

Warning: Ce cours est un cours d'ouverture d'esprit au numérique et à la programmation. Les définitions ne sont donc pas complètes. Il y a donc entre les différentes étapes que je décris dans ce cours concernant l'ordinateur des processus bien plus compliqués que ceux que j'avance. Pour ceux qui veulent aller plus loin, je suis à votre disposition. J'aime quand on me pose des questions, cela m'aide à m'améliorer. Voilà, vous êtes prévenus.

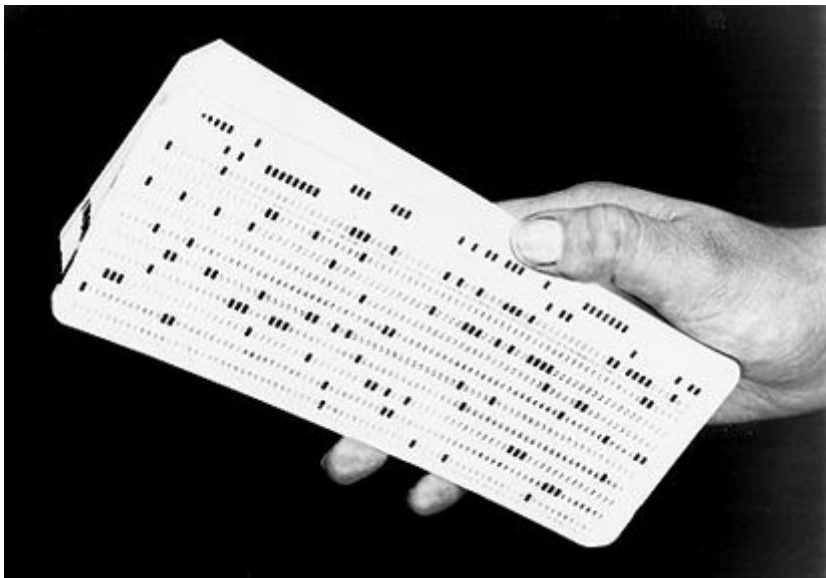
L'échelle de relation Ordinateur - System - Code:

- Partie historique

La programmation a déjà une longue histoire derrière elle. Au début, il n'existait même pas de clavier pour programmer ! On utilisait des cartes perforées pour donner des instructions à l'ordinateur.

extrait repris de

<https://openclassrooms.com/fr/courses/1894236-programmez-avec-le-langage-c/1894377-quest-ce-que-le-c>



- "1958 : il y a longtemps, à l'époque où les ordinateurs pesaient des tonnes et faisaient la taille de votre maison, on a commencé à inventer un langage de programmation appelé **l'Algol**.
- 1960-1970 : ensuite, les choses évoluant, on a créé un nouveau langage appelé le **CPL**, qui évolua lui-même en **BCPL**, puis qui prit le nom de **langage B** (vous n'êtes pas obligés de retenir tout ça par cœur).
- 1970 : puis, un beau jour, on en est arrivé à créer encore un autre langage qu'on a appelé... le **langage C**. Ce langage, s'il a subi quelques modifications, reste encore un des langages les plus utilisés aujourd'hui. %le plus utilisé, d'après langpop.com cité plus haut !%
- 1983 : un peu plus tard, on a proposé d'ajouter des choses au langage C, de le faire évoluer. Ce nouveau langage, que l'on a appelé « C++ », est entièrement basé sur le C. **Le langage C++** n'est en fait rien d'autre que le langage C avec plusieurs nouveautés. Il s'agit de concepts de programmation poussés comme la programmation orientée objet, le

polymorphisme, les flux... Bref, des choses bien compliquées pour nous pour le moment mais dont nous aurons l'occasion de reparler par la suite !"

