

Quest-ce qu'un
ordinateur ?
Notions de
Matériel et logiciel

Pierre Huguet

PLAN

- Matériel et logiciel (Hardware and Software)
- Types d'ordinateurs : généralistes et spécialisés
- Matériel : Qu'est-ce qu'un ordinateur ?
 - L'architecture de Von Neumann
 - Exemples
- Logiciel :
 - Langages
 - Architecture par couches (abstraction layers)
- Le Cloud
- Quiz

LE MATÉRIEL (HARDWARE)

- Composants électroniques et mécaniques de l'ordinateur.



Portable



Smart phone



Ordinateur de bureau



Disque dur



Composant mémoire



Mémoire Flash

LE LOGICIEL (SOFTWARE)

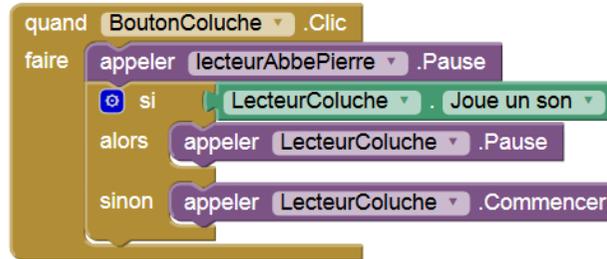
- Programme ou code informatique qui contrôle le matériel.

```
(define fact!  
  (λ (n)  
    (let loop ((n n) (acc 1))  
      (if (= n 1)  
          acc  
          (loop (- n 1) (* acc n))))))
```

Scheme

```
# This program says hello and asks for my name.  
print('Hello world!')  
print('What is your name?')  
myName = input()  
print('It is good to meet you, ' + myName)]
```

Python



Scratch, App Inventor

```
#include <stdio.h>  
  
main()  
{  
  printf("hello, world\n");  
}
```

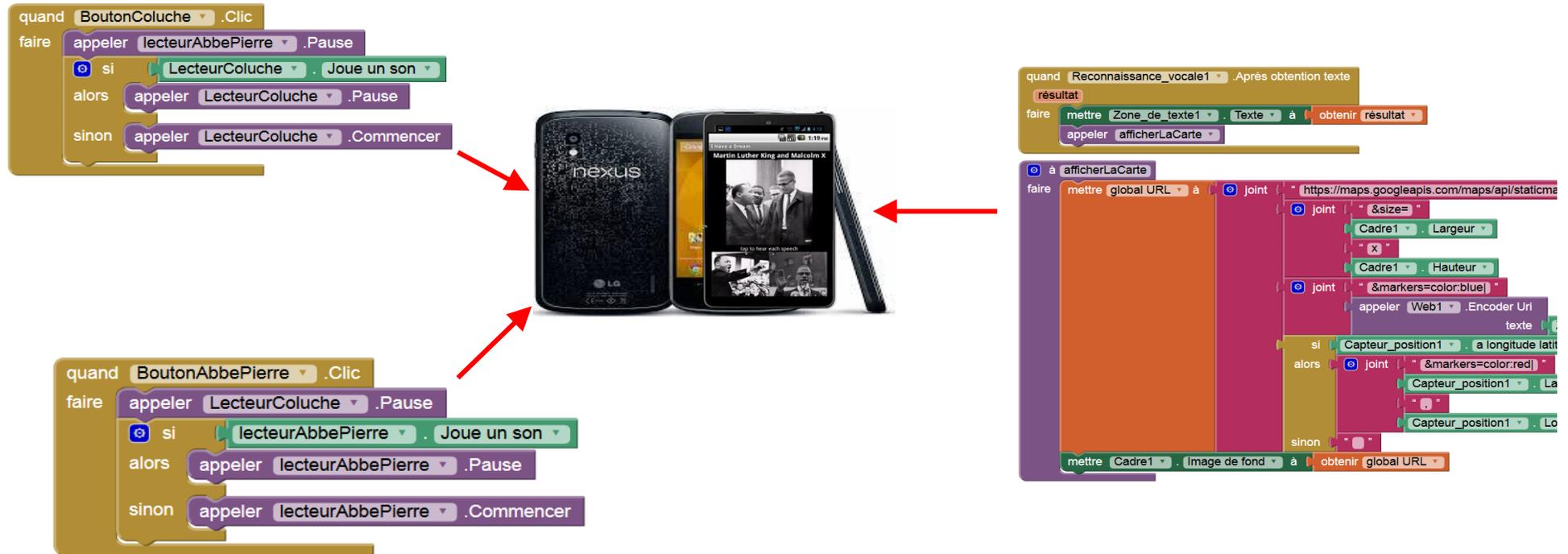
C

```
public class Hello {  
    public void main (String args[]) {  
        if (System.currentTimeMillis() > 1000000)  
            System.out.println("Good morning");  
        else  
            System.out.println("Good evening");  
    }  
}
```

Java

LES ORDINATEURS GÉNÉRALISTES

- Un ordinateur généraliste est un ordinateur qui peut exécuter une variété de programme choisis par nous



ORDINATEURS SPÉCIALISÉS



Programme d'une montre

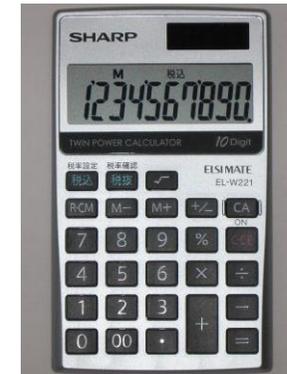


Programme de contrôle du micro-onde

- Les ordinateurs spécialisés exécutent un programme défini, que l'on ne peut pas changer.



Programmation ABS



Calculatrice

DÉVELOPPEUR ?

- Un développeur ou analyste programmeur est un informaticien qui réalise des logiciels et les met en œuvre à l'aide de langages de programmation, conçoit et écrit des logiciels.
(Wikipedia)
- Vous avez commencé!
Vous serez bientôt un programmeuse ou un programmeur avertis!



Vous

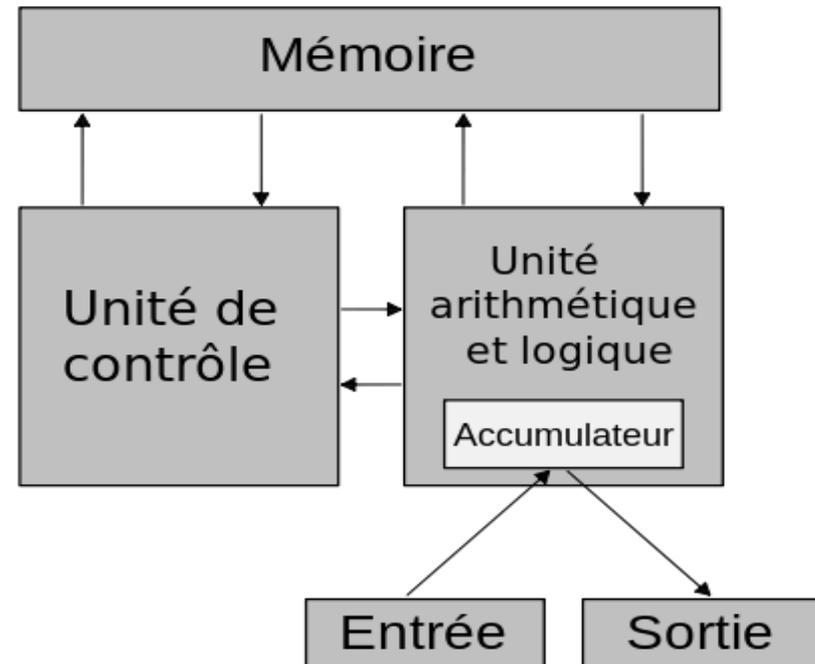
Le matériel



QUE FAUT-IL AU MINIMUM DANS UN ORDINATEUR ?

