

**.ordinateur**

# ordinateur

Un survol en image de l'avènement de l'ordinateur,  
ainsi que des informations sur son fonctionnement.

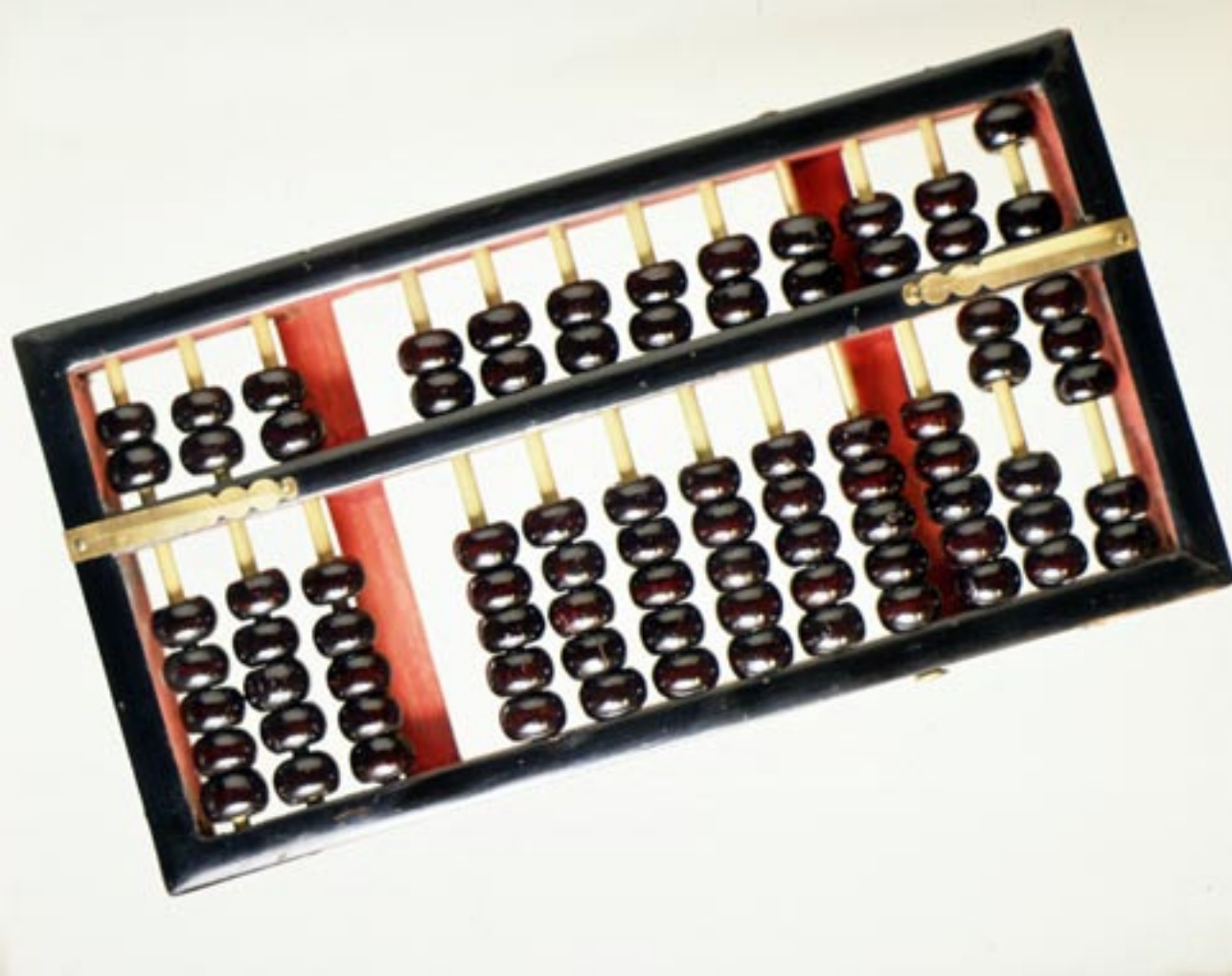
Erg (École de Recherche Graphique) - Bruxelles.  
Arts numériques 1e, 2e, 3e & 4e année.  
Professeur: Marc Wathieu.

*Mise à jour: 04 février 2007.*

*Ce livret PDF a été conçu comme un diaporama destiné à être projeté et commenté.  
Pour un affichage optimisé, je vous recommande une résolution 1024 X 768,  
une visualisation avec Acrobat Reader  
et le raccourci ctrl+I (Windows) ou pomme+I (Mac OSX).*

*Télécharger ici Acrobat Reader.*

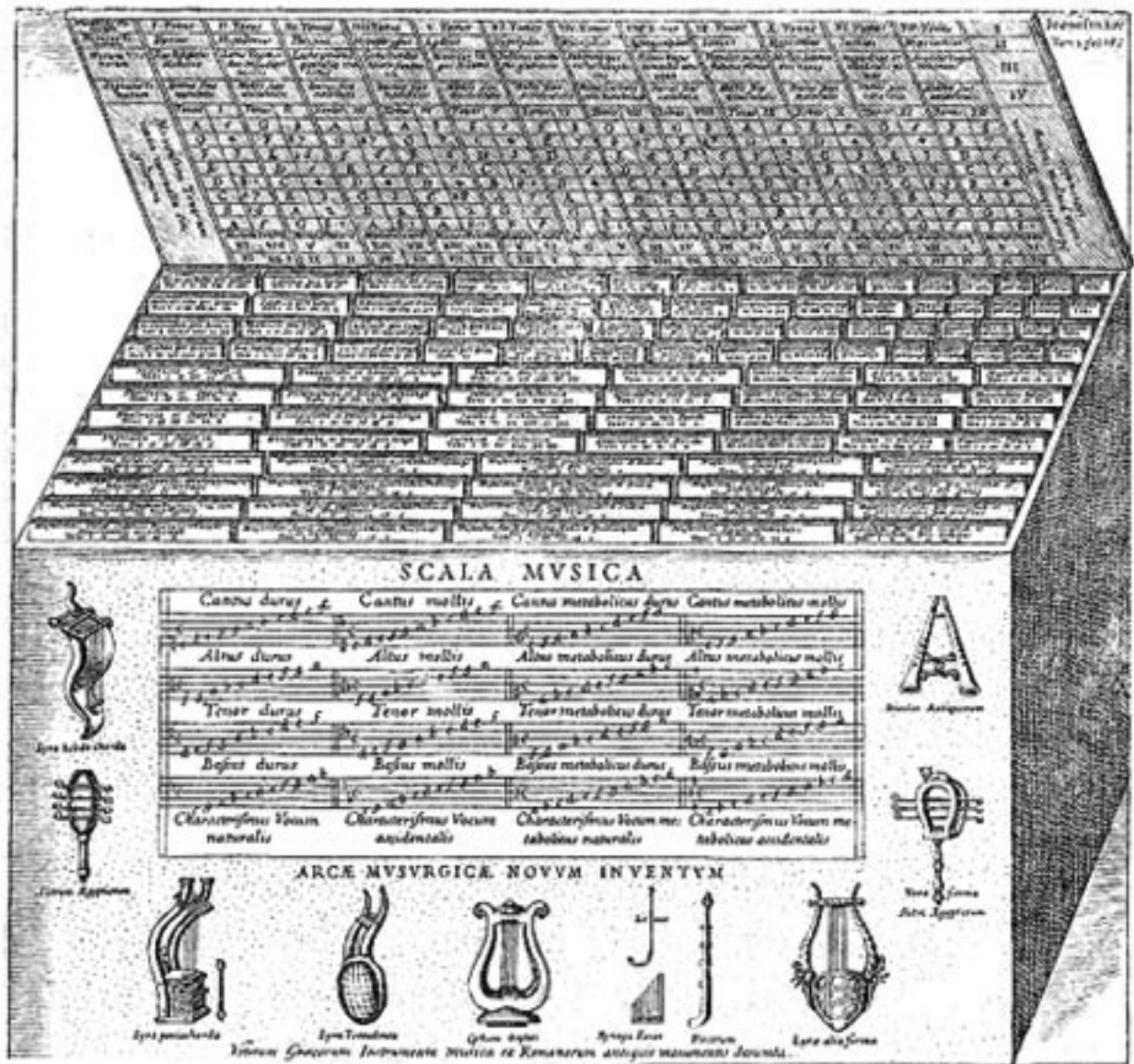
.survol historique



XIIe siècle : Le boulier chinois (suan pan).

Chaque tige correspond, de droite à gauche, respectivement aux unités, dizaines, centaines, milliers etc. En 1945, Kiyoshi Matsuzaki, un comptable japonais muni d'un soroban (boulier japonais) remporta un match contre Nathan Woods, un opérateur de calculatrice électrique, par un score de 4 à 1.

