

## 6.1.1 Variables

### Fiche d'exercices

Dans (presque) tous les exercices suivants, anticipez la valeur contenue dans chaque variable à la fin du programme. Vérifiez ensuite en exécutant le code (CTRL+ENTER).

#### 1. Exercice 1: affichage

```
In [ ]: a = 2
        b = a + 1
        a = 5
        c = 3 * a

        print(a, b, c)
```

**Remarque:** vous l'aurez compris, la fonction `print` sert à afficher une ou plusieurs variables (ou tout simplement une valeur)... À utiliser avec parcimonie, mais diablement utile pour déboguer un programme. Mais ça, c'est une autre histoire...

#### 2. Exercice 2: incrémentation

```
In [ ]: a = 2
        a = a + 1
        a = a + 1
        a = a + 1

        print(a)
```

**Remarque:** ajouter 1 (ou une autre valeur entière fixée) à une variable, de façon répétitive, s'appelle **incrémenter** la variable.

#### 3. Exercice 3

Dans cet exercice seulement, il ne s'agit pas d'anticiper les valeurs, mais d'écrire le code correspondant aux instructions:

1. On initialise une variable `score` à 100 puis on l'augmente de 15.
2. On initialise une variable `cellule` à 1 puis on la multiplie par 2.
3. On initialise une variable `capital` à 1000 puis on lui enlève 5%.

```
In [ ]: #1. Écrivez ci-dessous vos deux lignes de code

#2. Écrivez ci-dessous vos deux lignes de code

#3. Écrivez ci-dessous vos deux lignes de code
```

## 4. Exercice 4: échange?

```
In [ ]: a = 1
b = 2
a = b
b = a

print(a, b)
```

Comment modifier le code précédent pour vraiment échanger les valeurs des variables `a` et `b` ?



## 5. Exercice 5: opérations

Même consigne, avec un peu de calculs. Pour vous aider:

Loading [MathJax]/extensions/Safe.js

| Opérateur      | Symbole Python |  | Opérateur               | Symbole Python |
|----------------|----------------|--|-------------------------|----------------|
| Addition       | +              |  | Puissance               | **             |
| Soustraction   | -              |  | Quotient de la division | //             |
| Multiplication | *              |  | Reste de la division    | %              |
| Division       | /              |  |                         |                |

```
In [ ]: a = 2
        b = 15
        c = 6
        d = (b%c) ** 2
        e = b/c - b//c

        print(d, e)
```

## 6. Exercice 6: mélange de types...

Dans cet exercice, prévoir également le type de la variable.

```
In [ ]: a = 3 + 2.0
        b = 6 / 2
        c = 6 // 2
        d = 1.5 ** 3
        e = "py" + "thon"
        f = 3 * "miaou"
        g = 2 + "a"
```

## 7. Exercice 7: commentaires

Avec Python, tout ce qui suit le caractère dièse # (ALTGR+3) n'est pas interprété. On parle de **commentaire**.

En programmation, il y a deux usages:

- documenter/expliquer son code, pour faciliter la lecture (par soi plus tard, ou par quelqu'un d'autre);
- ne pas exécuter des lignes de code sans les supprimer (on pourra les décommenter en supprimant simplement le #), notamment en phase de debug.

```
In [ ]: # L'exécution de ce code va provoquer une erreur.
        # Essayez de deviner quelle instruction est fautive et commentez la ligne.
        # Vérifiez en exécutant la cellule

        lisa = 2
        marge = 5
        bart = lisa + 2*marge
```

```
maggie = bart / (10 - 2*marge)
homer = bart * "D'oh!"
```