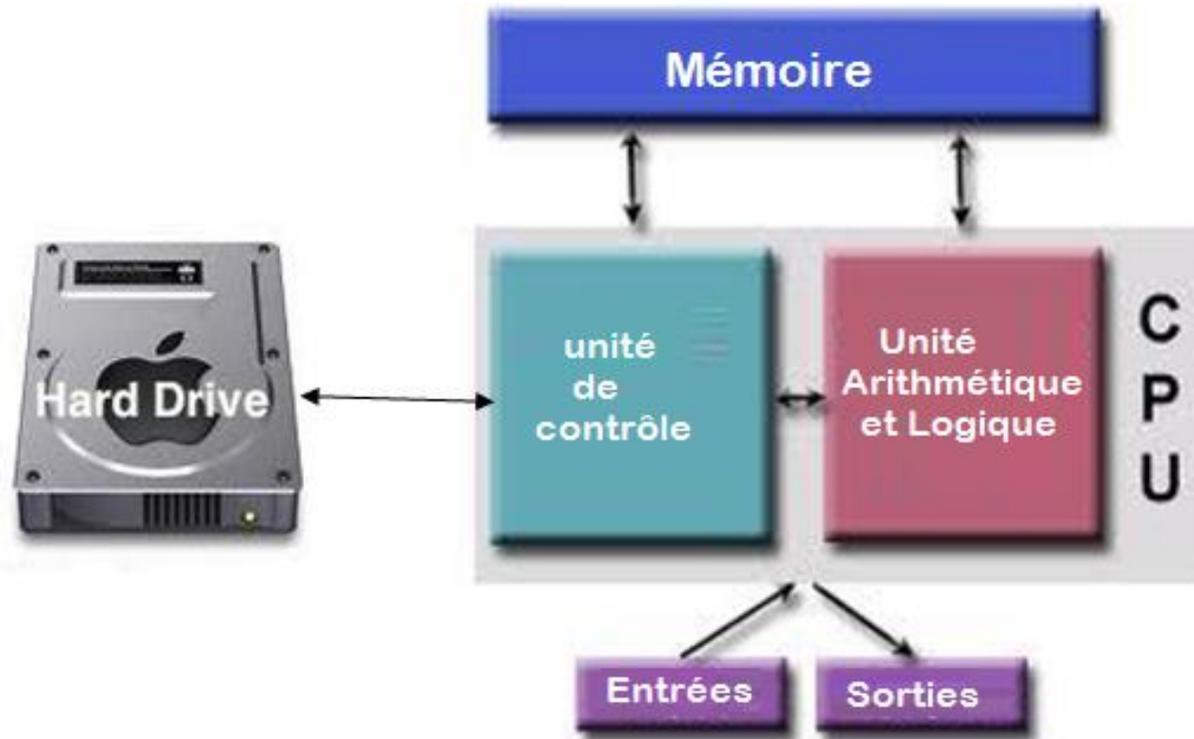
- L'architecture de Von Neumann (1945) :

- l'unité arithmétique et logique : effectue les opérations de base,
- l'unité de contrôle: chargée du « séquençage » des opérations,
- la mémoire contient les données et le programme qui indiquera à l'unité de contrôle quels sont les calculs à faire sur ces données,
- les dispositifs d'entrée-sortie, qui permettent de communiquer avec le monde extérieur.

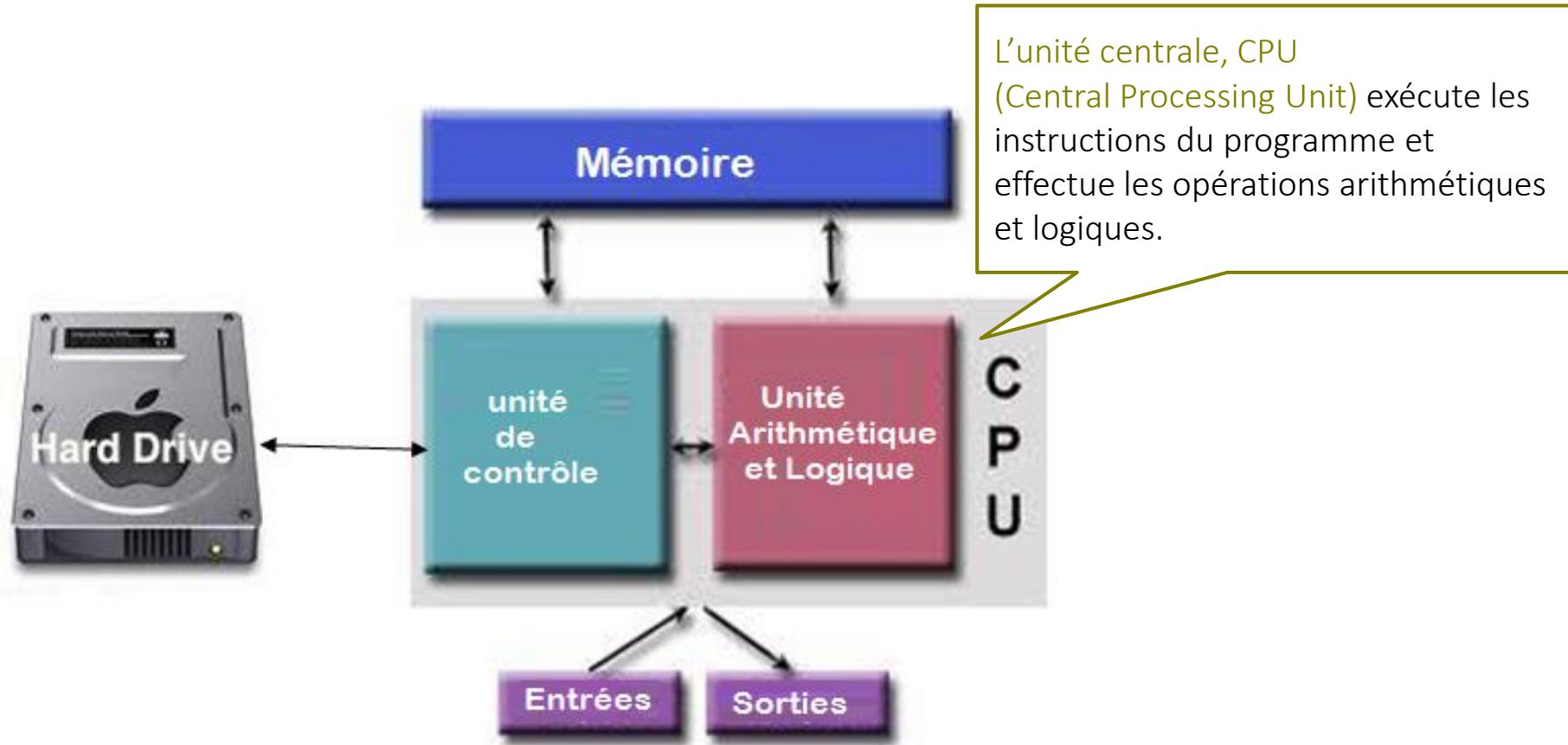


(d'après Wikipedia)

PRINCIPAUX COMPOSANTS MATÉRIELS



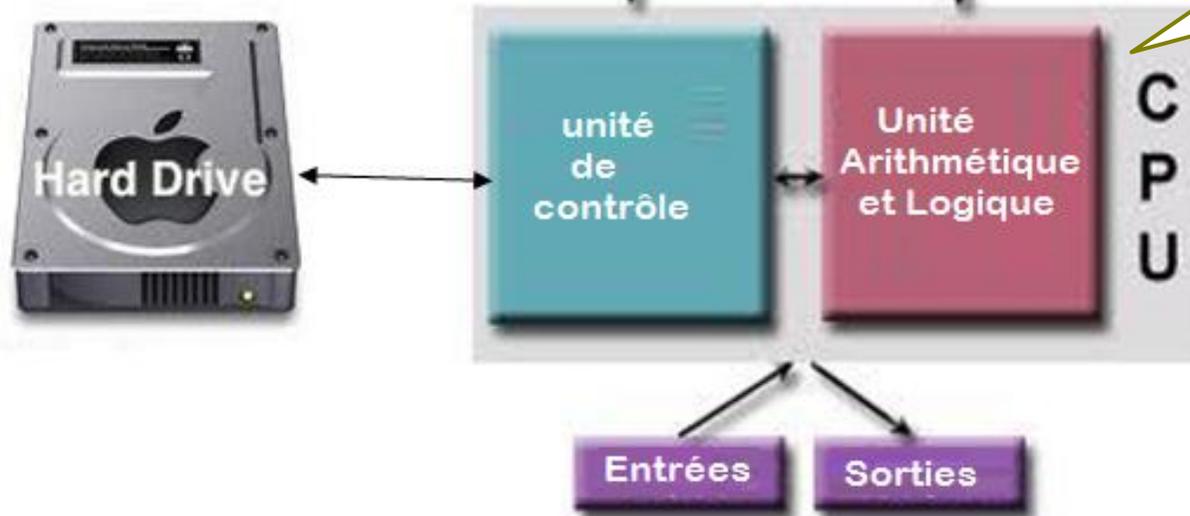
PRINCIPAUX COMPOSANTS MATÉRIELS



PRINCIPAUX COMPOSANTS MATÉRIELS

La mémoire vive, ou Random Access Memory (RAM) stocke **temporairement** les programmes et les données lorsque l'ordinateur est allumé

L'unité centrale, CPU (Central Processing Unit) exécute les instructions du programme et effectue les opérations arithmétiques et logiques.



