

# Informatique en CPP

- **Cours :**
  - Amphi 1 :** Matthieu Moy <Matthieu.Moy@imag.fr>
  - Amphi 2 :** Frédéric Devernay <Frederic.Devernay@inria.fr>
- **Travaux pratiques :**
  - **A :** Frédéric Devernay
  - **B :** Djamel Aouane <djamel.aouane@grenoble-inp.fr>
  - **C :** Matthieu Moy
  - **D :** Djamel Aouane

# Objectifs de l'enseignement

- Fondements de l'informatique
- Notion d'algorithme
  - Concevoir un algorithme
  - Se convaincre de sa correction
  - Évaluer ses performances (*complexité*)
- Méthodes numériques simples
- Langage utilisé : Python 3

# Organisation du cours

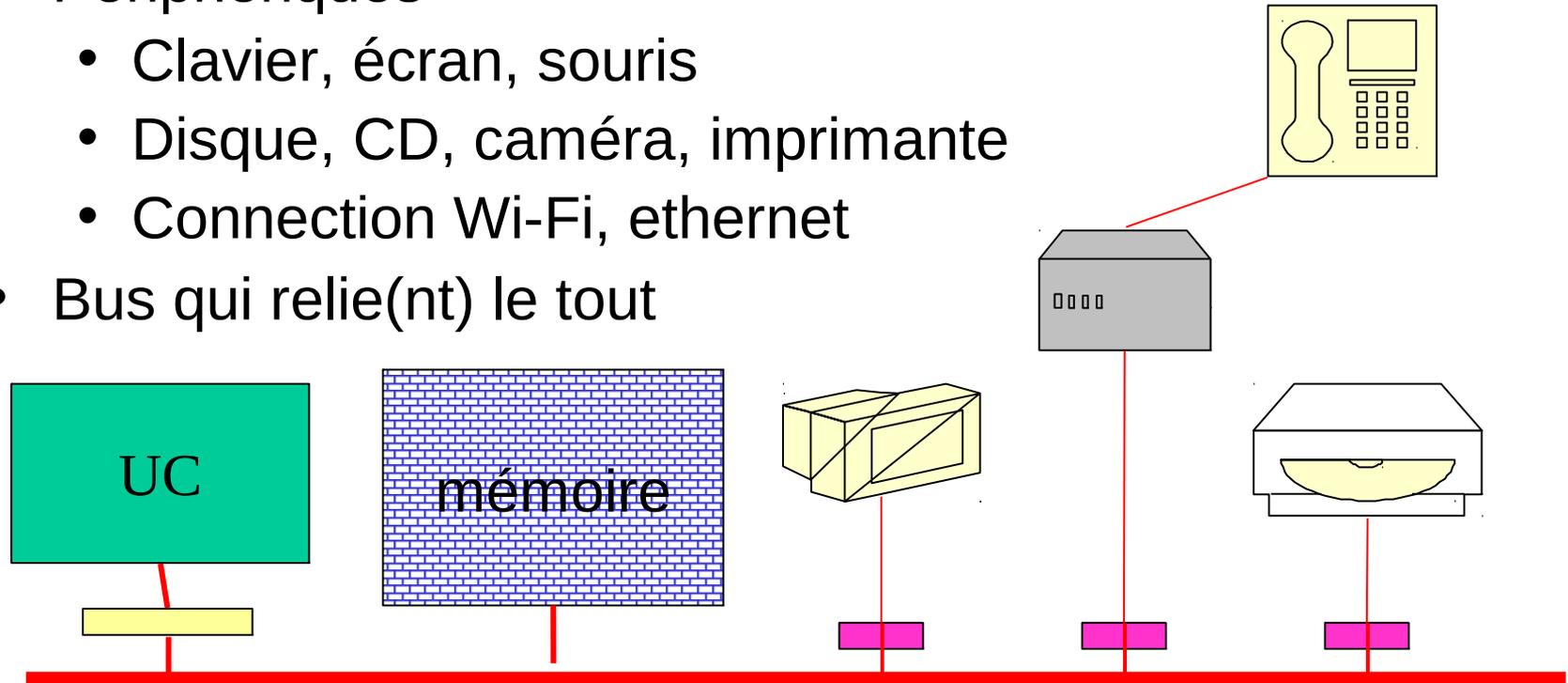
- 5 cours magistraux
- 5 séances machines (à l'Ensimag)
- 2 TP libre service notés
- 1 examen
  - une partie sur machine
  - une partie sur copie papier
- Supports en ligne :
  - <http://www-verimag.imag.fr/~moy/cours/infocpp-S2/>  
(en lien depuis mycpp.fr, section Informatique/P22)
- Prenez des notes !

