

# **Éléments de programmation informatique**

**Nouveaux programmes de Physique-Chimie  
2019**

# 1- Qu'est-ce-que programmer ?

**Programmer** une machine consiste à lui communiquer une séquence d'instructions à exécuter qu'elle devra pouvoir reproduire à la demande :

Exemples : programmation d'un radio-réveil, d'un lave-linge, d'une feuille de calcul de tableur, ...

Élaborer un programme nécessite en premier lieu d'écrire un **algorithme** (suite de tâches élémentaires qui s'enchaînent selon des règles précises, sans place pour l'interprétation personnelle) ; cet algorithme doit ensuite être implémenté (traduit) dans un langage de programmation (Python, C/C++, Java, Javascript, assembleur etc).

Implémenter un algorithme nécessite l'écriture d'un **script** (ensemble d'instructions rédigées dans un langage donné) : ce script constitue le **code source** .

### Exemple de code source :

```
void main() {
    int A, B;
    WriteLine("entrez deux entiers : ");
    A = ReadIntFromKeyboard();
    B = ReadIntFromKeyboard();
    if (B == 0) {
        WriteLine("erreur : div par 0");
    } else {
        float C = CalculerDivision(A,B);
        WriteLine("le résultat est : " + C);
    }
}

float CalculerDivision(float U, float V) {
    float Resultat;
    Resultat = U / V;
    return Resultat;
}
```

## 4- La programmation par blocs

Certains outils permettent de manipuler des blocs graphiques qui dispensent de maîtriser la syntaxe du langage. Scratch, Mblock ou AppInventor sont largement utilisés au collège.



Se connecter sur <https://scratch.mit.edu/> (créer éventuellement un compte)

Importer le fichier mouvements.sb3 (disponible ici : .....)  
qui contient les différents éléments des programmes à remettre en  
forme pour animer la voiture bleue d'un mouvement rectiligne  
uniforme  $x = 25 t - 150$  et la voiture rouge d'un mouvement  
rectiligne uniformément varié  $x = 2,5 t^2 - 150$ .

NB : La barre d'espace permet de repositionner les voitures sur la  
ligne de départ et réinitialiser le chrono.

# Exercice Scratch

The image shows a Scratch workspace with a script on the left and a stage on the right. The script consists of the following blocks:

- mettre t à chronomètre
- aller à x: -150 y: 0
- aller à x:  $25 * t + -150$  y: 0
- réinitialiser le chronomètre
- quand la touche espace est pressée
- quand le drapeau est cliqué
- répéter jusqu'à ce que (loop)
- stop tout

The stage features a road with vertical lines at x = -150, x = 0, and x = 150. A red car is at x = -150, and a blue car is at x = 0. The timer 't' is displayed as 5.288.

Sprite: blue-carT80  
x: -18, y: 0  
Afficher:   Taille: 100 Direction: 90

Scène: Arrière-plans 1

Les scripts peuvent s'écrire à l'aide simplement d'un **éditeur de texte**

**Exemples :** Notepad++



, SublimeText



...

Un **environnement de développement intégré** ou IDE intègre un éditeur et des outils tels que des fenêtres d'exécution, d'exploration, de compilation...

**Exemples :** Spyder, Pyzo sont des IDE python





Certains langages dits **interprétés** permettent d'exécuter directement le script dans l'IDE  
Exemple : Python ou de java.

Dans le cas des langages **compilés** le code source doit être préalablement traduit en langage machine (assembleur)

- La compilation révèle les fautes de syntaxe. Il est important de bien lire les messages indiqués
- Sous Windows, de la compilation d'un script résulte un fichier exécutable (.exe)
- Un logiciel est dit libre lorsque son code source peut être consulté, modifié, redistribué en fonction de la **licence** originale apposée par l'auteur.

