2 Variables, expressions et fonctions

I. Les nombres et les opérations

◇ Tout nombre comportant un point décimal ou donné en notation scientifique est considéré comme un *float* (approximation d'un nombre réel). Dans le cas contraire, le nombre est entier (*int*).

```
>>> 0.001
0.001
>>> 2.2e-3
0.0022
```

```
>>> 1e3
1000.0
>>> type(1e3)
<class 'float'>
```

```
>>> 12345
12345
>>> type(12345)
<class 'int'>
```

◇ On a les opérations usuelles `+`,`-`,`*`. On a également `%` (le reste de la division *euclidienne*), **`abs`** (valeur absolue).

⚠ La puissance est `**` et pas `^`.

```
>>> 10**2
100
>>> 10^2
8
```

⚠ La division `/` est la division usuelle. L'opération `//` donne le quotient dans la division euclidienne :

```
>>> 3.5/2.5
1.4
>>> 19/2
9.5
```

```
>>> 7/2
3.5
>>> 7//2
3
```

```
>>> 9//4
2
>>> 9%4
1
```

II. Les variables

◇ L'utilisation de variables est nécessaire dans la plupart des programmes. Une variable permet de faire référence à une valeur au moyen d'un nom. Il faut garder à l'esprit l'image

suivante : la mémoire de l'ordinateur est constituée de boîtes sur lesquelles sont collées des étiquettes et qui contiennent des données.

◇ Une variable est décrite par 3 éléments :

- Son *nom* : c'est l'étiquette de la boîte, c'est le moyen par lequel on peut faire référence à la valeur contenue dans cette boîte ;
- Sa *valeur* : c'est la donnée contenue dans la boîte ;
- Son *type* : c'est l'ensemble auquel appartient la valeur associée à cette variable. Pour l'instant, on ne connait que 2 types : *int* et *float* (le type *complex* sera vu en TP). Le type d'une variable conditionne les opérations que l'on peut réaliser avec cette variable. C'est PYTHON qui détermine le type d'une variable en fonction de sa valeur.

```
>>> x = 2
>>> print("Valeur de x :",x)
Valeur de x : 2
>>> y = x/100
>>> print("Valeur de y :",y)
Valeur de y : 0.02
>>> x = x+1
>>> print("Valeur de x :",x)
Valeur de x : 3
>>> print("Valeur de y :",y)
Valeur de y : 0.02
>>> type(x)
<class 'int'>
>>> type(y)
<class 'float'>
```

⚠ L'opération d'affectation (réalisée par le symbole **=**) permet de donner une valeur à une variable. C'est une erreur grave d'utiliser dans un programme une variable sans lui avoir auparavant donné une valeur.

◇ Les noms de variable que l'on utilisera commenceront systématiquement par une lettre majuscule ou minuscule (**a..zA..Z**) suivie éventuellement d'autres lettres, ou de chiffres (**0..9**) ou du caractère blanc souligné (_, *underscore*).

⚠ Pour rendre les programmes lisibles, il est *impératif* d'utiliser des nombres de variables explicites (par exemple **altitude** plutôt que **h** pour désigner une altitude). Noter que PYTHON fait la différence entre majuscule et minuscule.

Remarque. Dans l'interface de PYZO, la fenêtre *Workspace* permet de connaitre les variables qui sont actuellement définies dans la console (on retrouve les 3 informations *nom, type, valeur*). □

III. Les fonctions (introduction)

◇ Comme en mathématiques, on peut définir des *fonctions* qui reçoivent des *paramètres* et produisent un *résultat*.

⚠ Le concept de fonction est essentiel et présente de nombreuses subtilités (on lui consacrera une séance complète).

```
import math
def volume_cylindre(rayon,hauteur):
    base = math.pi*rayon**2
    return hauteur*base
print(volume_cylindre(1,2))
```

```
6.283185307179586
```