age=15
nom= "Loulou"
Taille=1,7
print "%s a %d ans et il mesure %f " %(nom,age,Taille)
```

Saisie

```
reponse=raw_input("question a poser")
```

Opérateurs de comparaison

- ==: egal à
- <>: différent de
- <:strictement plus petit que
- >:strictement plus grand que
- <=: plus petit que
- >=: plus grand que

Opérateurs logiques

- *and*
- *or*
- *not*

Instructions conditionnelles

if condition1: *Instruction1* *else:* *Instruction2*	*if condition1:* *Instruction1* *elif condition2:* *Instruction2* elif *condition3:* *Instruction3* Else: *Instruction4*

Les Exceptions

Les problèmes dans un programme ce sont les bugs. On voudrait faire tourner même s'il y a un bug. On peut pour cela utiliser la syntaxe:

try:
instruction1
except:
instruction2

Boucle Pour...faire

La syntaxe de cette boucle est un peu inhabituelle dans python. Je fais un tableau comparatif:

Algorithmique	Python
Pour i=p à n faire *Sequence* *FinPour*	*for i in range(p,n-p+1):* *Sequence*

En fait, Python crée un tableau; range(p,q) est un tableau contenant q–p valeurs. la première valeur vaut p et ensuite elle est incrément de 1 en 1. Quand on écrit *for i in range(0,n+1), i prend successivement toutes les valeurs du tableau.*

Boucle Tant...que

La syntaxe est:

while bouleen:

sequence

Les fonctions

def nomFonction(arg1,...,arg2):
corpsFonction

Les tableaux dynamiques (list python)

Construction:*[val1,val2,...,valn]* crée un tableau indicé de 0 à n–1
type: *list()*

exemple:

>>>*list("toto")*

['t','o','t','o']

cas particulier de l'intervalle d'entiers: *range(n,p,q)*

opération sur les tableaux (remarquer que cela ressemble beaucoup à une chaine de caractère)

- appartenance: *valeur in Tableau*
- indice: *Tableau[i]* (attention le 1er indice est 0)
- Découpage (même chose que pour les chaines de caracteres)
- *opération:*
  tableau[n:p] retourne le tableau contenue dans "tableau" entre le nième exclus et le pième indice inclus
  tableau[p:] enlève les p premiers indices
  tableau[:p] enlève les p derniers indices

- longueur: *len(tableau)*
- affectation: *tableau [i]=valeur*
- Ajout d'une valeur: *tableau.append(v)*