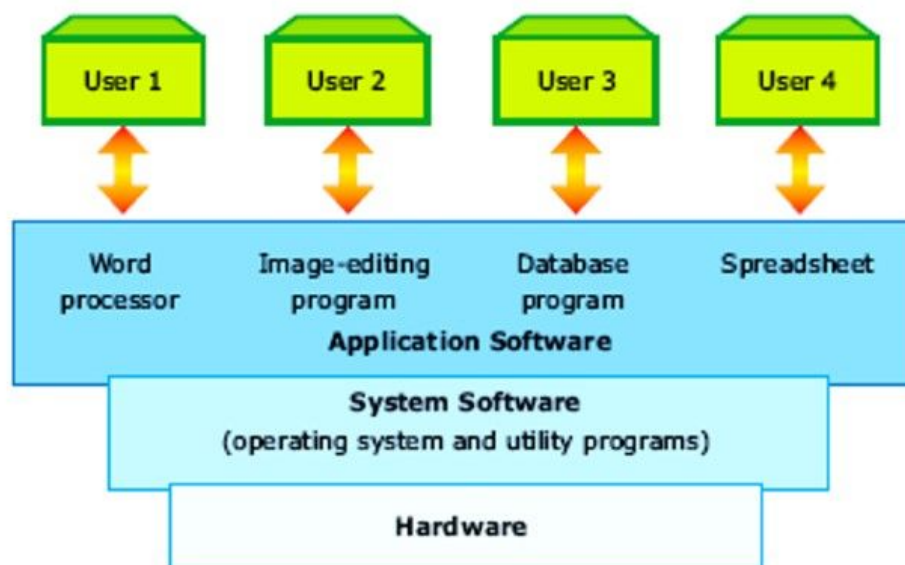
La hiérarchie informatique

La hiérarchie informatique est l'ordre de communication des composants matériel et virtuels d'une machine informatique.

Il pense cela comme un orchestre:

Il y a un chef d'orchestre qui contrôle ses musiciens qui doivent eux aussi contrôler leurs instruments de musique.

Hierarchy of computer software



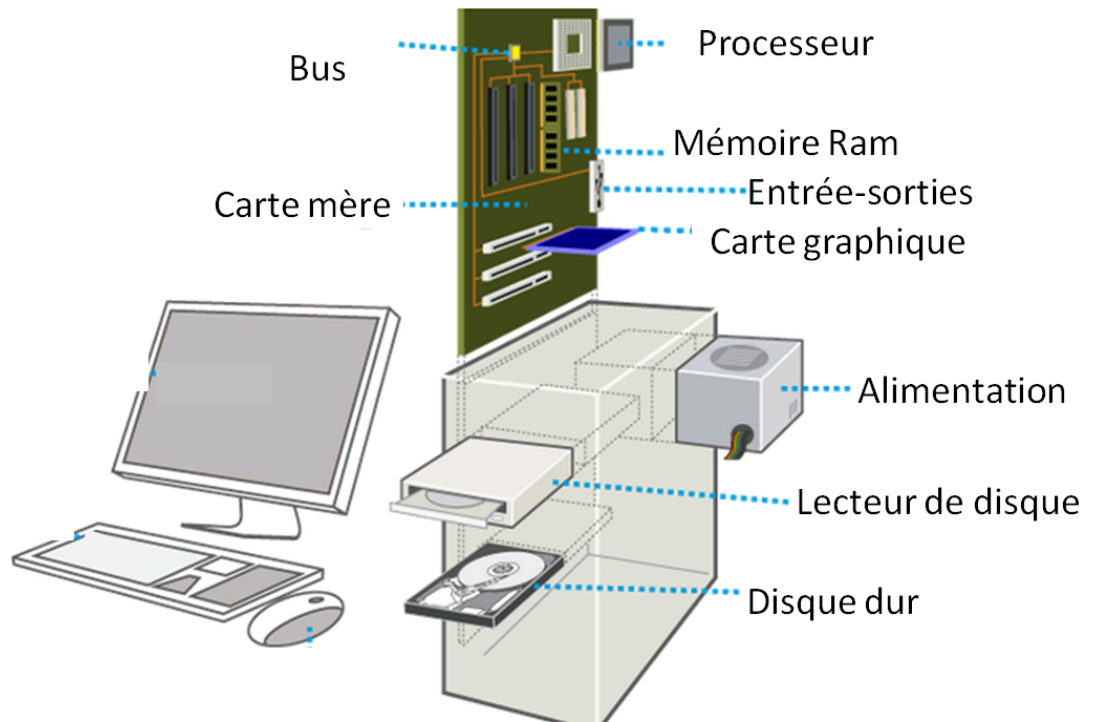
- **Hardware:** Ce qui est maniable à la main - les composants
- **Software:** Ce qui se passe dans les composants. Donc les communications bit qui font le Bios Qui contrôlent le système d'exploitation. (première interaction avec l'utilisateur)
- **Application Software:** Une **application**, un **applicatif** ou encore une **appli**, une **app** est directement utilisé pour réaliser une tâche, ou un ensemble de tâches élémentaires d'un même domaine ou formant un tout.
 - Les applications s'exécutent en utilisant les services du système d'exploitation pour utiliser les ressources matérielles.

C'est la symbiose informatique.

La communication des composants d'un ordinateur:

- **La carte mère** est le chef d'orchestre de l'ordinateur.
- **Les cartes:** Je ne parle pas de celles pour jouer mais de celles qui sont reliées à la carte mère et qui donnent accès aux différents périphérique que, nous, utilisateurs, pouvons utiliser. (La carte graphique pour internet/ethernet, la carte son pour les prises jack,... etc)
- **Les disques durs:** Ou aussi appelé mémoire vive. La mémoire de stockage il en existe plusieurs types (SSD, HDD,... etc)
- **La Ram:** Ou aussi appelé mémoire morte, c'est la place de calcul de l'ordinateur. C'est la que pendant que vous préparer un powerpoint et en même temps lisez un texte la magie s'opère. C'est une mémoire qui dès qu'elle n'est plus utilisé, elle est supprimé.