PRINCIPAUX COMPOSANTS MATÉRIELS

La mémoire vive, ou Random Access Memory (RAM) stocke **temporairement** les programmes et les données lorsque l'ordinateur est allumé



Mémoire



L'unité centrale, CPU (Central Processing Unit) exécute les instructions du programme et effectue les opérations arithmétiques et logiques.

Entrées

Sorties

Les entrées /sorties E/S : Les *entrées* (souris, écran tactile, micro, ...) transmettent à l'ordinateur des informations sur l'environnement.

Les *Sorties* (écran, haut parleur) transfèrent de l'information de l'ordinateur vers l'extérieur.

PRINCIPAUX COMPOSANTS MATÉRIELS

La mémoire vive, ou Random Access Memory (RAM) stocke **temporairement** les programmes et les données lorsque l'ordinateur est allumé



Les *périphériques de stockage, Storage devices* (disques, mémoire flash, CDs,) stockent les données de façon **permanente**. Elles restent enregistrées même après avoir éteint l'ordinateur.

Mémoire

L'unité centrale, CPU (Central Processing Unit) exécute les instructions du programme et effectue les opérations arithmétiques et logiques.

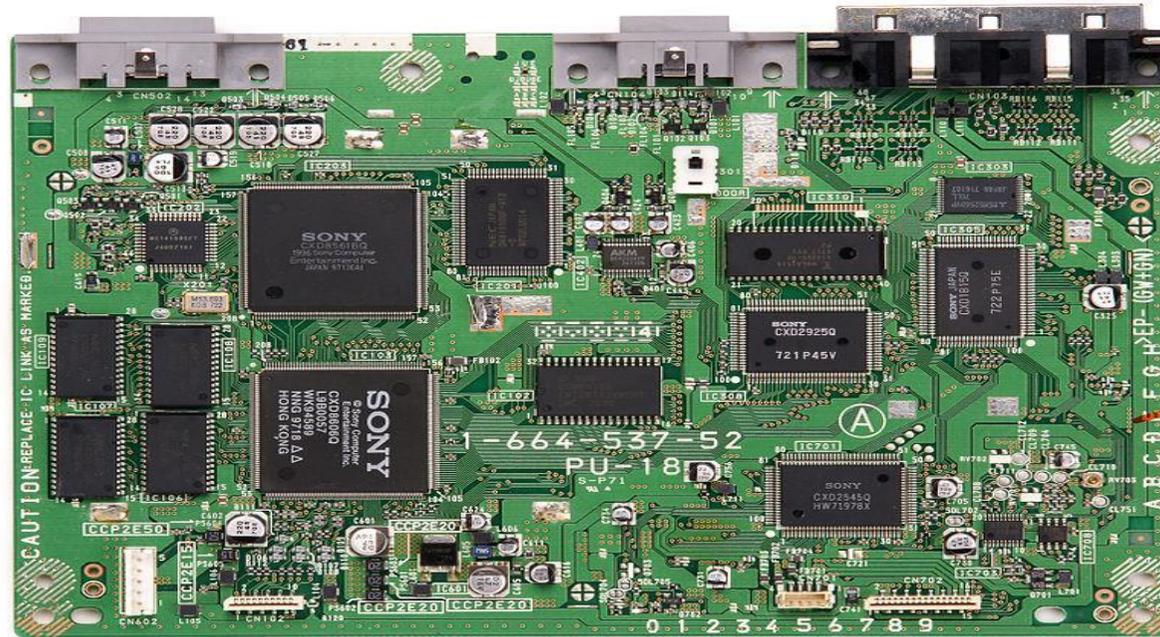


Les entrées /sorties E/S : Les *entrées* (souris, écran tactile, micro, ...) transmettent à l'ordinateur des informations sur l'environnement.

Les *Sorties* (écran, haut parleur) transfèrent de l'information de l'ordinateur vers l'extérieur.

CARTE MÈRE ET CIRCUITS (CHIPS)

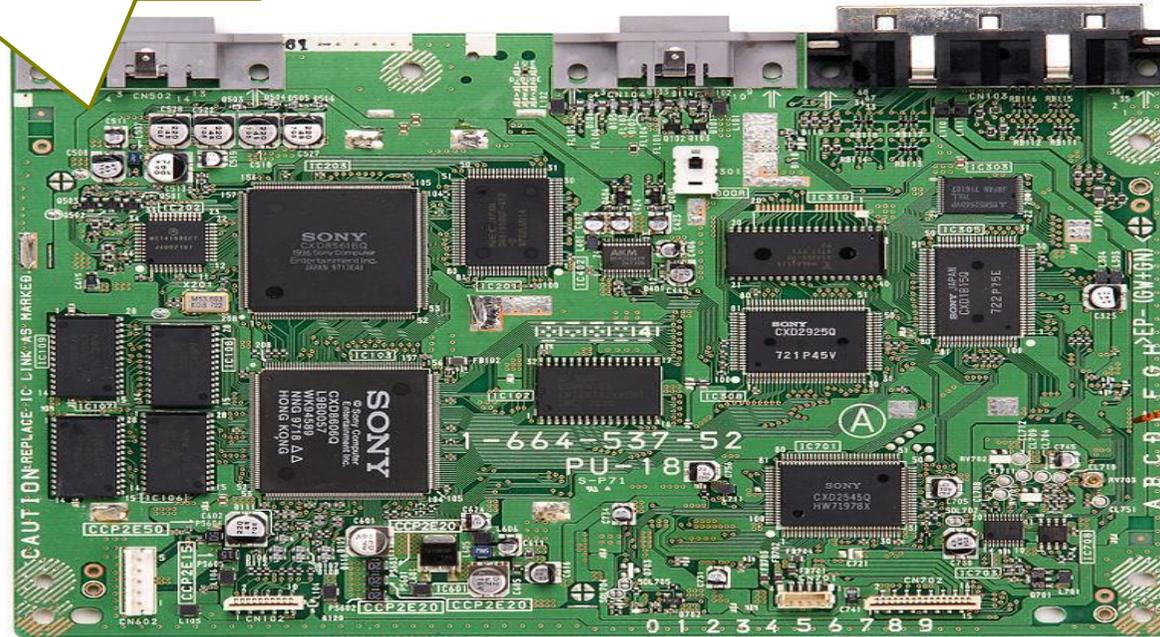
Sony Playstation



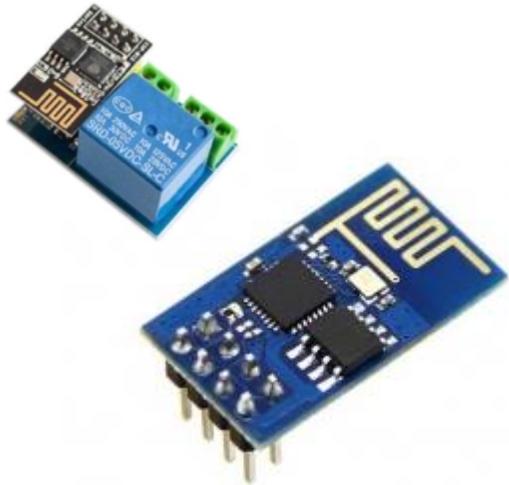
CARTE MÈRE ET CIRCUITS (CHIPS)

La *carte mère* est un circuit imprimé sur lequel sont installés les principaux composants électroniques de l'ordinateur.

