

Introduction à Python

Maupetit Julien

19 septembre 2007

1 Langage de programmation

1.1 Définition

[Wikipedia](#) nous dit que :

Un langage de programmation est un code de communication, permettant à un être humain de dialoguer avec une machine en lui soumettant des instructions et en analysant les données matérielles fournies par le système, généralement un ordinateur. Le langage permet à la personne qui rédige un programme, de faire abstraction de certains mécanismes internes, généralement des activations et désactivations de commutateurs électroniques, qui aboutissent au résultat désiré.

L'activité de rédaction du code source d'un programme est nommée programmation. Elle consiste en la mise en oeuvre de techniques d'écriture et de résolution d'algorithmes informatiques, lesquelles sont fondées sur les mathématiques. À ce titre, un langage de programmation se distingue du langage mathématique par sa visée opérationnelle (une fonction et par extension, un programme, doit retourner une valeur), de sorte qu'un « langage de programmation est toujours un compromis entre la puissance d'expression et la possibilité d'exécution. »

1.2 Historique

- 40's les premiers langage très bas niveau font leur apparition
- 1957 - FORTRAN (*mathematical FORMula TRANslating system*)
- 1972 - C (Kernigan & Ritchie)
- 1975 - BASIC (Gates & Allen)
- ...
- 1987 - Perl
- 1991 - Python

1.3 Les différents langages

Il existe différents types de langages de programmation, et plusieurs manière de les classifier. Classiquement, nous pouvons distinguer les langages :

- compilés (C, Fortran ...) et interprétés (Perl, Python, Java, PHP, ...)
- non objet (C, Fortran ...), orientés objet (Perl, Python) et purement objet (C++, ...).

Langage interprété ... contrairement au compilateur, l'interprète exécute les instructions du programme (ou en évalue les expressions), au fur et à mesure de leur lecture pour interprétation. Du fait de cette phase sans traduction préalable, l'exécution d'un programme interprété est généralement plus lente que le même programme compilé. La plupart des interprètes n'exécutent plus la chaîne de caractères représentant le programme, mais une forme interne, telle qu'un arbre syntaxique.

Langage compilé En pratique, un compilateur sert le plus souvent à traduire un code source écrit dans un langage de programmation en un autre langage, habituellement un langage d'assemblage ou un langage machine. Le programme en langage machine produit par un compilateur est appelé code objet.

Programmation orientée objet C'est un assemblage de briques logicielles appelées objets. Chaque objet possède des données qui décrivent sa structure interne : ses attributs. La réponse à la réception d'un message par un objet est appelée une méthode ; elle décrit comment est réalisé le message.

2 Le langage Python

Les liens indispensables :

- Pour vous mettre en jambes <http://condor.ebgm.jussieu.fr/~fuchs/python/>
- Pour aller plus loin <http://diveintopython.adrahon.org/>
- La documentation officielle <http://www.python.org/>

2.1 Les variables

Les différents types de variables en python sont : les entiers, les flottants, les chaînes de caractères, les listes, tuples et dictionnaires. En python, les variables sont déclarées et initialisées en même temps.

```
>>> x = 2
>>> x
2
>>>
>>> x = 3.14
>>> x
3.1400000000000001
>>>
>>> pdb = "1abz"
>>> pdb
'1abz'
```

Nous pouvons réaliser des opérations sur les variables.

```
>>> a = 18
>>> a / 2
9
>>> a = "waza"
>>> b = a + "a" * 10
>>> b
'wazaaaaaaaaaa'
```

et afficher les valeurs qu'elles contiennent.

```
>>> seq = "ATGCTCGCGGCGCTAGCTACTAGCTAGCA"
>>> print seq
'ATGCTCGCGGCGCTAGCTACTAGCTAGCA'
>>> print "%5s %3d %s" % ( "MaSeq", len(seq), seq )
MaSeq 29 ATGCTCGCGGCGCTAGCTACTAGCTAGCA
```

ou convertir leur type

```
>>> x = "12.3"
>>> print x
'12.3'
>>> float(x)
12.300000000000001
```

Exercice 1 :

- Calculer la surface d'un disque de 10 cm de diamètre en utilisant deux variables : Pi et D, le diamètre.
- Générez une séquence polyA de 100 bases sans taper les 100 caractères.