Les éléments connectés à la carte mère et les constituant de l'ordinateur



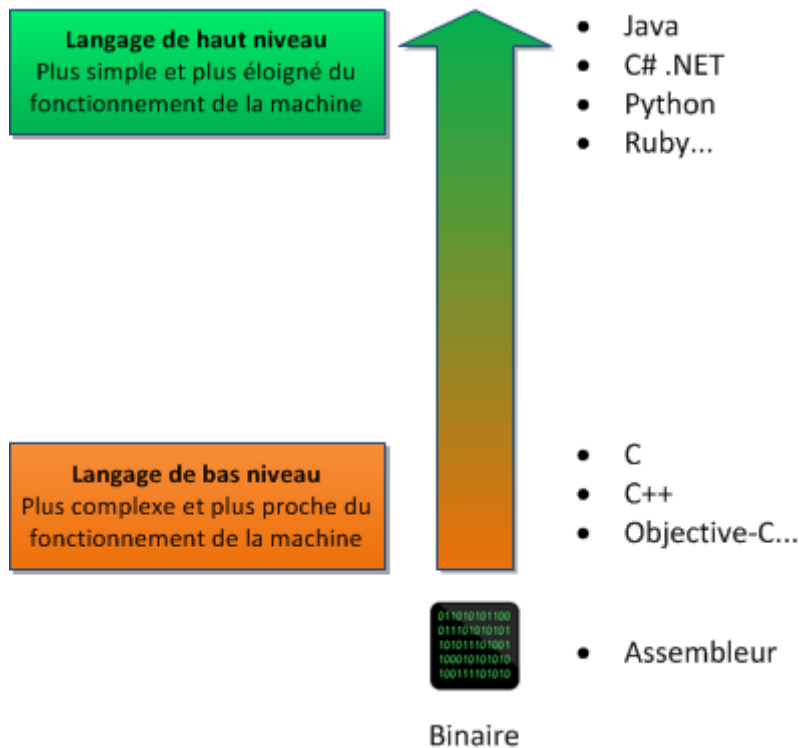
Transition:

Une des hiérarchies de la carte mère est le langage de programmation. Elle se situe juste après le système d'exploitation.



-Mais! Ce système est de quel niveau?

Je ne parle pas de la puissance ou je ne suis pas en train de déterminer si tel code est mieux qu'un autre mais je parle du degré d'interprétation d'un langage de programmation.



Plus un langage est "haut", plus il est éloigné de la machine et donc plus il est éloigné du langage machine.

Le langage machine? Oulà! Non ne vous inquiétez pas, je ne parle pas de la future prise de la bastille de la machine mais je parle du binaire. Le binaire:

"01100010011010010110111001100001011010010111001001100101" Pour ceux qui n'en ont vraiment jamais entendu parlé, il s'agit de la technique de communication des ordinateurs ou même de votre portable.

Mais pourquoi des "1" et des "0"? Car un ordinateur communique avec des transistors. Un transistor est un composant électronique qui va soit pouvoir s'allumer soit va juste s'éteindre. (1 ou 0)

Plus un langage est dit haut et plus on dit de lui qu'il est interprété. En générale on aime dire que les langages les plus interprétés sont plus simples car moins de choses sont à préciser.

Le Langage de programmation et ses outils (IDE):

- Les Niveau d'interprétation avec la machine

- Langage de programmation à aussi sa propre hiérarchie.

***RAPPEL:** Il faut savoir que l'ordinateur ne comprend pas directement la langue anglaise des langages de programmation. Il comprend des entrées ou des sorties. Donc des 1 ou des 0 → donc pour ceux qui connaissent: le langage binaire.*

- C'est quoi un langage de programmation? (ça se parle?)

- Un Langage de programmation est un outil qu'utilise les programmeurs afin qu'un ordinateur exécute des actions de manière automatique et sans avoir besoin de l'intervention d'un humain.
- Il y a énormément de langage de programmation pour toutes les utilisations possibles. Allant du HTML pour la programmation WEB jusqu'au Ti-Basic sur votre calculatrice. Chaque langage a ses avantages mais surtout ses inconvénients. Ainsi il m'est impossible de répondre à LA grande question: "Quel est le meilleur langage de programmation?". Car un langage qui touche à tout est forcément mauvais en tout. Et pour répondre à la question "Par quel langage commencer?" Je répond souvent que cela dépend de comment comment le questionneur se représente la programmation. Et oui! car personne n'est bon en tout, la programmation c'est comme trouver ses études, certains sont meilleurs dans un domaine que dans l'autre.

