CHAPITRE

3 Variables, types et valeurs

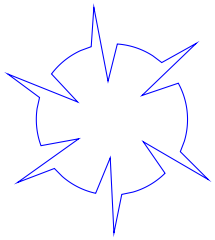

Table des matières

1 Introduction

Les variables sont des éléments incontournables dans les langages de programmation : elles servent essentiellement à ***stocker*** de l'information.

Selon les langages, leur déclaration est plus ou moins facile. Le langage Python a cet avantage de proposer des déclarations simplifiées contrairement à d'autres langages comme le C.

2 Nom des variables

Dans les langages informatique, les ***variables*** sont déclarées et souvent ***affectées*** d'une valeur. Exemples en Python :

```
age = int(input("Donner votre âge: "))
annee = 2024
nom = "Etienne"
```

Retenez :

Le signe = permet d'affecter une ***valeur*** a une ***variable***.

Le nom d'une variable est au choix de l'utilisateur, mais retenez :

Le nom d'une variable doit toujours être explicite : sa simple lecture doit nous faire comprendre l'information qu'elle stocke !

Donc, évitez des trucs comme :

```
a = int(input("Donner votre âge: "))
b = 2024
c = "Etienne"
```

Si vous voulez « composer » le nom de votre variable, il faut éviter l'utilisation d'***espaces*** et de tout signe de ponctuation. Il existe alors deux façons de procéder :

(Wikipédia) Le ***snake case*** est une convention typographique en informatique consistant à écrire des ensembles de mots, généralement, en minuscules en les séparant par des tirets bas(underscore). Cette convention s'oppose par exemple au ***camel case*** qui consiste à mettre en majuscule les premières lettres de chaque mot.

Moi, j'ai choisi le ***snake case*** pour l'écriture de vos programmes : il faudra choisir votre style.

```
ma_couleur_perso = "red"  #snake case
MaCouleurPerso= "red"   #camel case
```

JAMAIS d'accents dans les noms des variables !!

Ni dans les noms des fichiers d'ailleurs !

3 L'affectation

Affecter une variable, c'est lui donner une valeur. Il existe plusieurs façon de procéder.

3.1 Affectation manuelle

Quand au début d'un programme, on choisit d'attribuer des valeurs aux variables définies, ces affectations sont manuelles. Exemples :

```
couleur = "black"
longueur = 100
largeur = 153
```

On peut aussi demander à l'utilisateur de ***saisir*** une valeur au clavier et de la ***stocker*** dans une variable.

La fonction `input()` permet de saisir une information au clavier

Exemples :

```
age = int(input("Donner votre âge: "))
nom = input("Quel est votre prénom?")
```

Vous remarquerez certainement une petite différence entre les deux instructions pourtant voisines : l'absence du `int` pour le second. La raison en sera donnée un peu plus loin...

3.2 Affectation automatique

On peut affecter la valeur d'une variable à partir de la valeur d'autre(s) variable(s).

```
longueur = 15.1
largeur = 7.2
aire = longueur*largeur
perimetre = (longueur + largeur)*2
```

Une fois la variable ***initialisée***, sa valeur peut évoluer au sein d'un programme.

```
poids = 10
poids = poids + 1
```

Dans l'exemple précédent,

- la variable `poids` est initialisée à 10
- la ***nouvelle*** valeur de `poids` est ensuite égale à l'***ancienne*** à laquelle on ajoute 1.

L'opération qui consiste à faire évoluer la valeur d'une variable de 1 s'appelle l'***incrémentation***.

Cette opération est très utilisée en informatique. La décrémentation consiste à la diminuer de 1. On peut aussi intégrer ces transformations dans des boucles :

```
note = 10
for i in range(7):
    note = note +  2  # la variable note subit 7 hausses consécutives de 2 unités
```

On peut réduire l'instruction `note = note + 2` en `note += 2`, qui signifie la même chose.

4 Type de variables

Un programme informatique manipule des données stockées sous des formes différentes, des ***types*** différents. Pour l'instant, vous devez connaître trois types, objets de base en Python :

- le type ***entier***(`int`) qui représente des nombres ***entiers*** ;
- le type ***décimal***(`float`) qui représente des nombres ***décimaux*** ;
- le type ***chaînes de caractères***(`str`) qui représente des ***caractères***(lettres ou phrases)

Il est important de connaître le type des variables pour connaître le type d'opérations qui agissent sur celles-ci. On peut par exemple ajouter deux ***entiers***, deux ***chaînes de caractères*** mais pas un ***entier*** et ***une chaîne***.

On appelle souvent ***string***, une chaîne de caractères en informatique...

Pour affecter à une variable, une valeur numéraire aucune difficulté. En revanche, il faut utiliser la double quote `""` pour désigner un type ***string***.

```
1 age = 18
2 taille = 1.78
3 departement = "47"
4 ville = "AGEN"
```

La variable `age` est de type `int`, la variable `taille` de type `float`, la variable `departement` de type `str` car déclarée avec des `""` et la variable `ville` de type `str` car aussi déclarée avec des `""`.

Si vous faites l'instruction `nom = Jean`, vous n'affectez pas à la variable `nom` la valeur `Jean` !

Il manque en effet les double quotes... Si la variable `Jean` existe alors vous affectez sa valeur à celle de `nom` sinon vous provoquez une erreur...

Retour sur le ***input()***. L'appel de la méthode ***input()*** permet de stocker l'information saisie au clavier dans une une variable qui sera alors automatiquement du type ***str***.

```
1 age = input("Quel est votre âge?")
2 print(type(age))
```

Ces instructions affichent le type de la variable `age` qui sera alors un `str` ***même*** si vous tapez ***8*** au clavier. Si vous voulez que la variable `age` soit un nombre, il faut alors préfixer `input` de la mention `int` ou `float` selon la nature du nombre choisie.
Il existe d'autres types que nous rencontrerons cette année. Citons déjà :

- les ***booléens*** ;
- les ***listes*** ou les ***tuples*** ;
- les ***dictionnaires***

Mais ce sera dans un prochain cours...