

## 1.2 Les données

La plupart des problèmes nécessitent le traitement de valeurs : certaines sont des données de l'énoncé, d'autres issues des calculs réalisés par l'algorithme.

Pour pouvoir être manipulées, les valeurs sont stockées dans des variables.

### 1.2.1 Définition de variable

Une variable désigne un emplacement mémoire qui permet de stocker une valeur.

Une variable est définie par :

1. Un nom unique qui la désigne
2. Un type de définition unique
3. Une valeur attribuée et modifiée au cours du déroulement de l'algorithme

### 1.2.2 Nom d'une variable

Le nom ou l'identifiant d'une variable permet d'identifier cette variable de manière unique au cours de l'algorithme.

#### Remarque 4 :

Pour faciliter la lecture des algorithmes, il faut respecter quelques règles de nommage de ces variables

1. Le nom d'une variable commence par une minuscule
2. Le nom d'une variable ne comporte pas d'espace
3. Si le nom de la variable est composé de plusieurs mots, il faut faire commencer chacun d'eux par une majuscule
4. Donner aux variables un nom explicite

#### Exemple 2 :

1. Exemple de nommage de variable : `maVariableAMoi` est un bon nommage alors que `Ma Variable A Moi` ne l'est pas du tout
2. Il m'arrive d'utiliser un mix de ces recommandations. J'écris souvent `Ma_Variable_A_Moi`<sup>1</sup>

### 1.2.3 Type d'une variable

Le type ou le domaine de définition d'une variable indique l'ensemble des valeurs que la variable peut prendre.

#### Remarque 5 :

1. Les variables peuvent appartenir à différents domaines : **entier**, **réel**, **caractère** ou **booléen**.
2. La valeur de la variable est la seule qui soit modifiée au cours de l'algorithme. Au début de l'algorithme (*au moment de la déclaration*), toutes les variables ont des valeurs inconnues.

### 1.2.4 Affectation d'une variable

L'affectation est une opération qui donne une valeur à une variable

#### Remarque 6 :

La syntaxe pour l'affectation diffère suivant les langages ou les constructeurs

1. Sur des calculatrices, c'est souvent : `←`
2. En Python, nous utiliserons `=`

---

1. J'ai horreur de commencer une phrase par une minuscule

**Exemple 3 :**

Nous cherchons à calculer le double d'un réel donné. On introduit donc deux variables associées respectivement à la donnée et au résultat.

```
#Algorithme "Double"
#Cet algorithme donne le double d'un nombre donné

>>> nombre=7           #On affecte le nombre 7 à la variable nombre
>>> nombre=nombre*2    #On multiplie la variable nombre par 2
>>> nombre             #On demande d'afficher la nouvelle valeur de la variable nombre
14                    #C'est la valeur donnée par la variable nombre
```

**Remarque 7 :**

Pour éviter bien des erreurs, suivez ces recommandations :

1. Une variable n'est déclarée qu'une seule fois dans un algorithme.
2. Avant de pouvoir utiliser la valeur d'une variable, une valeur doit lui être attribuée.
3. Attention : l'égalité mathématique diffère totalement de l'affectation informatique.

**Exercice 1 :**

On considère le programme suivant :

```
#algo mystere
>>> A=0
>>> B=4*A+3
>>> C=4*B+3
>>> A=B
>>> B=C
```

Quelles sont les différentes valeurs des variables en fin d'exécution du programme? (*Bien entendu qu'il est tout à fait possible de connaître le résultat en utilisant IDLE, mais, pour progresser il serait bon de le faire sur papier*)