EXEMPLE: Si minecraft est aussi connu c'est en grande partie pour sa génération de monde impressionnante. Et ceci est grâce à Notch qui est un génie en algorithmes et qui à 14 ans tel un challenge impossible commence la programmation de minecraft.

- Les outils de programmation:

- recette de cuisine de la programmation:
 - Un ordinateur avec un système d'exploitation
 - Un langage de programmation installé sur cet ordinateur.
 - Un éditeur de texte
 - Un compilateur
 - Un exécuteur

- Un IDE?

- Integrated Development Environment -> ou EDI en français
 - Un Environnement de développement

LE DEBUT!! Là on va commencer la programmation!

Notre IDE pour ces prochains premiers cours de programmation sera minecraft.

Nous allons nous amuser avec les turtles. Des tortues?!

Une turtle est un block dans minecraft modé computer-craft auquel nous allons pouvoir donner des instructions exactement comme à un ordinateur.

Le cours se déroulera de la sorte:

En chaque début de cours, je vous donnerais un .txt avec dessus une clé d'activation minecraft pour 100 minutes.

Les prérequis de ce cours sont disponibles dans le fichier mega qui suit:

Je tiens à préciser qu'une grosse configuration pc de gamer n'est évidemment pas nécessaire. Ce cours a été pensé pour les hp fournis par Grand-Est donc 4 GB RAM cadencé à 1,10 GHz avec un rafraîchissement de 30 Ghz.

Dedans vous y trouverez:

Des .txt avec des numéros. Prenez le numéro qui vous est attiré.

- un lien de téléchargement du launcher minecraft. (le lancer est gratuit).
- un lien de téléchargement du mod Computercraft et du mod Optifine.
- un tutoriel d'installation.
- Et enfin votre launcher ID valable pour 100 minutes (Un nouveau vous sera attiré en chaque début de cours)

VOILA! on peut commencer... mais longue est la route jeune padawan.

La turtle et ses mouvements de base XYZ:

First things first...

RAPPEL: La turtle est un bloc qui se comporte tel un véritable ordinateur. On peut lui assigner des tâches à faire.

Pour réussir cette partie du cours et donc de contrôler nos turtle, il faut penser en ordre. Vous n'allez pas demander poliment à votre turtle d'avancer! Vous allez lui ordonner!

Le système de déplacement des turtle fonctionne comme un repère orthonormé tridimensionnel.

Commencez par poser votre turtle. ensuite il faut faire un clic droit dessus afin de voir le terminal de commande apparaître. Un terminal de commande est une fenêtre disponible sur tout les systeme d'exploitation. Cette fenêtre permet à un utilisateur de pouvoir avoir accès au plus profond de la machine avec des commandes.

Ici c'est exactement la même chose: la fenêtre noire qui c'est ouverte est notre lieu de coordination des différentes tâches de la turtle. Nous allons commencer par créer un endroit dans lequel nous allons pouvoir inscrire des tâches. Oui, nous allons créer un fichier.

Pour créer un fichier c'est comme sur linux, il faut appeler nano puis le nom du programme et .lua:

```
> nano test.lua
```

pourquoi le point lua? pour indiquer en quel langage nous allons programmer. Ici nous n'avons de toute façon que le lua.

instant wikipédia: <https://fr.wikipedia.org/wiki/Lua#Usages>

“L'interpréteur Lua est écrit en langage C ANSI strict, et de ce fait est compilable sur une grande variété de systèmes. Il est également très compact, la version 5.0.2 n'occupant que 95 ko à 185 ko selon le compilateur utilisé et le système cible. Il est souvent utilisé dans des systèmes embarqués tels qu'OpenWrt4 où cette compacité est très appréciée. Il profite de la compatibilité que possède le langage C avec un grand nombre de langages pour s'intégrer facilement dans la plupart des projets. Il est particulièrement apprécié pour l'embarqué, le développement réseau et les jeux vidéo.”

Maintenant nous avons notre éditeur de texte.

Pour donner un ordre à la turtle il faut d'abord appeler son nom, puis mettre “.action”. En programmation en général, les points sont utilisés pour accéder à une fonction issue d'un autre fichier. Et c'est ce qui se passe ici. Dans le fichier turtle nous prenons “action”.

essayez: (Je vous laisse faire joujou quelques minutes)

Mémento des “actions”:

-forward()	-turnRight()
-back()	-turnLeft()
-up()	
-down()	

LES types de variables:

Loops:

Une course de qui va le plus vite pour faire avancer sa turtle d'un point “a” à un point “b”. (10 blocs)

Je fais une boucle et arrive rapidement

→ Moyen d'automatiser les mêmes instructions

Les conditions:

IF - ELSE - ELSEIF

WHILE

FOR