Sony
Playstation



LES OBJETS CONNECTÉS



ESP8266 ordinateur avec
Internet et Entrées sorties
Environ 1 €

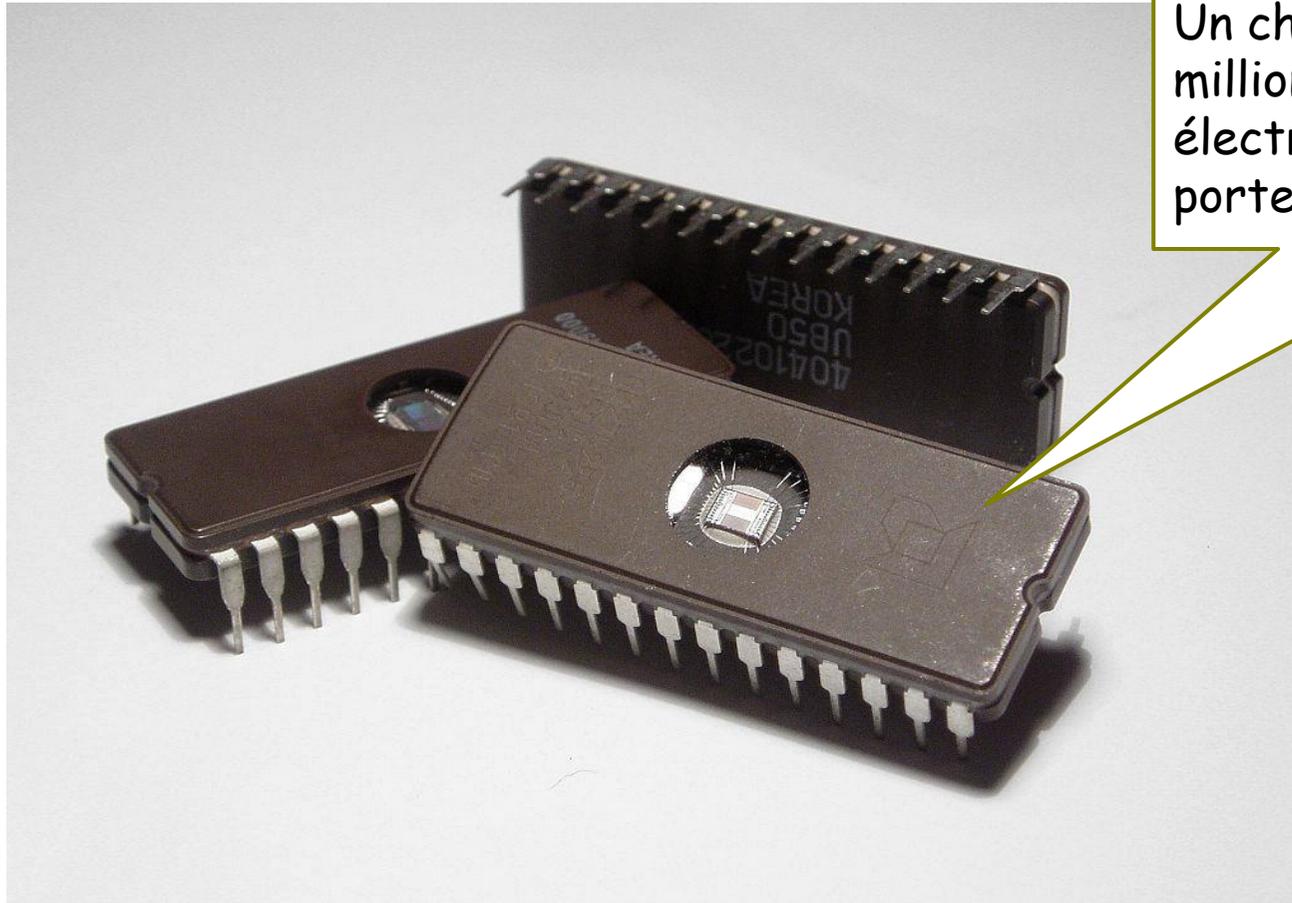


Arduino
10 à 20€



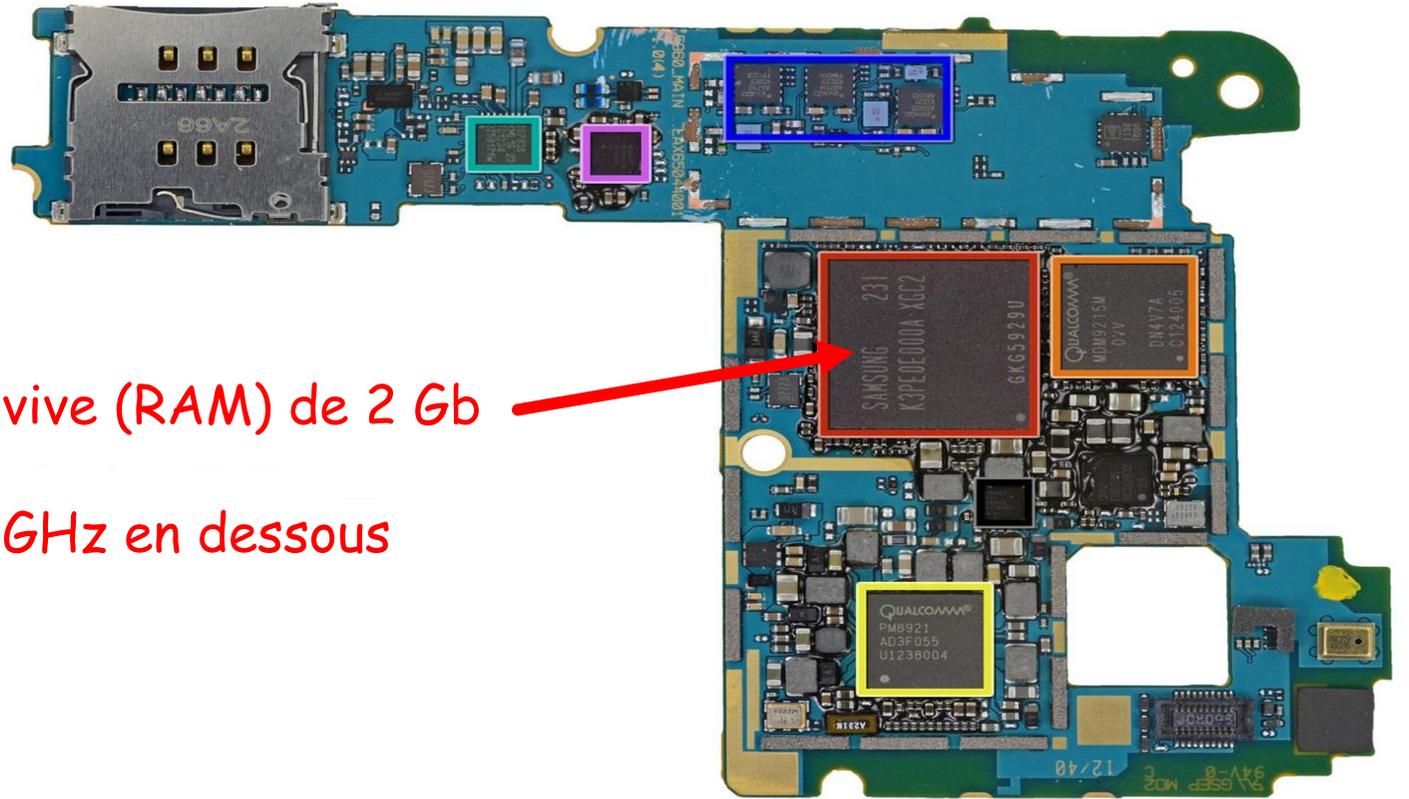
Raspberry Pi
20 à 40 €
Ordinateur complet

CIRCUIT INTÉGRÉ



Un chip peut contenir des millions de composants électroniques (diodes, switchs, portes logiques, circuits).

