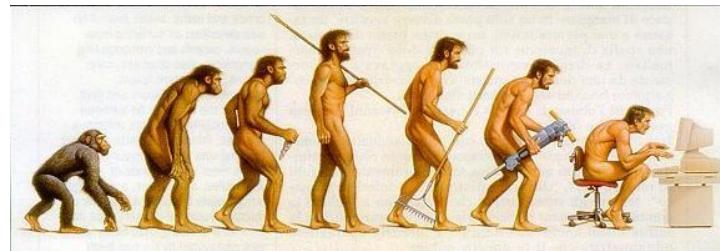




# Introduction à la programmation

➤ L'objectif de ce chapitre est de découvrir les notions fondamentales liées à la programmation.



## ➤ Un programme, c'est quoi ?

La rédaction et l'impression d'un courrier, le calcul des intérêts d'un prêt, la recherche d'informations sur internet, ou encore la possibilité de communiquer avec des amis à l'autre bout du monde : tous ces actes courants sont possibles grâce aux **ordinateurs**. (au sens large : PC, tablettes, smartphones, calculateurs embarqués, ...).

Classiquement, un ordinateur est une **machine** capable d'exécuter automatiquement une série d'opérations simples qu'on lui a demandées. Il se contente de **faire ce qu'on lui dit de faire**. L'intérêt des ordinateurs est de savoir manipuler très rapidement et sans erreur d'énormes quantités d'informations.

Remarque : Il existe aujourd'hui des systèmes dotés d'une capacité d'apprentissage, de jugement, d'improvisation, à l'image du fonctionnement de notre cerveau : on parle alors d'intelligence artificielle (I.A.). Ce domaine en très fort développement utilise des algorithmes inspirés du fonctionnement de notre cerveau qu'on nomme réseaux de neurones.

Une intervention humaine est nécessaire pour qu'un ordinateur puisse accomplir des tâches utiles. C'est le rôle du **programmeur** (appelé également **développeur**). Il va fournir les ordres que la machine doit exécuter en écrivant des programmes.

De manière générale, un programme sert à **automatiser un traitement sur des données**. Son résultat dépend des données qui lui sont fournies.

On peut schématiser un programme de manière suivante : **E-T-R** (Entrées-Traitements-Résultats).

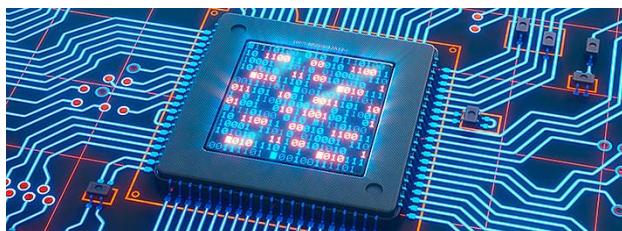


L'objectif du programmeur est donc d'écrire un programme qui, une fois exécuté par l'ordinateur, donnera les résultats attendus en fonction des données d'entrée.



# ➤ Les langages de programmation

## 1. Les langages de bas niveau



Dans un système de traitement de l'information, c'est le microprocesseur qui traite les informations. Un microprocesseur est un composant électronique qui comporte un certain nombre de fonctions utiles au traitement des données. Ces fonctions sont identifiées par un code binaire.

Vue du microprocesseur, un programme est une suite de codes binaires et de valeurs exprimées également en binaire. On peut donc écrire directement un programme en binaire ou dans un langage très proche qu'on appelle **assembleur**.

Ce type de programmation est complexe et long à mettre en œuvre. Ci-contre l'exemple d'un programme affichant simplement **Hello, world !**

Par contre, le fait de programmer directement le microprocesseur permet d'optimiser le temps d'exécution du programme.

```
cseg    segment
assume cs:cseg, ds:cseg
org    100h
main   proc
        mov dx, offset Message
        mov ah, 9      ; Fonction afficher une chaîne.
        int 21h       ; Appel à MS-DOS
        ret          ; Fin du programme
main   endp
Message db      "Hello, world!"
cseg    ends
end     main
```

## 2. Les langages de haut niveau

On appelle **langage de haut niveau** un langage qui permet d'écrire un programme en utilisant des « **mots clés** » issus du langage naturel (en Anglais principalement). On distingue deux types de langages de haut niveau.