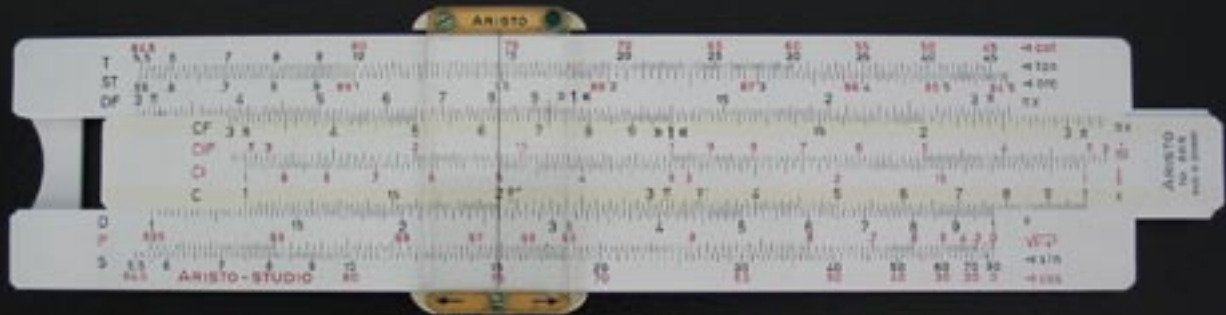
<http://en.wikipedia.org/wiki/Abacus>



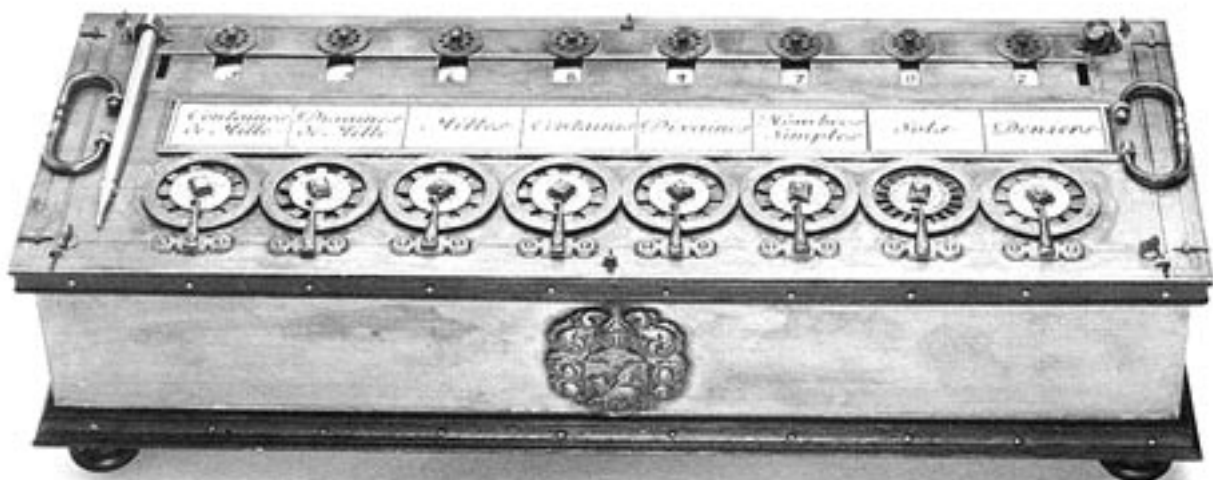
Athanasius Kircher (1601-1680) a inventé un système destiné à créer des partitions musicales, ce qui fait de lui le père de la musique algorithmique générative. Il est également l'auteur de concepts d'instruments de musique automatisés.

[http://en.wikipedia.org/wiki/Athanasius\\_Kircher](http://en.wikipedia.org/wiki/Athanasius_Kircher)  
<http://kircher.stanford.edu/>



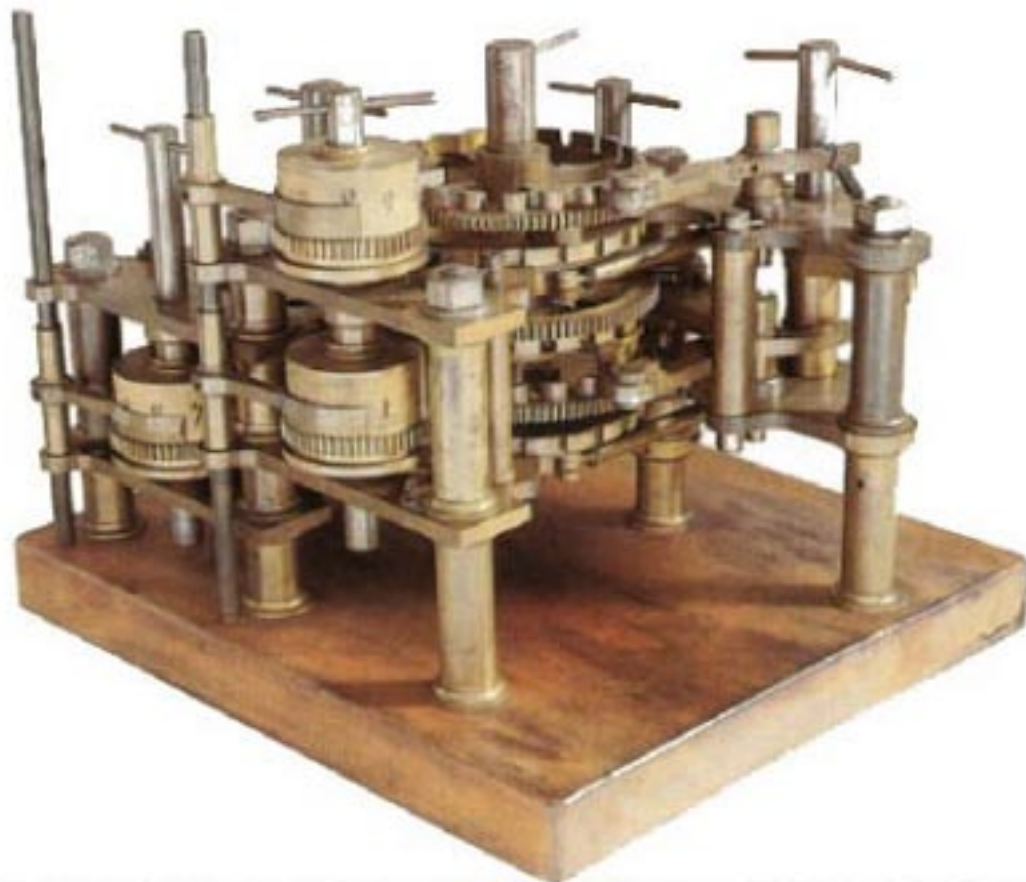


1630 : William Oughtred invente une règle à calculer circulaire (mechanical analog computer : to compute, calculer), ancêtre de l'ordinateur. Ces règles à calculer étaient encore utilisées dans les années 60, notamment à bord des missions Appolo.



1642 : Blaise Pascal invente la «Pascaline»,  
une des premières calculatrices mécaniques.

[http://fr.wikipedia.org/wiki/Blaise\\_Pascal](http://fr.wikipedia.org/wiki/Blaise_Pascal)



1833 : Charles Babbage imagine et tente de réaliser une «machine à différences» puis une «machine analytique», contenant les concepts de ce que sera l'ordinateur moderne.

[http://en.wikipedia.org/wiki/Charles\\_Babbage](http://en.wikipedia.org/wiki/Charles_Babbage)





1896 : Herman Hollerith invente  
une «machine à statistiques à cartes perforées».

[http://en.wikipedia.org/wiki/Herman\\_Hollerith](http://en.wikipedia.org/wiki/Herman_Hollerith)

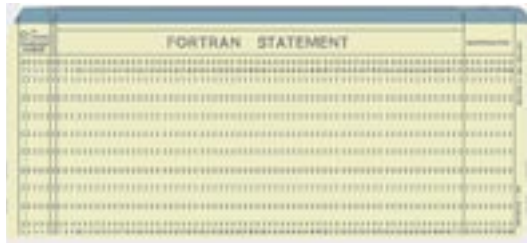
Le système des cartes perforées inventé par Herman Hollerith était encore utilisé dans les années 70.



Le programmeur remplissait un bordereau de saisie.



Une opératrice encodait les données pour produire une carte perforée.