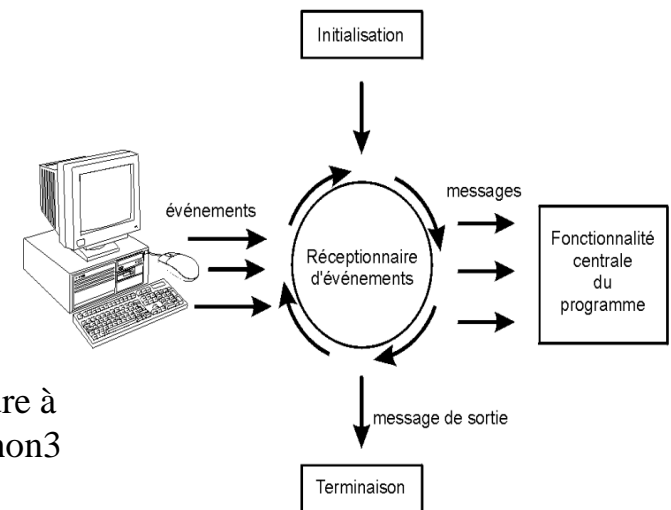
# 7- Types de programmation

Il existe plusieurs types de programmation, chacun étant plus ou moins adapté à un problème donné, parmi lesquels :

La **programmation impérative** (C, python,...) : traitements réalisés par des séquences d'instructions successives ;

La **programmation objets** (C++, Java, python,...) : basée sur la structuration des données

La **programmation évènementielle** (Arduino, python) : le programme doit réagir à des actions réalisées par l'utilisateur (clic sur un bouton...)



Illustrations : Apprendre à programmer avec Python3 de Gérard Swinnen

# 8- Structures de base de la programmation

- **les variables et leur affectation** : Une variable permet de stocker une information susceptible d'être modifiée au cours de l'exécution du programme. Certains langages exigent une déclaration préalable des variables et éventuellement leur typage (entier, flottant, chaîne de caractère ou booléen)

- Exemple d'affectation en python : `a=2`

- **la séquence ou bloc** : plusieurs instructions

- **l'instruction conditionnelle** (test) : Si(condition) alors (action 1) sinon (action2)

- **les boucles** (bornées ou non) : Pour i de min à max faire ....

Tant que (condition vraie) faire .....

- **la fonction** : sous programme, permettant de regrouper en un bloc et, sous un nom, un ensemble d'instructions. Une fonction peut être appelée à tout moment dans un programme. L'usage de fonction réduit la longueur du code, simplifie sa lecture et réduit les erreurs.

- **les commentaires** : ils ne sont pas exécutés comme les instructions, mais sont absolument nécessaires pour la compréhension, la communication et la maintenance du code.

# 9- Présentation du langage python

Python est un “langage interprété”, c'est-à dire qu'il nécessite un programme auxiliaire : « l'interpréteur » pour interpréter, c'est-à-dire traduire les instructions (le code source) en langage machine qui lui seul est à même d'être exécuté par le processeur. Le programme python.exe est cet interpréteur.



Un script (programme) python est une suite d'instructions écrites avec une syntaxe propre au langage ; cette suite d'instructions doit être enregistrée dans la plupart des cas dans un fichier d'extension py, par exemple *mon\_programme.py*.

Les **IDE** (environnements de développement intégrés) : Spyder, Pyzo, JupyterLab, PyScripter, Idle ... facilitent la rédaction, le test et le debugage du script.

Le langage python offre par ailleurs une multitude de possibilités (calcul numérique, liaison série, tracés, bases de données, calcul formel, traitement d'images et de sons, développement de jeux et de sites web ...). Ces possibilités, offertes par des fonctions développées par de nombreux contributeurs, sont regroupées par thématiques dans des bibliothèques (packages).

Ces bibliothèques, pour être disponibles, doivent être installées sur l'ordinateur, et il est nécessaire de les « charger » quand le script rédigé nécessite l'usage d'une fonction appartenant à l'une de ces bibliothèques.

# 9- Installation de python

Pour installer python sur un ordinateur, le plus simple est d'avoir recours à une « distribution ». Une distribution est un fichier exécutable à télécharger sur le web. A l'installation, l'interpréteur python sera bien évidemment installé et, suivant la distribution, un ou plusieurs IDE seront également installés. Certaines distributions installent aussi plusieurs bibliothèques. Ce point est important à prendre en compte car la procédure d'installation postérieure d'une nouvelle bibliothèque peut être déroutante pour le débutant.

En résumé, installer une distribution revient en une seule fois à installer :

- l'interpréteur python.exe
- un ou plusieurs IDE
- potentiellement des bibliothèques

Quelques distributions python :

Attention : choisir python version 3, et, suivant l'ordinateur, la version 32 bits ou 64 bits ainsi que la version adaptée au système d'exploitation (windows, linux, macOS).

- <https://www.anaconda.com/distribution/>
- <http://winpython.sourceforge.net/>
- <https://edupython.tuxfamily.org/>
- <https://sourceforge.net/projects/portable-python/>