### a. Langages compilés

Un compilateur est lui-même un programme capable de traduire un code écrit en langage de haut niveau en code objet binaire. On obtient le programme sous forme de fichier binaire exécutable directement par le microprocesseur (fichier .exe sous Windows). Pour un même algorithme, les langages compilés sont toujours les plus rapides.

Les principaux langages compilés sont les langages **C** et **C++**. Les systèmes d'exploitation (Linux, Windows, MacOs, ...) et de nombreuses applications sont codées en C ou C++.

**Java** est une sorte de langage compilé dont le code objet (on parle de byte-code) est destiné à fonctionner sur une machine virtuelle java. Ses performances sont moins bonnes que celles obtenues avec un code en langage C mais comme le code est exécuté sur une machine virtuelle, il ne dépend pas du système utilisé.

```
#include <stdio.h>

int main()
{
    printf("Hello, World!");
    return 0;
}
```

Java est également très populaire chez les programmeurs. C'est le langage de programmation utilisé pour Android par exemple.



### b. Langages interprétés

Un interpréteur est un programme qui traduit et exécute au fur et à mesure les mots clefs et structures du langage. Le code doit être interprété à chaque exécution ce qui rend les langages interprétés moins rapides que les langages compilés.

Cependant, avec l'évolution des capacités de calcul et de la vitesse des microprocesseurs, les langages interprétés sont aujourd'hui suffisamment rapides pour coder des applications complexes avec des temps d'exécution satisfaisants.

Comme on peut écrire un interpréteur pour chaque système, les programmes écrits en langage interprétés peuvent fonctionner sur tout type de machines.

Le Basic, PHP, Python, sont des langages interprétés.

```
# Code python pour afficher Hello, World!
```

```
Print ('Hello, World!')
```



## ➤ Glossaire :

**Algorithme** : suite de raisonnements ou d'opérations de calculs permettant de trouver la solution à un problème en un nombre fini d'étapes.

**Programme** : ensemble d'instructions définissant les opérations qui devront être exécutées par l'ordinateur pour résoudre un problème.

**Programmation** : ensemble des activités orientées vers la conception, la réalisation, le test et la maintenance de programmes.

**Langage informel ou pseudo-code** : un moyen de présenter l'algorithme décrit pour résoudre un problème sans trop se soucier de la grammaire et du vocabulaire propres au langage de programmation utilisé.

**Algorigramme** : représentation graphique d'un programme

SYMBOLE	DESIGNATION	SYMBOLE	DESINATION
<b>Symboles de traitement</b>		<b>Symboles auxiliaires</b>	
	<b>Symbolé général</b> Opération ou groupe d'opérations sur des données, instructions, pour laquelle il n'existe aucun symbole normalisé.		<b>Renvoi</b> Symbole utilisé deux fois pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée.
	<b>Sous-programme</b> Portion de programme considérée comme une simple opération.		<b>Début, fin, interruption</b> Début, fin ou interruption d'un algorigramme.
	<b>Entrée - Sortie</b> : Mise à disposition d'une information à traiter ou enregistrement d'une information traitée.		<b>Commentaire</b> Symbole utilisé pour donner des indications marginales.
<b>Symbol de test</b>		Les différents symboles sont reliés entre eux par des lignes de liaisons.	
	<b>Décision</b> : Exploitation de conditions variables impliquant le choix d'une voie parmi plusieurs. Symbole couramment utilisé pour représenter une décision ou un aiguillage.		
<b>Sens conventionnel des liaisons</b>			