

I) Les variables

Le but de ce paragraphe est de se familiariser avec la notion de variable.

Une variable est un bout de mémoire consultable et modifiable. La création d'une variable en Python se fait lors de la première affectation.

Exercice 1

Taper dans l'interpreteur actif :

```
x=3
```

```
print(x)
```

Ouvrir un nouvel interpreteur, et taper

```
print(x)
```

Que se passe-t-il ? Pourquoi ?

Morale : Une variable est définie dans l'univers d'un programme, ici un interpreteur donné. Un autre interpreteur sera totalement indépendant du précédent, sans aucun lien : ils ne se parlent pas. Les variables de l'un sont inconnues de l'autre, ce sont deux bulles étanches.

En particulier, quand vous exécutez un programme (depuis l'éditeur) il vaut mieux le faire dans un *nouvel interpreteur*, pour qu'il n'y ait pas de variable déjà affectée qui traîne et le perturbe. □

Exercice 2

Voici une suite de commande, qui seront exécutées dans un interpreteur :

```
x=3
```

```
y=x
```

```
x=2
```

Prédire les valeurs de x et de y puis vérifier.

Morale : Les commandes sont exécutées dans un certain ordre, et cet ordre a une importance ! □

Exercice 3

Prédire et vérifier, dans chacun des cas suivant, la valeur des variables :

1) $x=42$

$y=10$

$x=y$

$y=x$

2) $x=42$

$y=10$

$z=x$

$x=y$

$y=z$

3) $(x, y) = (42, 10)$

$(x, y) = (y, x)$

La dernière ligne de code du 3) est une affectation simultanée.

II) Les types

Le type d'une variable s'affiche avec la commande `type()`.

Une variable est un nom sur une case de la mémoire ayant une certaine taille, donc sur une suite de 0 et de 1. Pour que la machine lise cette suite de 0 et de 1 comme un entier, un nombre à virgule flottante, ou d'autres type de données (chaînes de caractères, etc...), il faut lui expliquer. C'est ce qu'on appelle le *type* de la variable.

Définition 1 Un type est une étiquette sur la variable, qui va décrire comment la manipuler (et parfois aussi les valeurs possibles de cette variable : IEEE 754 pour les flottants). Le type peut être déclaré a priori par l'utilisateur (typage explicite) ou détecté par la machine (typage implicite).

Remarque : En Python, le typage est implicite. □

1) Entiers (integer : int)

Liste des commandes à connaître :

Opérations internes à l'ensemble des entiers (rend un entier) :

| | | | |
|---|----------------|----|-------------------------------------------|
| + | addition | ** | puissance, avec la convention $0^{**}0=1$ |
| - | soustraction | // | quotient de la division euclidienne |
| * | multiplication | % | reste de la division euclidienne |

Exercice 4

Tester les enchaînements de commandes suivantes :

- | | | |
|--------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------------|
| 1) $x=4$ $n=5$ $y=x**n$ $z=(y-1) \% 3$ | 2) $x=4$ $n=50$ $y=x**n$ $z=(y-1) \% 3$ | 3) $x=4$ $n=500$ $y=x**n$ $z=(y-1) \% 3$ |
|--------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------------|

Utilisateurs avancés : faire une boucle dans l'éditeur pour tester tous les n de 1 à 1000 (même si une boucle `for` est plus adaptée, on pourra reprendre dans un premier temps la boucle `while` du TP précédent).

Morale : Lorsqu'on manipule un type de variable (très exactement une *classe*), taper le nom de la classe suivie d'un point dans l'interpréteur donnera la liste des méthodes associées à cette classe, c'est à dire toutes les commandes associées à cette classe. □

Opérations booléennes (rend True ou False) :

| | | | |
|----|------|----|--------------|
| == | égal | != | différent de |
| <= | ≤ | < | < |
| >= | ≥ | > | > |

Exercice 5

Tester les commandes suivantes : $3==5$, $841>0$, $2**10==1024$.

Quelle est l'ordre d'exécution des commandes dans la dernière instruction ?

Exercice 6 (Pour aller plus loin)

Taper `int.` (bien mettre le point) dans un interpréteur. La liste des « méthodes » associées aux entiers (`int`) apparaît. Reconnaître (avec l'aide) les commandes déjà testées.

2) Réels (float)

Liste des commandes à connaître. Les opérations internes à l'ensemble des float (rend un float) sont les mêmes que pour les entiers, sauf la division :

| | | | |
|---|------|----|----------|
| + | idem | ** | idem |
| - | idem | / | division |
| * | idem | | |

Les opérations booléennes (rend True ou False) sont les mêmes que pour les entiers.

Exercice 7 (Différentes façon de passer en virgule flottante)

Observer dans un interpréteur l'effet des commandes. Quel est le type du résultat ?

- | | | |
|------|-------------|-------|
| 1) 1 | 2) float(1) | 3) 1. |
|------|-------------|-------|

Exercice 8 (Le dépassement dans l'infiniment petit)

Observer la différence entre

$2**60+1-2**60$ et $2**60+1.-2**60$

Faire un programme dans l'éditeur qui affiche les nombres et le type des nombres calculés par Python dans chacun des deux cas.

Exercice 9 (Des bizarreries)

Observer le résultat de :

$1.1+2.7$, $1.1*2.7$, $(1./3**2)*3**2$ $(1./3**6)*3**6$ et $.1+.1+.1==.3$

Expliquer le phénomène. Proposer une meilleure approche pour le dernier test.