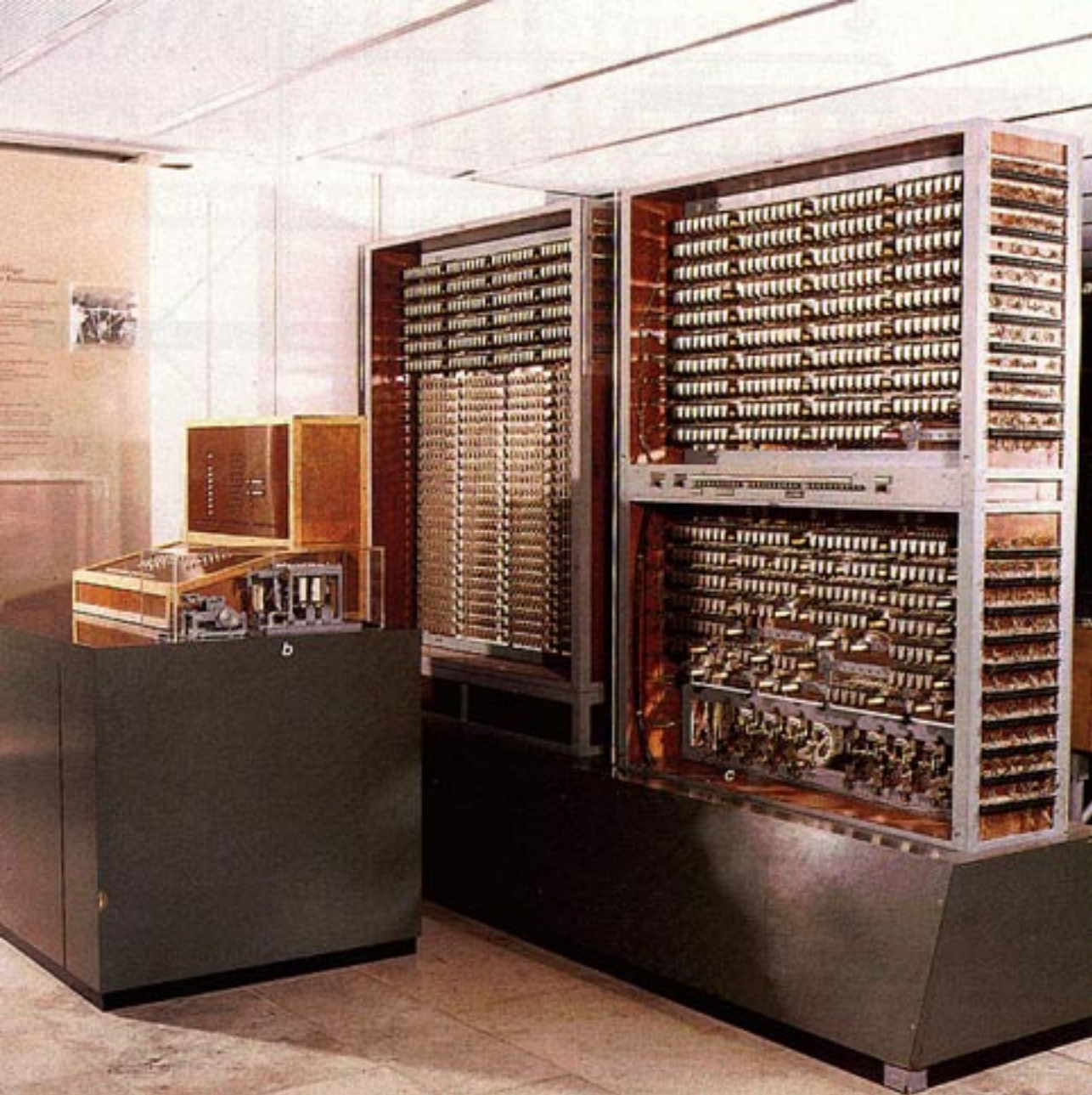


Les cartes étaient lues par une unité périphérique et le programme pouvait être exécuté.



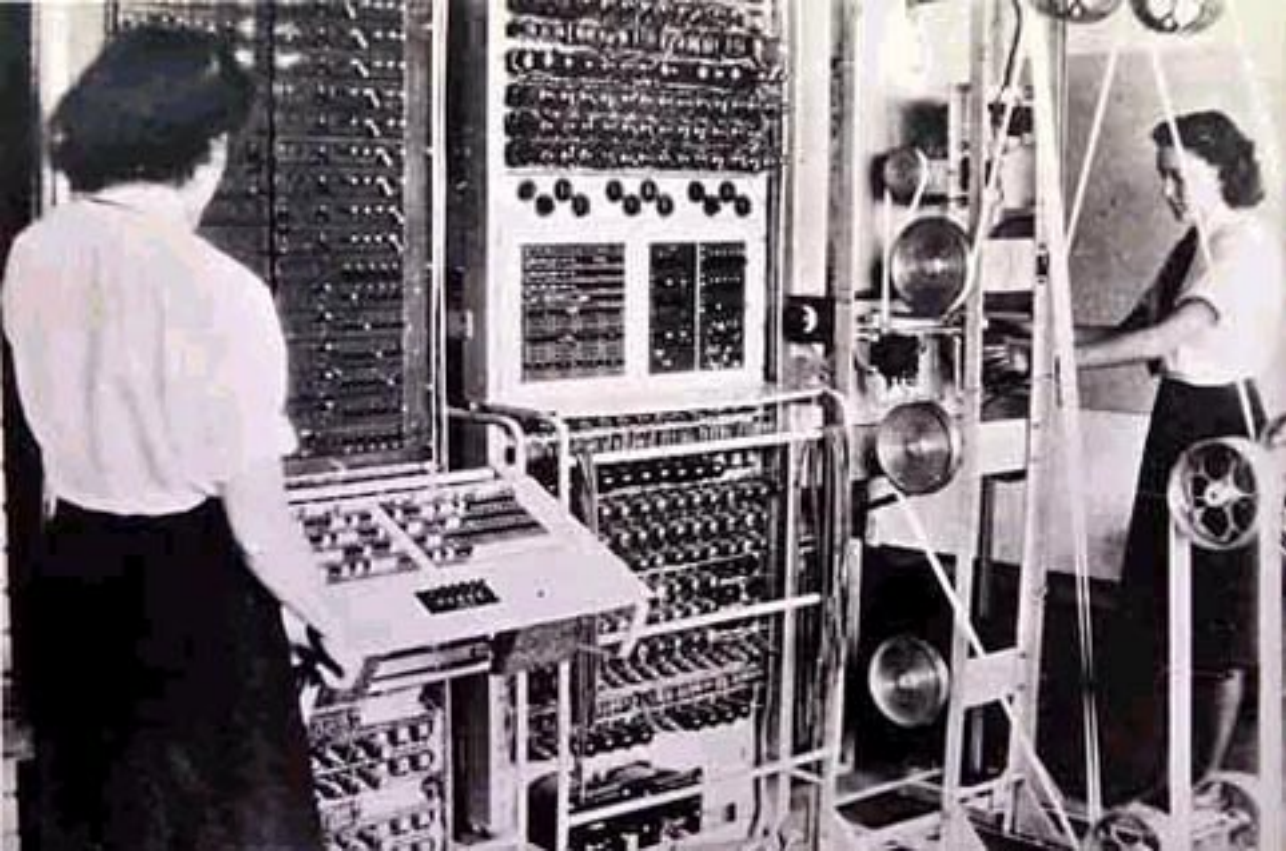
1936 : Alan Turing, mathématicien et logicien anglais, est considéré comme un des pères fondateurs de l'informatique moderne. Il formalise le concept d'algorithme en concevant un système logique programmable, appelé depuis «Machine de Turing».

[http://en.wikipedia.org/wiki/Alan\\_Turing](http://en.wikipedia.org/wiki/Alan_Turing)



1941 : Konrad Zuse conçoit le Z3,  
premier ordinateur programmable mécaniquement  
(ici une reconstruction vers 1960).