CARTE MÈRE D'UN NEXUS 5 (FACE ARRIÈRE)



Composant mémoire vive (RAM) de 2 Gb

CPU Snapdragon 1.4 GHz en dessous

Src: <http://www.ifixit.com/Teardown/Nexus+4+Teardown/11781/3>

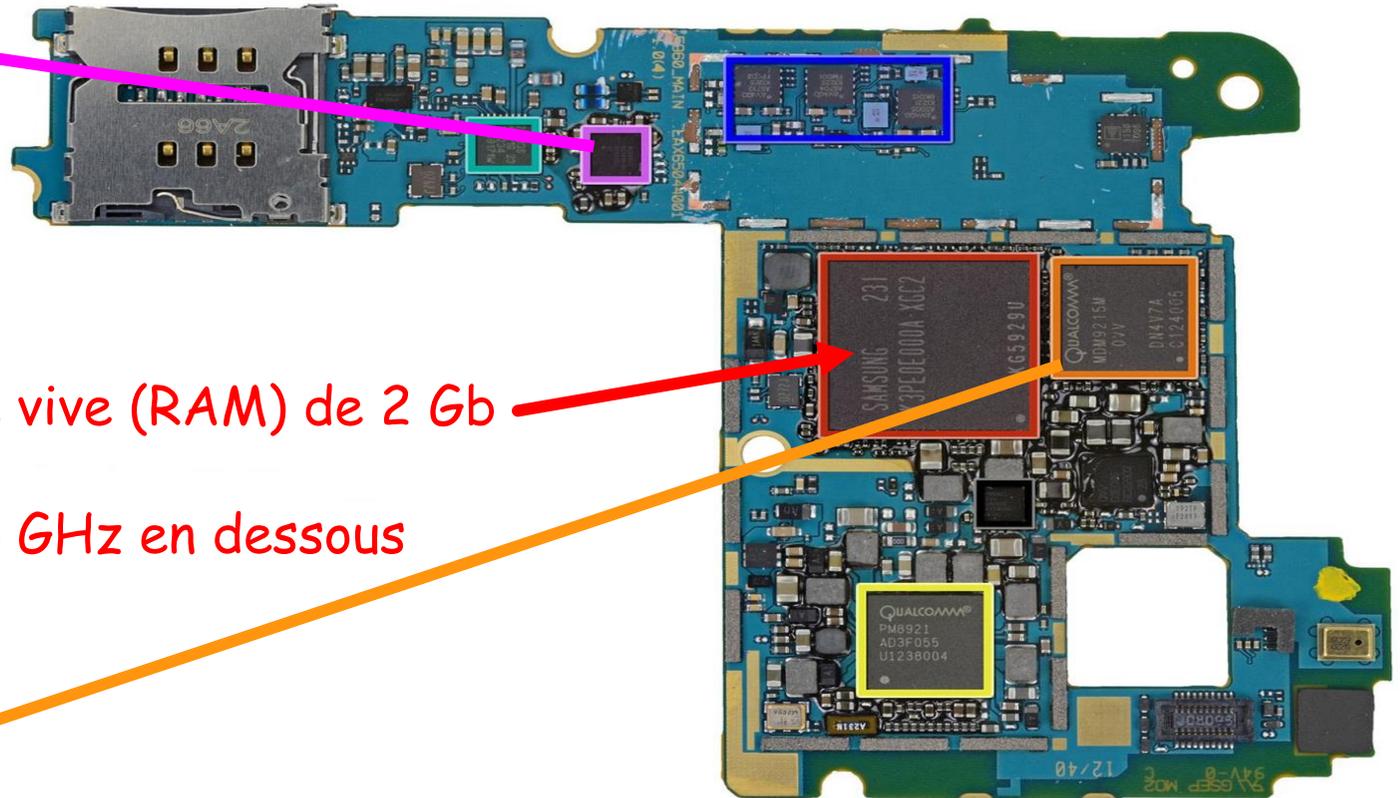
CARTE MÈRE D'UN NEXUS 5 (FACE ARRIÈRE)

Codeur/cécodeur
(codec) audio
Qualcomm

Composant mémoire vive (RAM) de 2 Gb

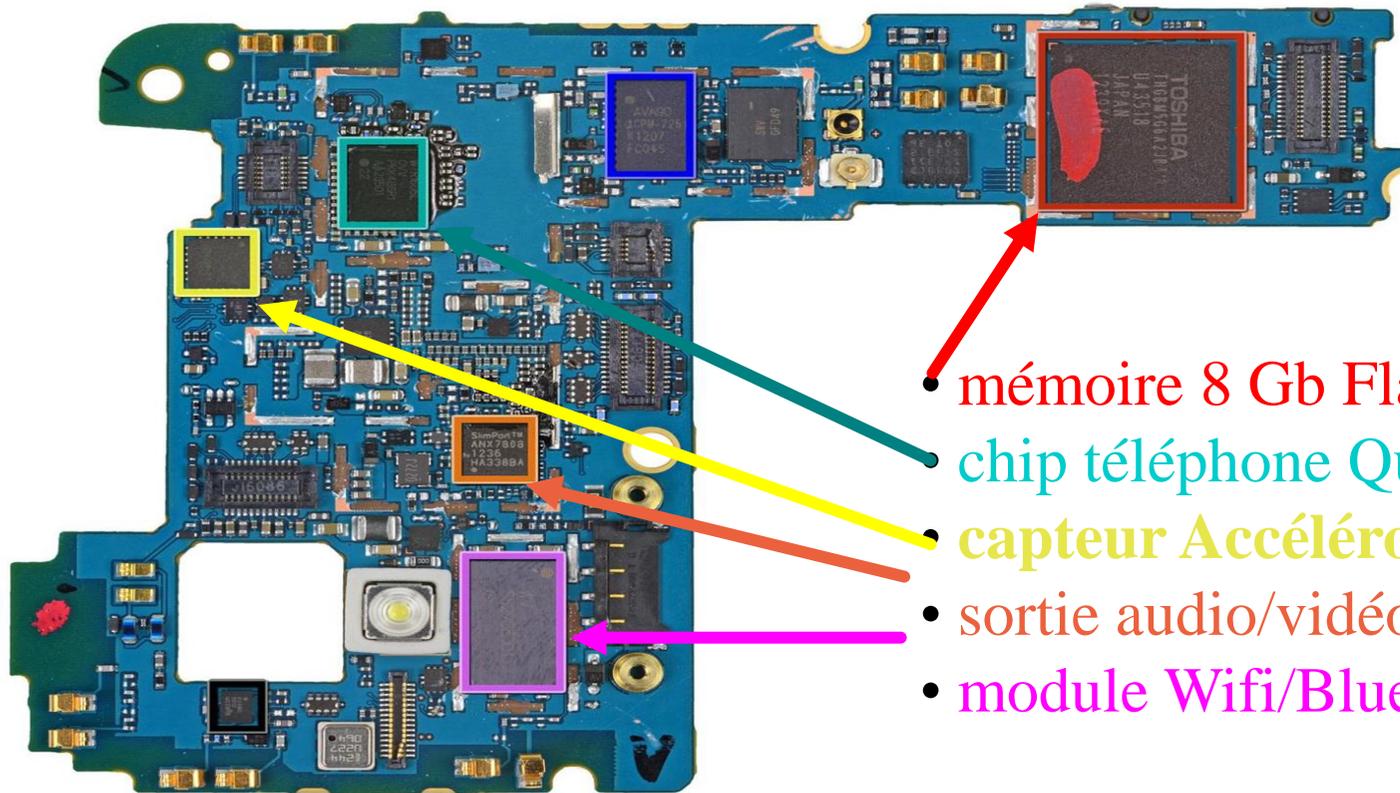
CPU Snapdragon 1.4 GHz en dessous

Modem Qualcomm



Src: <http://www.ifixit.com/Teardown/Nexus+4+Teardown/11781/3>

CARTE MÈRE D'UN NEXUS 4 (FACE AVANT)



- mémoire 8 Gb Flash – stockage permanent
- chip téléphone Qualcomm 4 Gb LTE
- capteur Accéléromètre
- sortie audio/vidéo vidéo HDMI
- module Wifi/Bluetooth

Src: <http://www.ifixit.com/Teardown/Nexus+4+Teardown/11781/3>

Le logiciel



LOGICIEL : PROGRAMME INFORMATIQUE

- Un **programme informatique** est une liste d'instructions qui contrôlent l'ordinateur.

Programme Assembleur

```
TrapBrea proc near
    push ds
    push cs
    pop ds
    mov dx,offset PGROUP:Bret
    mov ah,25h
    mov al,23h
    int 21h
    pop ds
    ret
```

Programme javascript

```
<script>
var text = "";
var i = 0;
while (i < 10) {
    text += "<br>The number is " + i;
    i++;
} </script>
```

Programme scratch



Programme App Inventor



Programme Java

```
public class Hello {

    public void main (String args[]) {
        if (System.currentTimeMillis() > 1000000)
            System.out.println("Good morning");
        else
            System.out.println("Good evening");
    }
}
```

INTERFACES LOGICIELLES

- Le logiciel est une interface entre nous et nos ordinateurs



Human/Computer interface



Exemple : App Inventor ou Scratch



UNE ARCHITECTURE LOGICIELLE PAR COUCHES



Couche : application – Angry Birds

```
#import <Foundation/Foundation.h>
#include <stdio.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    float x,y;
    const char* string = "3.1415 6.28";
    sscanf(string, "%f %f", &x, &y);

    NSLog(@"x = %.4f, y = %.2f", x, y);

    [pool drain];
    return 0;
}
```



Couche : Matériel – smartPhone (Caméra, écran, accéléro, GPS, ...)