# Objectifs de l'introduction

- Idées sur ce qu'est un ordinateur
- Qu'est ce qu'est le traitement d'information
- Notion d'algorithme
- Les logiciels
- Un peu d'histoire
- Exemples élémentaires d'information et de leurs traitements en Python

# Structure d'un ordinateur

- Mémoire, octet (~caractère), mot, adresse
- Unité centrale, instructions
- Périphériques
  - Clavier, écran, souris
  - Disque, CD, caméra, imprimante
  - Connection Wi-Fi, ethernet
- Bus qui relie(nt) le tout

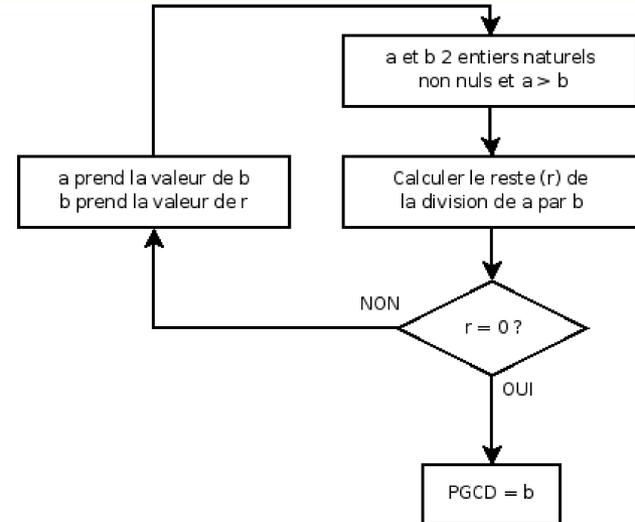


# Stockage de l'information : mémoire, disque dur

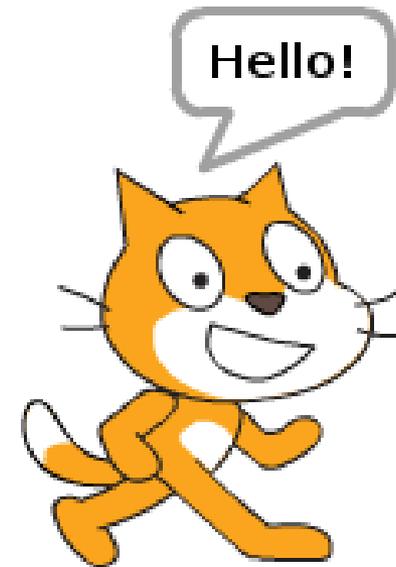
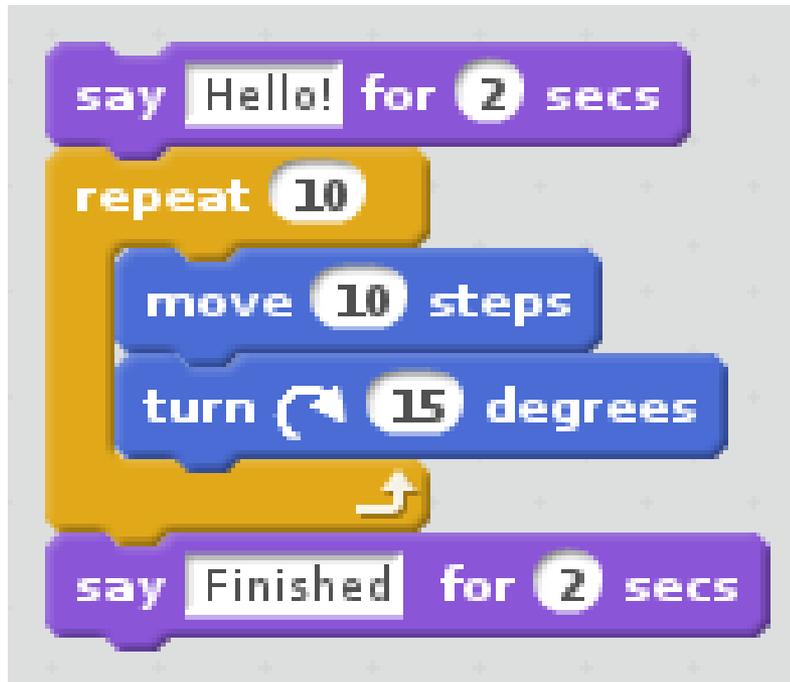
- Mémoire vive (RAM = Random Access Memory)
  - Utilisée pendant l'exécution des programmes
  - Perdue quand la machine est éteinte
  - En 2015, un ordinateur a typiquement 4 Go (gigaoctet = milliard d'octets) de RAM
- Disque dur (ou stockage SSD)
  - Stockage de l'information dans l'ordinateur
  - Préservée quand la machine est éteinte
  - En 2015, 1 disque dur = quelques centaines de Go