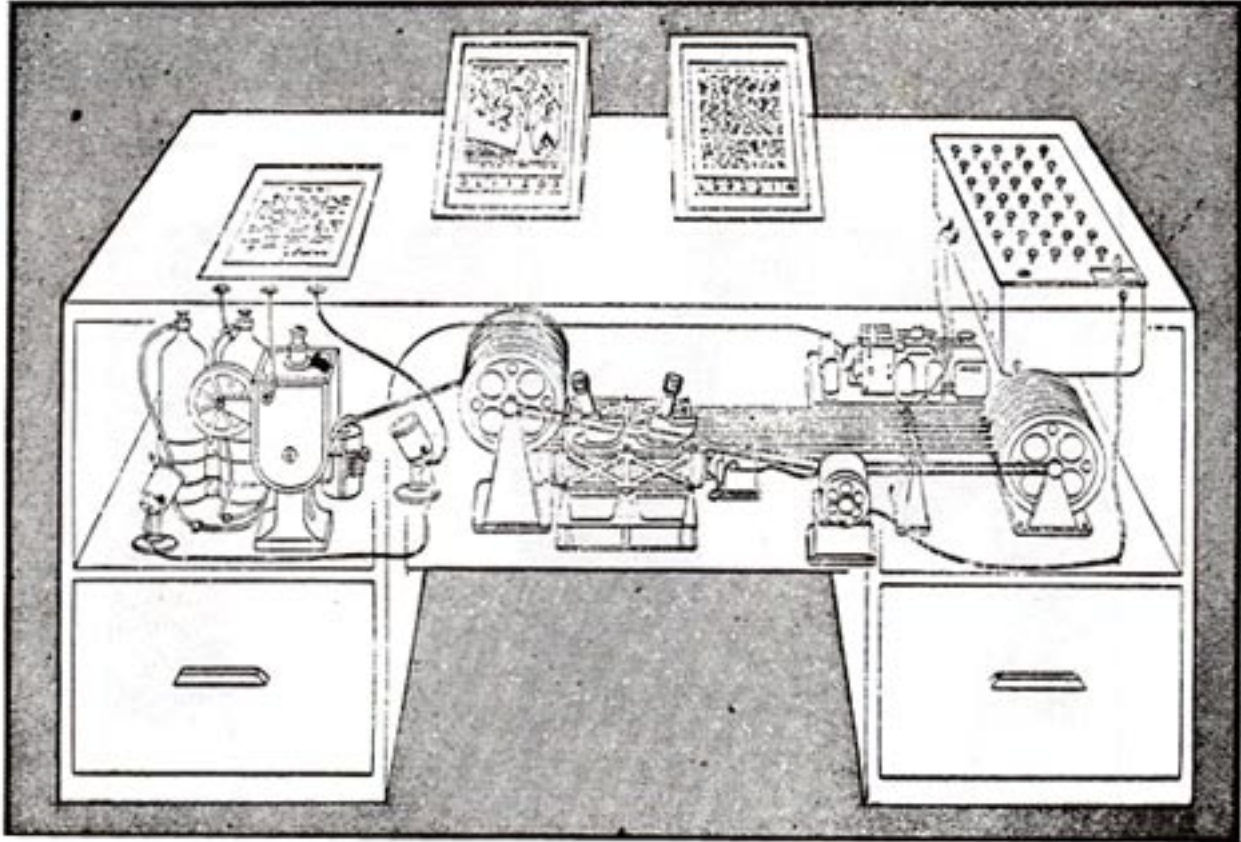




1944 : Tommy Flowers conçoit le Colossus,  
premier ordinateur électronique programmable.

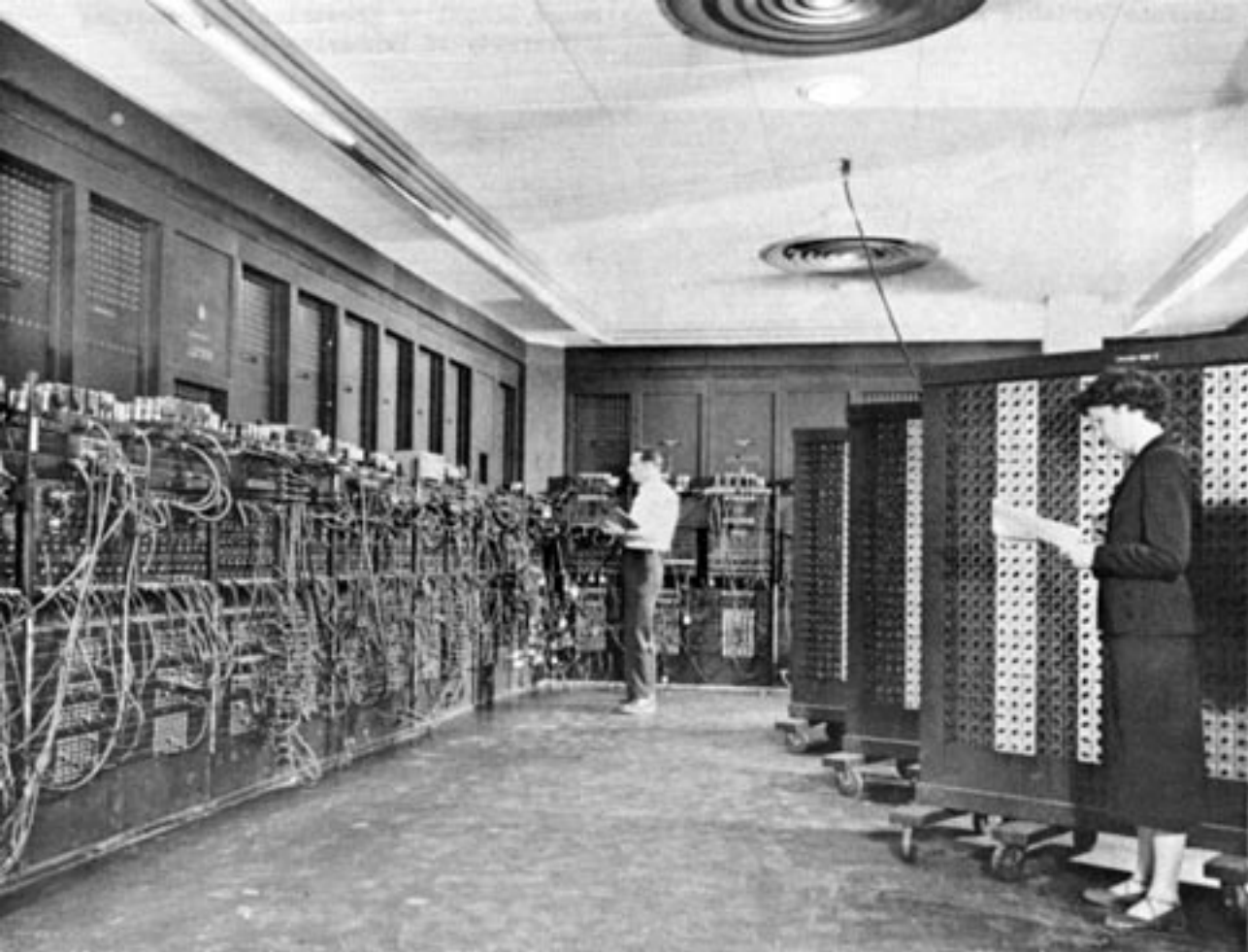
[http://en.wikipedia.org/wiki/Colossus\\_computer](http://en.wikipedia.org/wiki/Colossus_computer)





1945 : Vannevar Bush imagine le Memex (Memory extender), ordinateur analogique théorique et visionnaire.

<http://en.wikipedia.org/wiki/Memex>



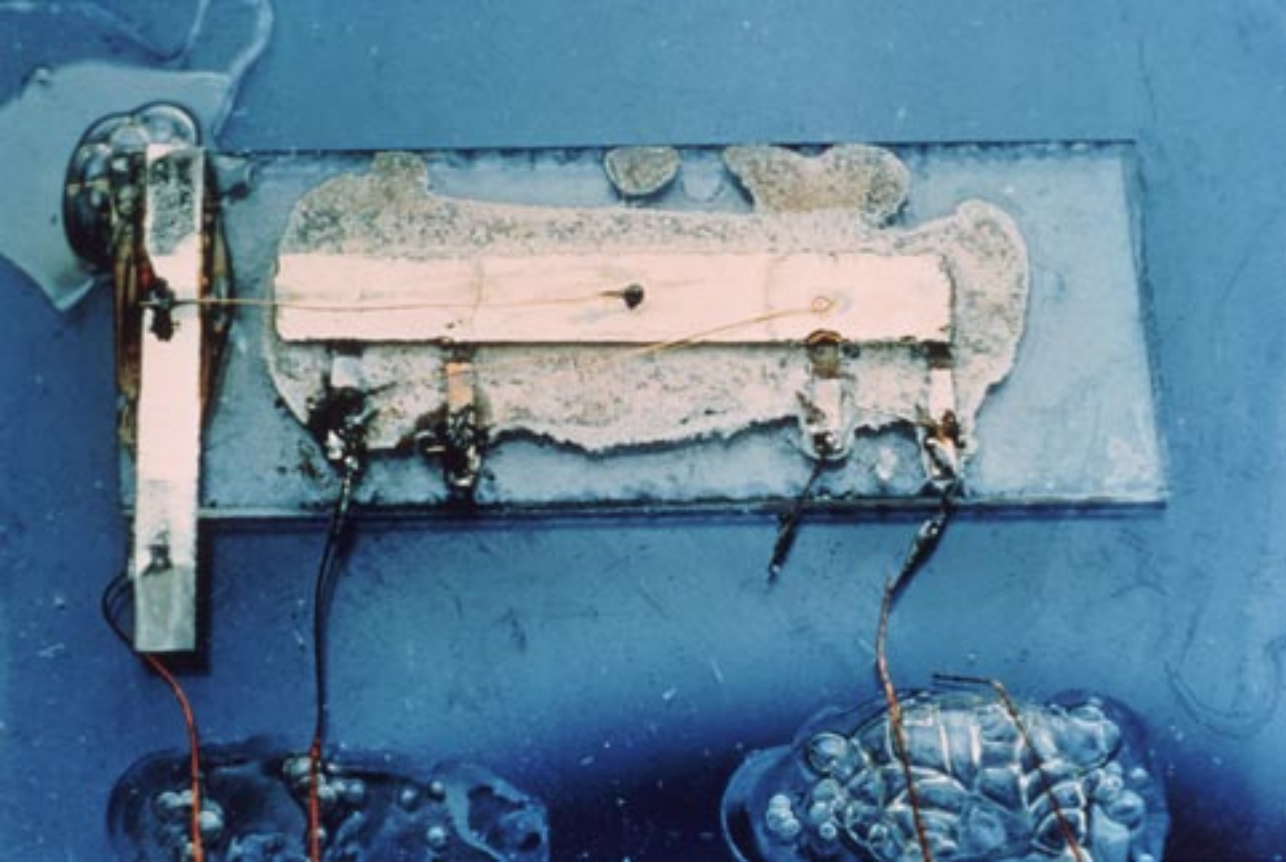
1946 : John William Mauchly & J. Presper Eckert conçoivent le ENIAC  
(Electronic Numerical Integrator And Computer),  
premier ordinateur entièrement électronique.