UNE ARCHITECTURE LOGICIELLE PAR COUCHES



Couche : application – Angry Birds

interface application / OS

Couche : Système d'exploitatin OS

Logiciel de chargement
d'appli (loader)

Logiciel de contrôle
de la Caméra

Logiciel de contrôle
de l'écran tactile

Interfaces entre logiciel et matériel

Couche : Matériel – smartPhone (Caméra, écran, accéléro, GPS, ...)

```
#import <Foundation/Foundation.h>
#include <stdio.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    float x,y;
    const char* string = "3.1415 6.28";
    sscanf(string, "%f %f", &x, &y);

    NSLog(@"x = %.4f, y = %.2f", x, y);

    [pool drain];
    return 0;
}
```



UNE ARCHITECTURE LOGICIELLE PAR COUCHES



Couche : application – Angry Birds

interface application / OS

Couche : Système d'exploitatin OS

Logiciel de chargement
d'appli (loader)

Logiciel de contrôle
de la Caméra

Logiciel de contrôle
de l'écran tactile

Interfaces entre logiciel et matériel

Couche : Matériel – smartPhone (Caméra, écran, accéléro, GPS, ...)

```
#import <Foundation/Foundation.h>
#include <stdio.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

    float x,y;
    const char* string = "3.1415 6.28";
    sscanf(string, "%f %f", &x, &y);

    NSLog(@"x = %.4f, y = %.2f", x, y);

    [pool drain];
    return 0;
}
```



LE LANGAGE MACHINE

Le langage Machine – lu par la machine

00101100 1011011000111100

00110011 1110100011100010

00111001 1010100001100001

LE LANGAGE MACHINE

Le langage Machine – lu par la machine

00101100 1011011000111100

00110011 1110100011100010

00111001 1010100001100001

Les ordinateurs ne comprennent que leur propre langage machine.

LE LANGAGE MACHINE

Le langage Machine – lu par la machine

```
00101100  1011011000111100
00110011  1110100011100010
00111001  1010100001100001
```

Les ordinateurs ne comprennent que leur propre langage machine.



LANGAGES DE HAUT ET BAS NIVEAUX

- Le logiciel est organisé par couches
- chaque couche masque les détails ou la complexité à la couche supérieure en lui fournissant une interface pour utiliser ses services : API

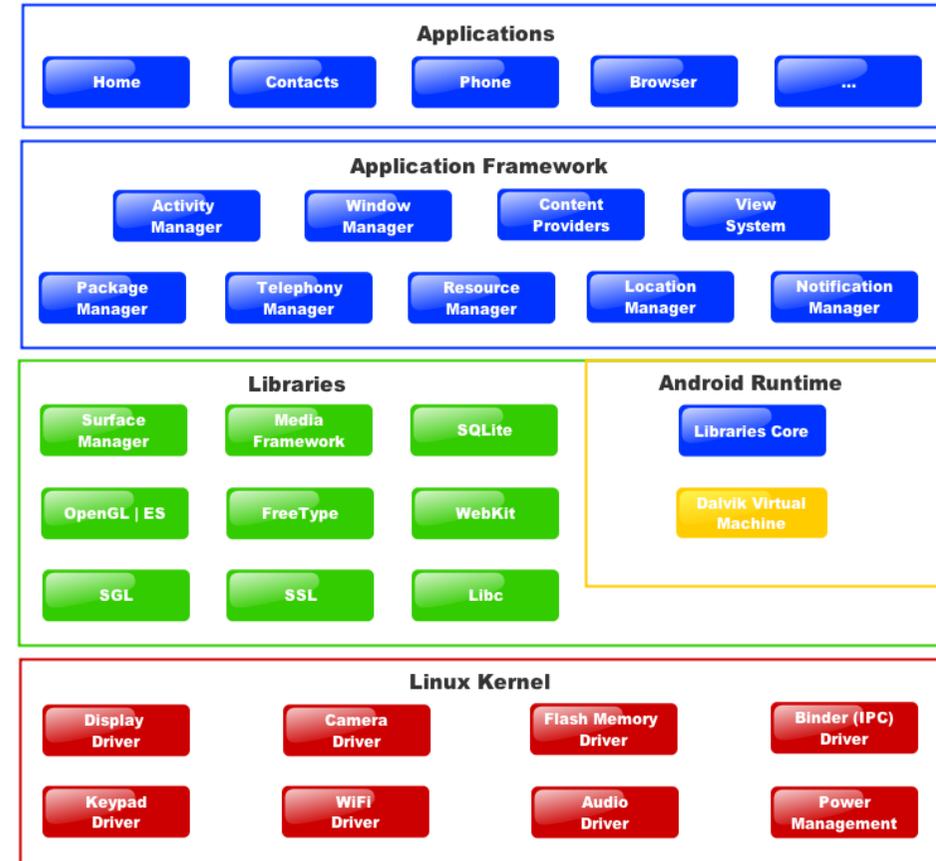
- c'est une "abstraction«

Abstraction par simplification (Wikipedia)

L'abstraction simplifie une réalité complexe pour pouvoir la penser. Il s'agit de négliger ce qui n'a pas d'importance au profit de ce qui en a. Au sens de l'abstraction par simplification l'abstrait est plus simple que le concret

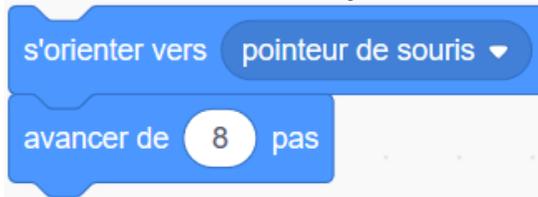
Le niveau bas s'appelle HAL : Hardware abstraction layer

Architecture d'Android



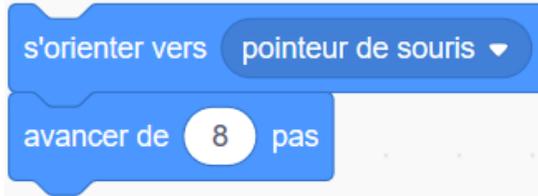
LANGAGES DE HAUT ET BAS NIVEAUX

Langage de haut niveau : compréhensible par l'homme



LANGAGES DE HAUT ET BAS NIVEAUX

Langage de haut niveau : compréhensible par l'homme

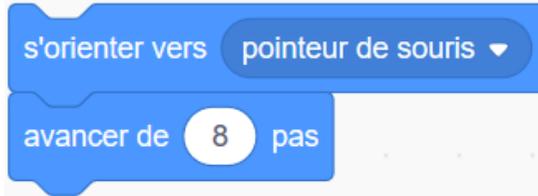


Langage Machine – lisible par la machine

```
00101100 1011011000111100
00110011 1110100011100010
00111001 1010100001100001
```

LANGAGES DE HAUT ET BAS NIVEAUX

Langage de haut niveau : compréhensible par l'homme



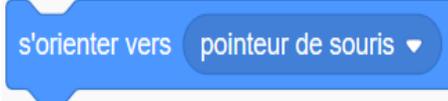
Chaque langage est associé à un logiciel de traduction du *code source* - de haut niveau compréhensible par l'homme en *code binaire* - de bas niveau exploitable par la machine

Langage Machine – lisible par la machine

```
00101100 1011011000111100
00110011 1110100011100010
00111001 1010100001100001
```