Les listes sont des objets qui peuvent contenir des valeurs de différents types.

```
>>> maliste = [ "foo", 64, "spam" ]
>>> maliste[0]
'foo'
>>> maliste[1] = 12
>>> maliste
[ 'foo', 12, 'spam' ]
>>> machaine = "AGADOU"
>>> machaine[1]
'G'
>>> machaine[-1]
'U'
>>> maliste[0:2]
['foo', 12]
>>> len(maliste)
3
>>> range(0,10,1)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> residuA = [ "Atome1", "Atome2" ]
>>> residuB = [ "Atome3", "Atome3" ]
>>> peptide = [ residuA, residuB ]
>>> peptide
[['Atome1', 'Atome2'], ['Atome3', 'Atome3']]
```

2.2 Les boucles et structures de contrôle

La boucle for.

```
>>> peptide = ['PRO', 'GLY', 'ALA', 'ARG', 'ASN', 'ASP', 'CYS', 'GLN', 'GLU', 'HIS']
>>> for i in peptide[3:7]:
...     print i
...
ARG
ASN
ASP
CYS
```

Les différents opérateurs de comparaison.

Opérateur	Signification
==	égal à
!=	différent de
>	supérieur à
>=	supérieur ou égal à
<	inférieur à
<=	inférieur ou égal à

```
>>> x = 5
>>> x == 5
True
>>> x > 10
False
>>> x < 10
True
```

La boucle while.

```
>>> i = 1
>>> while i <= 5:
...     print i
```

```

...   i = i + 1
...
1
2
3
4
5
>>>

```

Les tests.

```

>>> x = 2
>>> if x == 2:
...   print 'Le test est vrai !'
...
Le test est vrai !
>>> x = 'hipopotame'
>>> if x == 'hipopotame':
...   print 'Le test est vrai !'
...
>>>
>>> x = 2 ; y = 2
>>> if x == 2 and y == 2:
...   print "le test est vrai"
...
le test est vrai
>>>
>>> if x == 2 or y == 3:
...   print "le test est vrai"
...
le test est vrai
>>>

```

Exercice 2 :

- Calculez le pourcentage de chaque type de base dans la séquence suivante ATGCTCGCGGCGC-TAGCTACTAGCTAGCA.
- Calculez le barycentre des 10 premiers carbones alpha de la crambine (code PDB 1crn) à partir de la liste suivante :
Indice : les coordonnées x, y et z se situent aux indices 31 - 38 pour X, 39 - 46 pour Y, 47 - 54 pour Z (en partant de 1).

```

trace_ =
["ATOM_16_2_CA_THR_1_16.967_12.784_4.338_1.00_10.80_1CRN_71",
 "ATOM_13_9_CA_THR_2_13.856_11.469_6.066_1.00_8.31_1CRN_78",
 "ATOM_13_16_CA_CYS_3_13.660_10.707_9.787_1.00_5.39_1CRN_85",
 "ATOM_10_22_CA_CYS_4_10.646_8.991_11.408_1.00_4.24_1CRN_91",
 "ATOM_9_28_CA_PRO_5_9.448_9.034_15.012_1.00_4.25_1CRN_97",
 "ATOM_8_35_CA_SER_6_8.673_5.314_15.279_1.00_4.45_1CRN_104",
 "ATOM_7_41_CA_ILE_7_8.912_2.083_13.258_1.00_6.33_1CRN_110",
 "ATOM_5_49_CA_VAL_8_5.145_2.209_12.453_1.00_6.93_1CRN_118",
 "ATOM_5_56_CA_ALA_9_5.598_5.767_11.082_1.00_3.56_1CRN_125",
 "ATOM_4_61_CA_ARG_10_8.496_4.609_8.837_1.00_3.38_1CRN_130"]

```