<http://fr.wikipedia.org/wiki/ENIAC>



1951 : John William Mauchly & J. Presper Eckert  
conçoivent le UNIVAC I (UNIVersal Automatic Computer I),  
premier ordinateur commercialisé.

[http://fr.wikipedia.org/wiki/UNIVAC\\_I](http://fr.wikipedia.org/wiki/UNIVAC_I)



1958 : Jack Kilby (Texas Instruments) invente le circuit intégré.

[http://en.wikipedia.org/wiki/Texas\\_Instruments](http://en.wikipedia.org/wiki/Texas_Instruments)

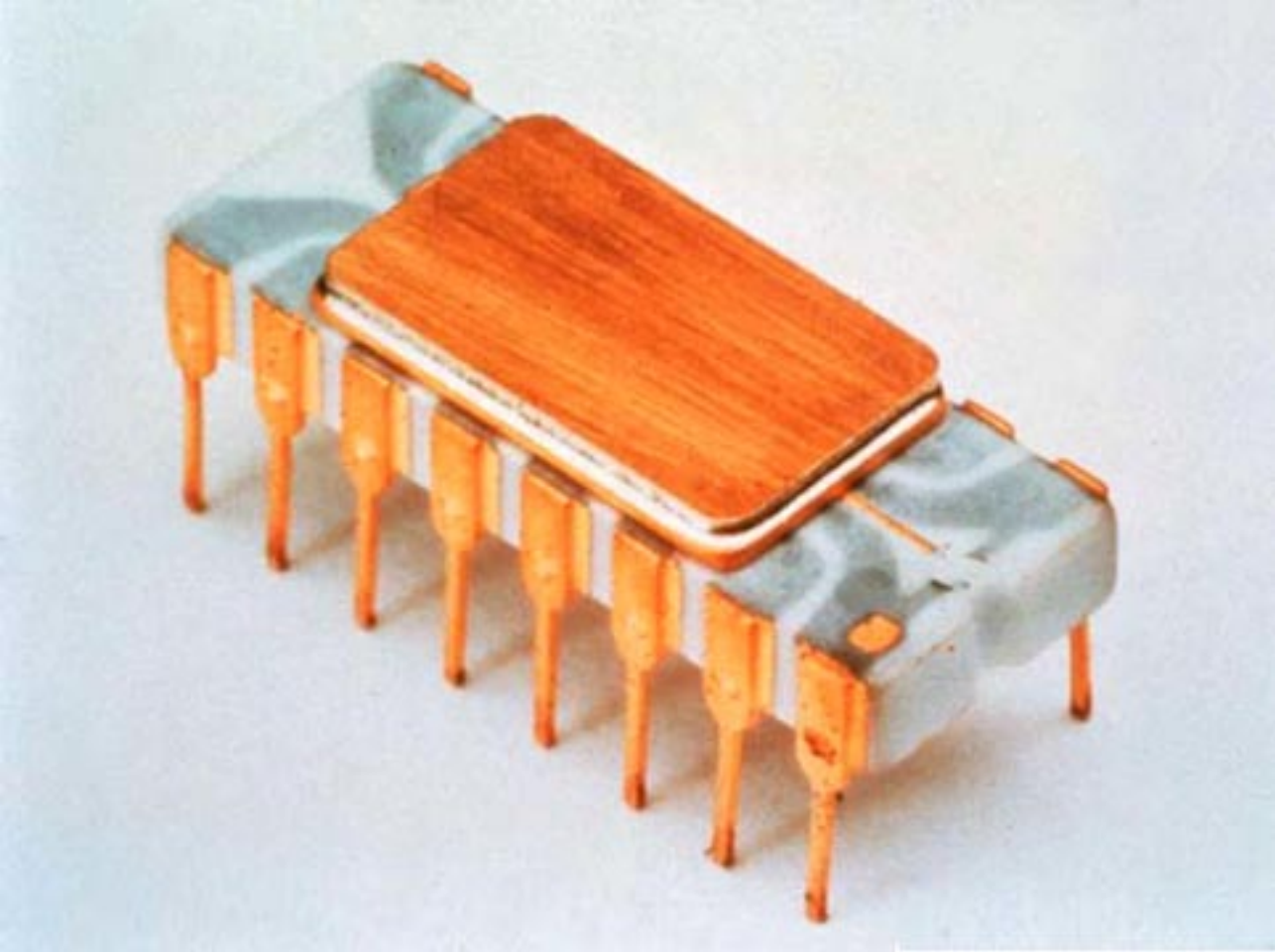




1967 : Texas Instruments dépose le brevet  
d'un premier prototype de calculatrice.

<http://fr.wikipedia.org/wiki/Calculatrice>





1971 : Intel 4004, conçu par l'ingénieur d'Intel Marcian Hoff, considéré comme le premier microprocesseur du monde.

<http://fr.wikipedia.org/wiki/Intel>



Voici à quoi pouvait ressembler un ordinateur au début des années 70 (ici, un système IBM 360). Un ordinateur utilisable par le grand public paraissait complètement absurde à l'époque.

<http://en.wikipedia.org/wiki/System/360>



Début 70 : une communauté de hackers (dont John Draper et le duo Steve Wozniak/Steve Jobs, fondateurs de Apple) développe la «Blue Box», un petit boîtier capable de simuler les tonalités de numérotation de la compagnie BELL, permettant ainsi d'effectuer des appels gratuits. Joe Engressia, autre célèbre «phone phreak», aveugle de naissance, piratait le système BELL en «sifflant» les fréquences précises correspondant aux numéros.

[http://en.wikipedia.org/wiki/Blue\\_box](http://en.wikipedia.org/wiki/Blue_box)



1971 : le Datapoint 2200, terminal (périphérique de commande) connectable à un mainframe (super-ordinateur), possède déjà les caractéristiques d'un ordinateur personnel.

[http://en.wikipedia.org/wiki/Datapoint\\_2200](http://en.wikipedia.org/wiki/Datapoint_2200)



Computers are mostly

used against people instead of for people  
used to control people instead of to free them

time to change all that -

we need a ...



1972 : un groupe d'activistes de Berkeley (Californie) publie «The People's Computers Newsletters», pour une démocratisation des ordinateurs.

<http://www.digibarn.com/collections/newsletters/peoples-computer/index.html>





1972-1973 : Micral, premier ordinateur autonome prêt à l'emploi, conçu autour d'un microprocesseur Intel 8008.

<http://fr.wikipedia.org/wiki/Micral>

<http://en.wikipedia.org/wiki/Micral>



1975 : le Altair 8800. Programmable à l'aide d'interrupteurs, il utilisait des lumières clignotantes pour l'affichage. Encore loin de ressembler à un ordinateur personnel tel que le futur Apple II, le Altair 8800 annonce la révolution de l'informatique personnelle.

[http://en.wikipedia.org/wiki/Altair\\_8800](http://en.wikipedia.org/wiki/Altair_8800)



1976 : Steve Jobs et Steve Wozniak : Apple I,  
premier ordinateur individuel à être conçu  
pour être combiné à un clavier et à un moniteur.

[http://fr.wikipedia.org/wiki/Apple\\_I](http://fr.wikipedia.org/wiki/Apple_I)



1977 : Steve Jobs et Steve Wozniak : Apple II.