UN LANGAGE EST UNE ABSTRACTION

Code Scratch ou javascript -- une *abstraction* adaptée à nous



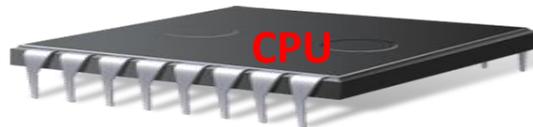
s'orienter vers pointeur de souris ▾

logiciel de traduction

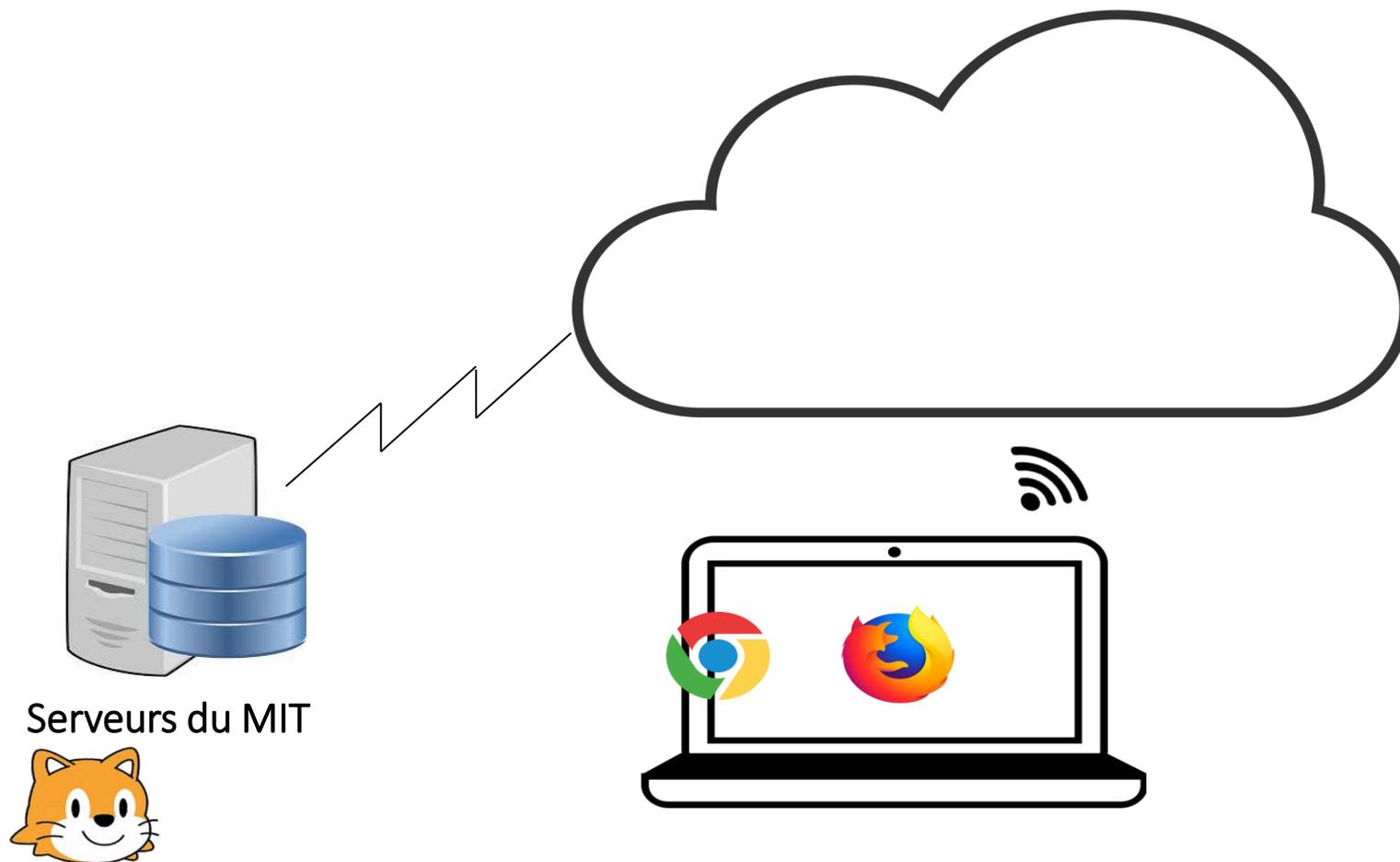
code binaire -- *abstraction* comprise par la machine.