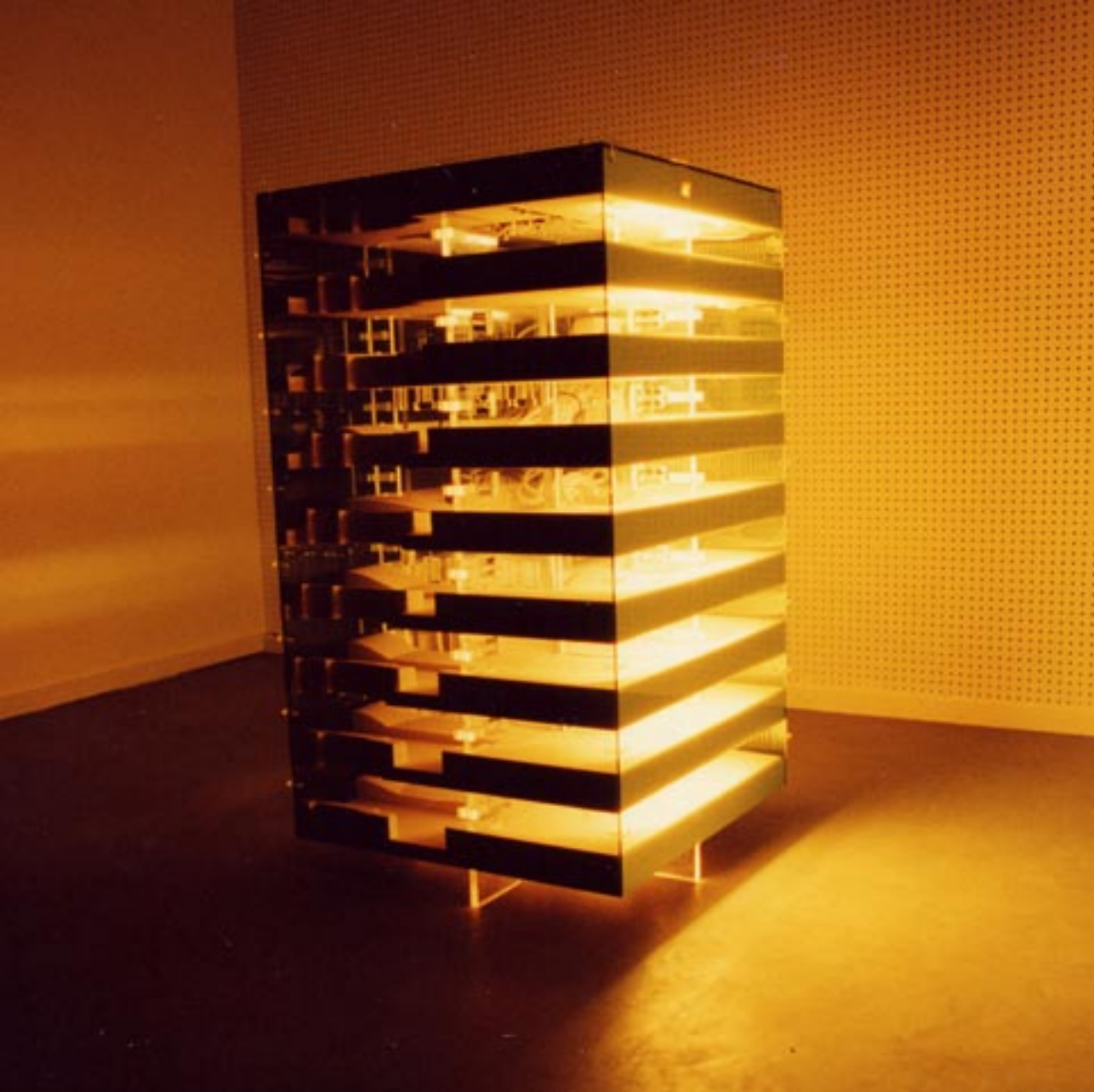
[http://fr.wikipedia.org/wiki/Apple\\_Computer](http://fr.wikipedia.org/wiki/Apple_Computer)



1984 : Apple Macintosh. Il était fourni avec trois applications révolutionnaires tirant partie de l'interface graphique : MacPaint, MacWrite et MacDraw. Ces trois logiciels inspireront tous les grands logiciels actuels, respectivement de retouche d'image (Photoshop), de traitements de texte (Word) et de dessin vectoriel (Illustrator).

[http://en.wikipedia.org/wiki/Macintosh\\_128K](http://en.wikipedia.org/wiki/Macintosh_128K)

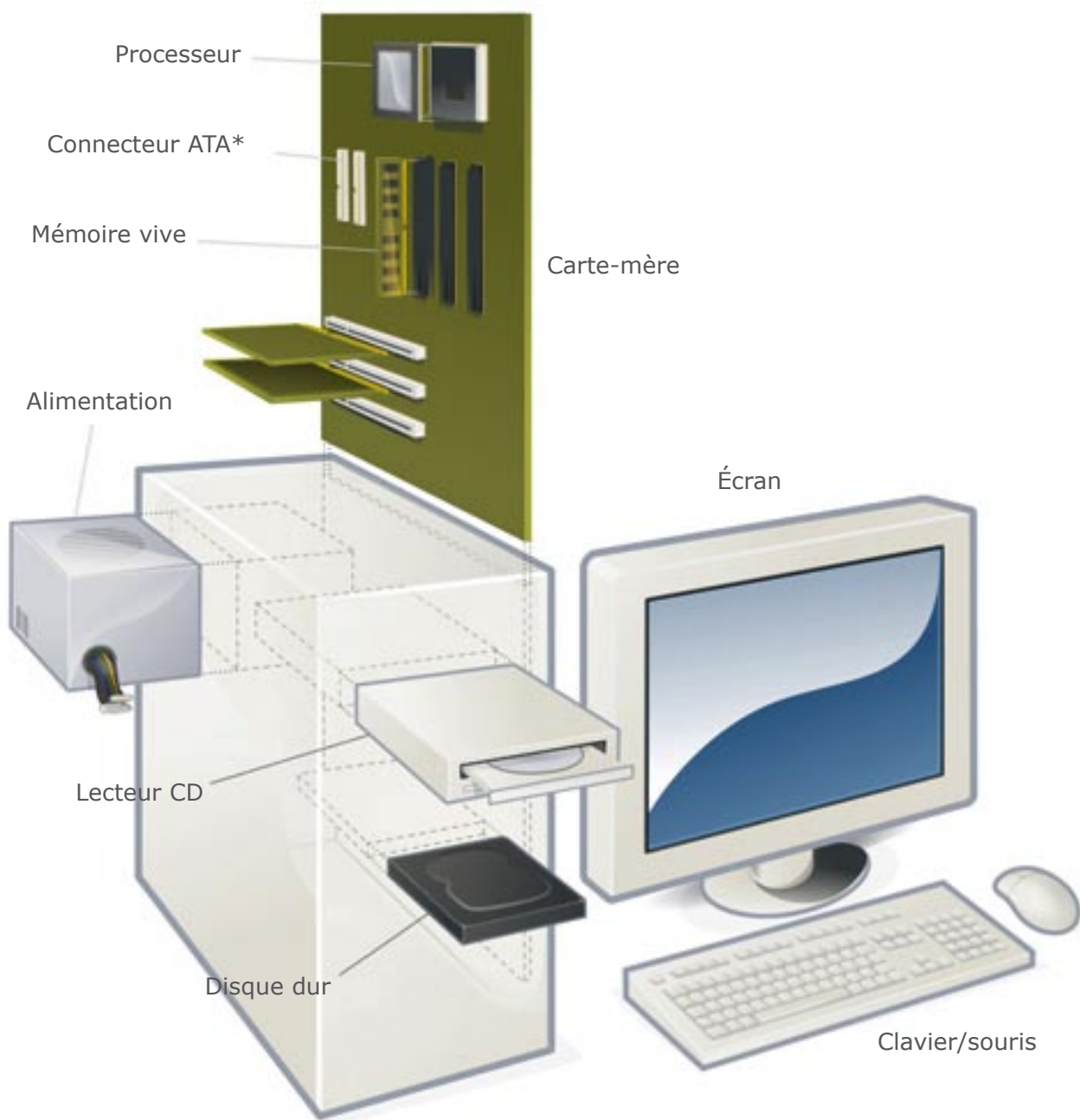




Un exemple de machine conçue par un artiste :  
Oval Process Terminal (container, écran LCD, Mac G4, logiciel - 2004)  
conçu par Markus Popp comme un environnement  
et une méthode de production sonore.

<http://www.medienkunstnetz.de/works/oval-process/>

.anatomie

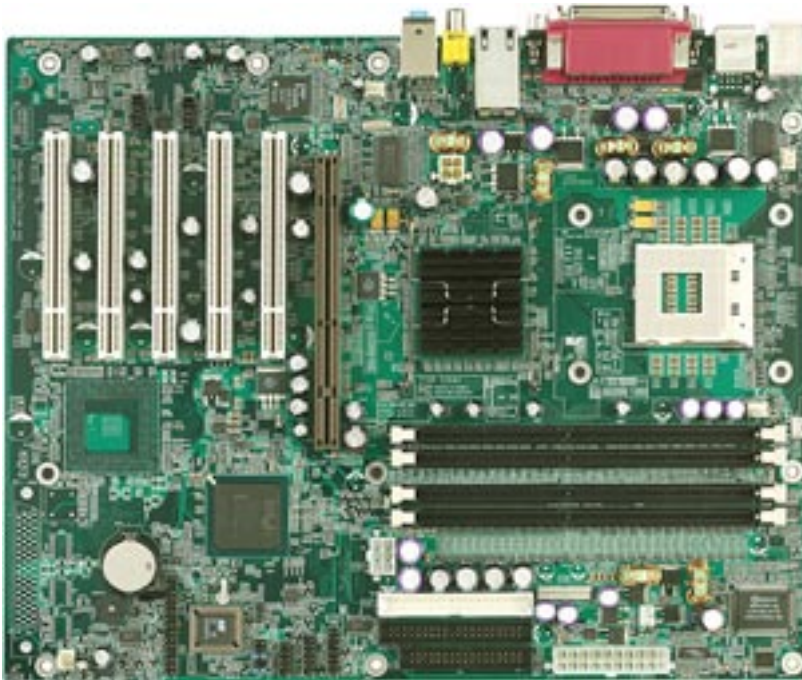


\* Le standard ATA (Advanced Technology Attachment) est une interface permettant la connexion de périphériques de stockage sur les ordis.

Schématiquement, un ordinateur est composé de trois parties distinctes:

- 1- la **Mémoire Centrale**.
- 2- l'**Unité Centrale**.
- 3- les **Périphériques**.

La carte-mère permet la connexion des éléments essentiels.



← Processeur  
(unité centrale)

← Mémoire

En savoir plus :

<http://www.commentcamarche.net/pc/carte-mere.php3>



1- la **Mémoire Centrale** permet de mémoriser les programmes pendant le temps nécessaire à leur exécution. On y trouve également les informations temporaires manipulées par ces programmes: données après leur introduction, résultats avant leur communication à l'extérieur, informations intermédiaires apparaissant pendant le déroulement d'un programme.

La mémoire centrale correspond à ce que l'on appelle la **Mémoire Vive** ou **RAM** (Random Access Memory, mémoire à accès direct).

Contrairement au stockage de données sur une mémoire de masse telle que le disque dur, la mémoire vive est volatile, c'est-à-dire qu'elle permet uniquement de stocker des données tant qu'elle est alimentée électriquement. Ainsi, à chaque fois que l'ordinateur est éteint, toutes les données présentes en mémoire sont irrémédiablement effacées.

Plus précisément, on peut distinguer :

- les **Registres** : petites mémoires rapides de 8, 16, 32 ou 64 bits. Suivant le type de processeur, le nombre global de registres peut varier d'une dizaine à plusieurs centaines.
- la **Mémoire Cache** (mémoire tampon) : mémoire rapide permettant de réduire les délais d'attente des informations stockées en mémoire vive.



En savoir plus :

<http://www.commentcamarche.net/pc/memoire.php3>

2- L'**Unité Centrale** (ou processeur, ou CPU - Central Processing Unit, ou UCT - Unité Centrale de Traitement) est la partie active de l'ordinateur. Elle est chargée de prélever une à une chaque instruction de programme située en mémoire centrale et de l'exécuter.

Plus précisément, on peut distinguer deux sortes d'instructions:

- celles qui agissent sur des informations situées en mémoire centrale. Ce sont elles qui permettent de réaliser le traitement escompté.
- celles qui assurent la communication ou l'archivage d'information. Elles réalisent en fait un échange d'information entre la mémoire centrale et d'autres appareils nommés **Périphériques**.



En savoir plus :

<http://fr.wikipedia.org/wiki/Processeur>

<http://www.commentcamarche.net/pc/processeur.php3>

3- Les **Périphériques** désignent tous les appareils susceptibles d'échanger des informations avec la mémoire centrale.

Ils sont de deux sortes :

- ceux qui assurent la communication entre l'homme et l'ordinateur (claviers, écrans, souris...)
- ceux qui assurent l'archivage d'information (disques durs, sticks USB...). Ces derniers ont un rôle de mémorisation d'information au même titre que la mémoire centrale, dont ils constituent ainsi une sorte de prolongement.



En savoir plus :

<http://www.commentcamarche.net/pc/peripherique.php3>

.fonctionnement



Lorsqu'un ordinateur **exécute** un programme,  
son travail consiste en grande partie à **gérer la mémoire** :

- soit pour y **lire** une instruction.
- soit pour y **stocker** une information.

En ce sens, l'ordinateur peut être perçu comme un **robot** qui agit en fonction d'**ordres** qui lui sont fournis.

Ces actions sont en nombre limité :

- **Déposer** ou **lire** une information dans une case mémoire.
- **Exécuter** des opérations simples.
- **Comparer** des valeurs.
- **Communiquer** une information élémentaire.
- **Coder** l'information.

## Déposer ou lire une information dans une case mémoire.

La mémoire est formée d'éléments, ou **cases-mémoire**, qui possèdent chacune un numéro (une **adresse**).

Chaque case-mémoire est en quelque sorte une **boîte aux lettres** pouvant contenir une information (une lettre).

Pour y déposer cette information, l'ordinateur (le facteur) doit connaître l'adresse de la boîte.





Lorsque le robot place une **information** dans une **case mémoire**, il en mémorise l'**adresse** afin de retrouver l'information en temps nécessaire.

Le robot sait déposer une information dans une case, mais il ne sait pas la retirer (au sens de prendre un courrier déposé dans une boîte aux lettres).

Lorsque le robot prend l'information déposée dans une case mémoire, il ne fait que la lire. En aucun cas il ne la retire ni ne l'efface. L'information lue reste toujours dans la case mémoire.



Remarque :

Pour effacer une information d'une case mémoire, il est nécessaire de placer une nouvelle information dans cette même case. Ainsi, la nouvelle donnée remplace l'ancienne, et l'information précédente est détruite.



Jacques Tati : Playtime (1967).  
Une unité centrale de calcul ?  
<http://www.tativille.com/>

## Exécuter des opérations simples.

La machine **lit** et **exécute** les opérations dans l'ordre où elles lui sont fournies.

Pour faire une addition ou une soustraction, il va chercher les valeurs à traiter dans les **cases-mémoire** appropriées (stockées, par exemple, aux adresses **a** et **b**) et réalise ensuite l'opération demandée.

Il enregistre alors le résultat de l'opération dans une case d'adresse **c**.

De telles opérations sont décrites à l'aide d'ordres ou d'**instructions**.

## Comparer des valeurs.

La machine est capable de **comparer** deux valeurs entre elles pour déterminer si l'une d'entre elles est plus grande, plus petite, égale ou différente de l'autre valeur.

Grâce à la comparaison, la machine peut tester une condition et exécuter un ordre plutôt qu'un autre, en fonction du résultat du test.

La réalisation d'une comparaison ou d'un test empêche le robot d'exécuter les instructions dans leur ordre d'apparition. En effet, suivant le résultat, il doit bifurquer et rompre l'ordre de la marche à suivre, en sautant une ou plusieurs instructions. C'est pourquoi il existe des **instructions** particulières de type **branchements**.

«**if...else**» est, par exemple, une instruction bien connue qui fixe les conditions d'une action.

En sautant des ordres, le robot peut aussi revenir à un ensemble d'opérations afin de les répéter en **boucle**.

En savoir plus :

<http://www.commentcamarche.net/logic/intro.php3>

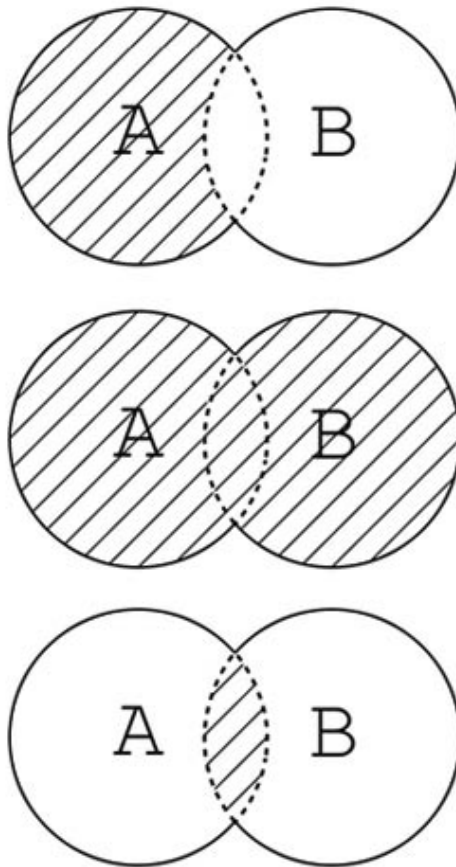


Pour exécuter une commande **if( ) ... else( )**, l'ordinateur doit vérifier si les conditions sont réunies pour éventuellement bifurquer vers une instruction optionnelle. Après comparaison, il constate que les conditions sont «vraies» (true) ou «fausses» (false).