00111001 1010100001100001

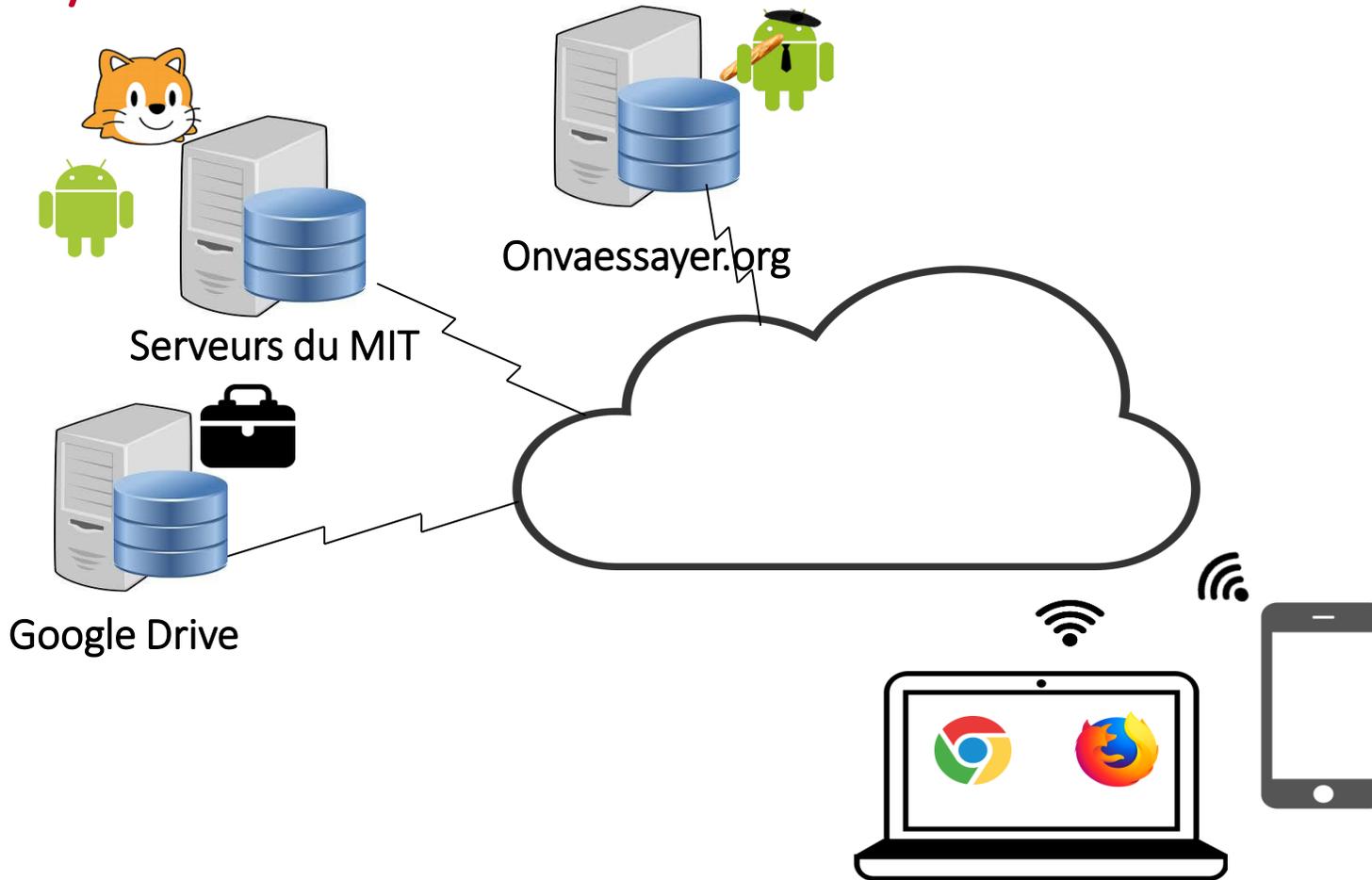
Interpréteur d'instructions binaires



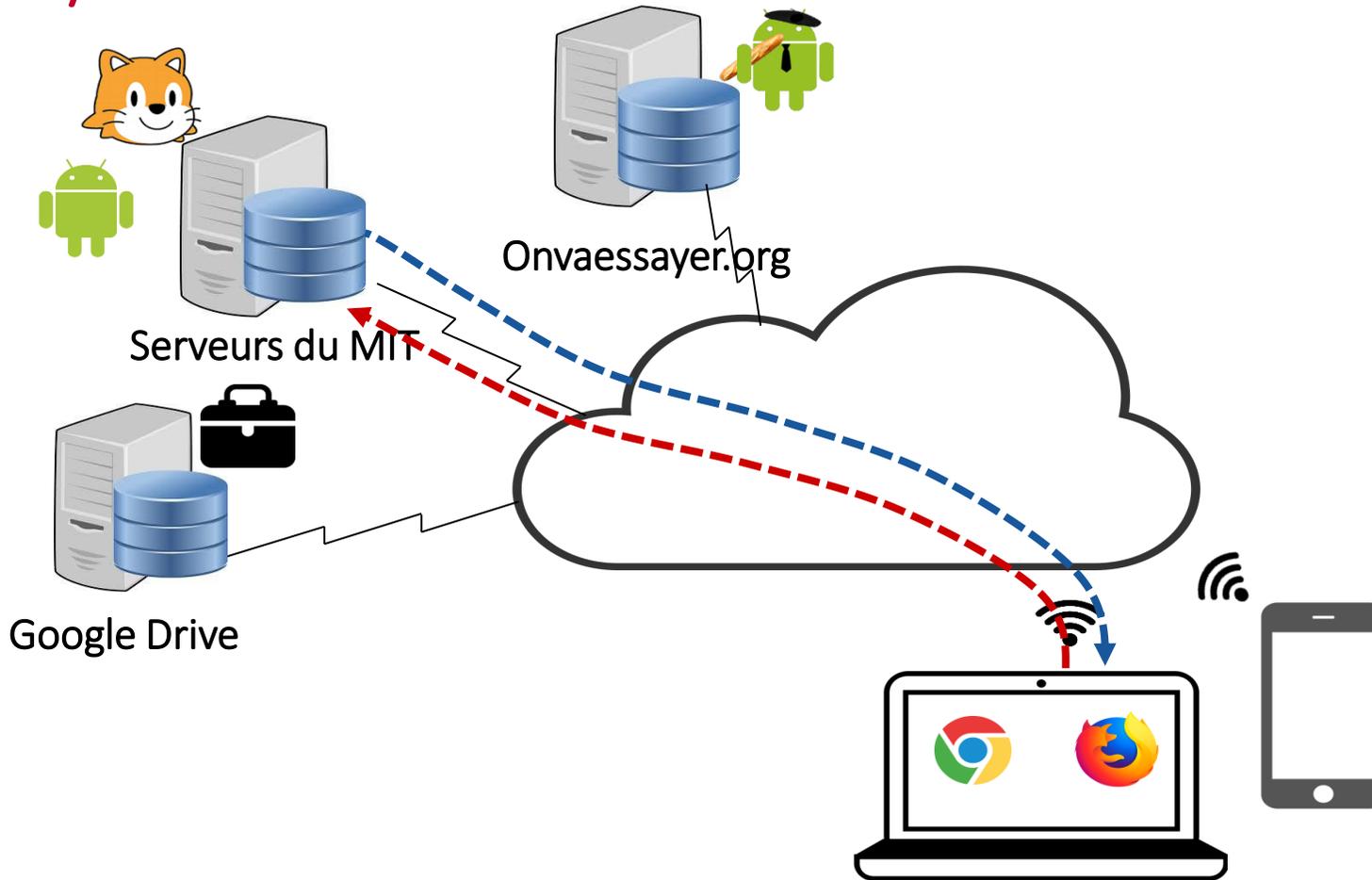
LE CLOUD



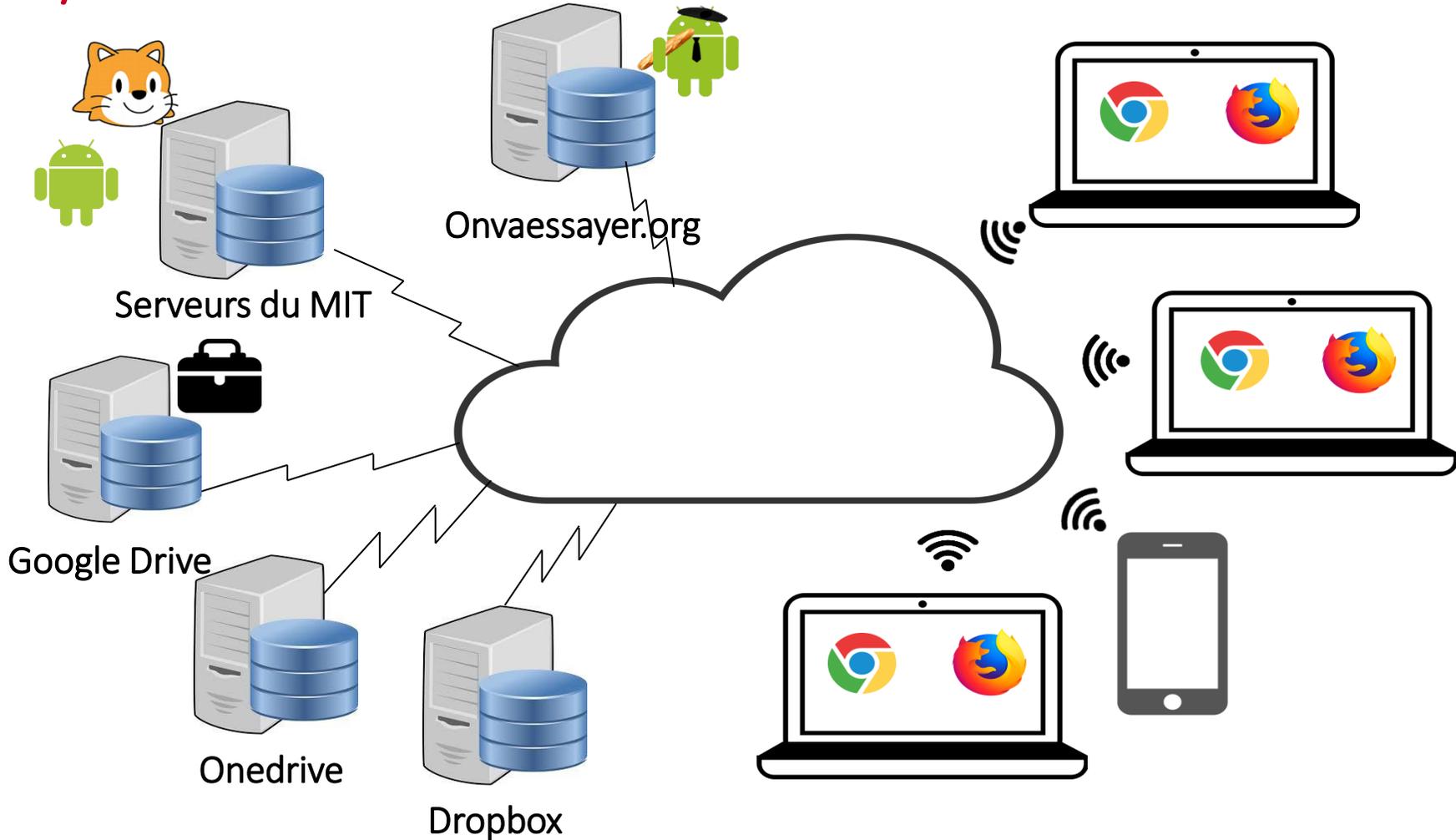
LE CLOUD / INTERNET



LE CLOUD / INTERNET



LE CLOUD / INTERNET



EXERCICE : METTEZ CHAQUE TERME EN FACE DE SA DÉFINITION

| Terme technique | Définition ou description |
|-----------------------------------|--|
| 1 Ordinateur | A Composants électroniques et mécaniques de l'ordinateur. |
| 2 Programme | B Ordinateur qui a un programme figé (ex : calculatrice, montre, freins d'une voiture). |
| 3 Matériel | C Contient les principaux composants électroniques de l'ordinateur. |
| 4 Logiciel | D Stocke temporairement les programmes et données temporairement lorsque l'ordinateur est branché. |
| 5 Ordinateur généraliste | E Une machine traite l'information sous le contrôle d'un programme. |
| 6 Ordinateur dédié | F Un circuit intégré (IC) qui contient des millions de petits circuits. |
| 7 RAM (mémoire à accès aléatoire) | G Un ordinateur qui peut exécuter beaucoup de programmes différents (ex : un smartphone). |
| 8 Unité centrale (CPU) | H Le matériel qui contient et exécute les instructions d'un programme. |
| 9 Carte mère | I Les programmes qui contrôlent l'ordinateur. |
| 10 Langage machine | J Un langage de programmation lisible par l'homme (ex : App Inventor) et qui permet au programmeur d'utiliser facilement des abstractions. |
| 11 Langage de haut niveau | K Un langage de programmation directement lisible par l'unité centrale de l'ordinateur. |
| 12 Composant électronique (chip) | L Une séquence d'instructions qui contrôle l'ordinateur. |