**True** ou **False** sont des constats réalisés grâce à des fonctions logiques, appelées opérateurs booléens (du nom de Georges Boole) : «égal à», «n'est pas égal à», «supérieur à», etc...

L'algèbre de Boole permet de modéliser des raisonnements logiques, en exprimant un «état» en fonction de conditions.

ET (AND), OU (OR) et NON (NOT) sont les 3 fonctions de base.



Algèbre de Boole : fonctions logiques OU, ET, NON.

[http://fr.wikipedia.org/wiki/Algèbre\\_de\\_Boole\\_%28logique%29](http://fr.wikipedia.org/wiki/Algèbre_de_Boole_%28logique%29)

## Communiquer une information élémentaire.

Un programme est essentiellement un outil qui traite l'information.

L'utilisateur **transmet** cette information à l'ordinateur par l'intermédiaire du clavier ou de la souris.

Cette transmission de données à l'ordinateur est appelée **entrée, input** ou encore **saisie**.

Après **traitement**, le programme fournit un **résultat** à l'utilisateur, soit par l'intermédiaire de l'écran, soit sous forme de fichiers.

Ce retour de données est appelé **sortie, output** ou encore **affichage**.

## Coder l'information.

De par la nature de ses composants électroniques, le robot ne perçoit que deux états: composant **allumé** et composant **éteint**. De cette perception découle le **langage binaire**, qui utilise par convention les deux symboles **0** et **1**.

Ne connaissant que le **0** et le **1**, l'ordinateur utilise un code pour représenter une information aussi simple qu'un nombre entier ou un caractère.

Ce code est un programme, qui différencie chaque type d'information et transforme une information (donnée numérique ou alphabétique) en valeurs binaires. À l'inverse, ce programme sait aussi transformer un nombre binaire en valeur numérique ou alphabétique.

Toute information fournie à l'ordinateur est, au bout du compte, codée en binaire. L'information peut être un simple nombre ou une instruction de programme.

Il existe autant de codes que de types d'informations. Cette différenciation du codage (en fonction de ce qui doit être représenté) introduit la notion de **type de données**.

Remarque :

Le code binaire associé à chaque code d'opération (addition, test, etc.) n'est pas nécessairement identique d'un ordinateur à un autre. Ce code binaire est déterminé par le constructeur de l'ordinateur. De ce fait, une instruction telle que l'addition de deux nombres n'a pas le même code binaire d'une machine à une autre. Il existe donc, pour un même programme, un code binaire qui diffère suivant le type d'ordinateur utilisé.

## La notion d'INSTRUCTION.

Les instructions exécutées par l'ordinateur sont des traitements élémentaires simples (comparaison de valeurs, addition, soustraction, multiplication ou division) opérant sur des données que l'ordinateur puise dans sa mémoire.

Ces données, préalablement transmises par l'utilisateur ou par le programme lui-même, sont des assemblages de bits (valeurs de 0 ou de 1).

Ces assemblages sont de deux types :

- l'**octet**, qui correspond à huit bits
- le **mot**, terme plus général qui comprend un nombre de bits variant selon les ordinateurs (8, 16, 32...)

En savoir plus :

<http://www.commentcamarche.net/langages/langages.php3>

## La notion de VARIABLE.

Une instruction machine effectue des opérations sur des **valeurs** repérées par leur **adresse**. Une telle adresse est représentée par un **nom**.

Une **variable** est précisément ce nom qui sert à désigner un **emplacement** donné de la mémoire centrale.

Cette notion, simple en apparence, contribue considérablement à faciliter la réalisation de programmes. Elle vous permet, en effet, de manipuler des valeurs sans avoir à vous préoccuper de l'emplacement qu'elles occuperont effectivement en mémoire: il vous suffit simplement de leur choisir un nom.

Bien entendu, la chose n'est possible que parce qu'il existe un programme de traduction (compilateur ou interprète) de votre programmation vers le langage machine : c'est lui qui attribuera une adresse numérique binaire à chaque nom de variable.

En savoir plus :

<http://www.commentcamarche.net/langages/structure.php3>



## La notion de TYPE.

Les informations conservées en mémoire sont toujours codées en langage binaire. Cela est vrai en particulier pour le contenu des variables. Comme il est nécessaire de pouvoir conserver des informations de nature différente (par exemple des **nombres** et des **lettres**), il faut employer plusieurs codes différents. Dès lors, la connaissance du contenu binaire d'une variable ne suffit pas pour déterminer l'information correspondante. Il est nécessaire de savoir, en outre, comment la valeur qui s'y trouve a été codée. Cette distinction correspond à la notion de **type**.

Ainsi, une variable destinée à recevoir des valeurs telles que **123** ou **12,45** est de type **numérique**. Une variable destinée à contenir des **lettres** sera dite de type **caractère**.

Le type limite les opérations : les opérations arithmétiques (addition, soustraction, multiplication, division), possibles avec des variables numériques, n'ont aucun sens pour des variables de type caractère. Par contre, les comparaisons seront possibles pour les deux types.

En résumé, le type d'une variable définit:

- la **nature des informations** qui seront représentées dans la variable (numériques, caractères).
- le **codage** utilisé.
- les **limitations** concernant les valeurs qui peuvent être représentées.
- les **opérations** réalisables avec les variables correspondantes.

En savoir plus :

<http://www.commentcamarche.net/langages/structure.php3>



A chalkboard with handwritten text in white chalk. The text is arranged in four lines. The first line contains the number '1955'. The second line contains the number '920'. The third line contains the number '141523', which is underlined with a horizontal line. The fourth line contains the string 'addcih'.

1955  
920  
141523  
addcih

Dans la vie courante, les êtres humains n'ont pas besoin de préciser le type des informations qu'ils échangent, car il est explicite : nous comprenons que 23 représente un nombre tandis que DUPONT est une suite de caractères. Ce n'est pas le cas pour l'ordinateur : il faut lui expliquer la différence.

.programme

Un programme traite l'information selon ce schéma :

### **Input.**

Entrée,  
saisie.

L'information  
est transmise à l'ordinateur

- **par l'utilisateur**  
(clavier, souris, capteur)

- **par le programme**  
lui-même qui les prélève  
sur le disque dur (archives,  
base de données).

### **Traitement.**

Processus,  
exécution.



### **Output.**

Sortie,  
résultat,  
retour,  
affichage.

Le programme fournit un **ré-  
sultat** à l'utilisateur, soit par  
l'intermédiaire de l'écran,  
soit sous forme de fichiers.

Une **application** = un **logiciel** = un ensemble de programmes  
(segments, modules séparés, ressources, appels de données, etc).



## Un système d'exploitation

(Operating System ou **OS** : Windows, Mac OSX, Linux)  
est constitué d'un ensemble de programmes  
qui assure le démarrage de l'ordinateur.

L'**OS** permet la reconnaissance et la communication  
entre les ressources matérielles d'un ordinateur (périphériques)  
et les applications (logiciels)  
sollicitées par l'utilisateur pour lire ou produire  
des données (texte, images, son, etc).



## Une **application**

permet de créer, de traiter ou d'exécuter un document :  
une image fixe ou en mouvement, un son.



Un **player**

permet de consulter un document, mais pas de le modifier.



## L'interactivité

est une caractéristique de certains types de documents qui contiennent une partie programmée, définissant les conditions et le type d'action possible sur une image, un son, etc.

Seule la programmation permet de réaliser des animations interactives ou des composants réactifs.

Un document interactif associe **2 types de composants** :

- un «objet» (image, son, etc).
- le programme qui gère l'interaction.

Les applications permettant de réaliser des documents interactifs contiennent un environnement spécifique de création, réservé à la programmation des comportements attendus ou générés.



## Un **programme génératif**

est capable d'évoluer et/ou de se modifier  
en fonction d'une information environnementale (capteur),  
d'une valeur aléatoire (hasard),  
voire même de données qu'il aura lui-même produites.

Un programme génératif

ne nécessite pas forcément l'action d'un élément extérieur à lui-même

Un programme génératif

ne présente jamais le même résultat.