# Algorithme

- Algorithme d'Euclide
- Recettes de cuisine
- Montages en kit, etc
- Dans tous les cas :
  - Instructions élémentaires connues
  - Description d'une suite finie d'instructions
- Contre-exemples : ?



# Algorithme : exemple avec Scratch



[http://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](http://scratch.mit.edu/projects/editor/?tip_bar=getStarted)

# Programme

- Un algorithme
  - Formalisme de « calcul »
- Destiné à être exécuté sur ordinateur
  - Transposition dans une structure en machine
- Écrit dans un langage informatique
  - Description formelle : **syntaxe** + **sémantique**

# Les logiciels

- Logiciels d'application
  - Éditeur (texte, photo, ...)
  - Gestion d'entreprise, de ressources, ...
  - Calcul d'écoulement en mécanique des fluides,
  - ...
- **Langages de programmation et compilateurs**
- Système (Windows, Linux, ...)
  - Gestion de fichiers, d'utilisateurs, de périphériques, couplage réseau, ...
- Serveurs web, serveur mail, ...
  - Tournent sur une autre machine que la votre
  - Utilisés via un client (par exemple un navigateur)

# Langages de programmation

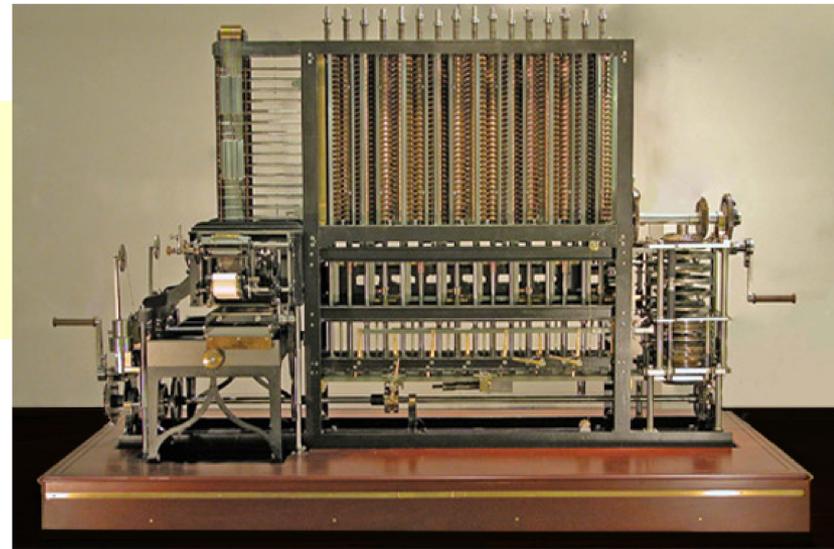
- Langage utilisé pour communiquer des instructions à un ordinateur (syntaxe et sémantique).
- Plus de 2000 langages différents :
  - Code machine
  - Fortran (1954)
  - Cobol (1960)
  - C (1972)
  - Ada (1983)
  - Caml
  - C++
  - Java
  - **Python**
  - ...

# Exemples d'informations et de leurs traitements

Informations	Traitements
Les nombres	Calcul scientifique
Maillage océan-atmosphère	Echanges d'énergie, prévision météo
Dossiers médicaux	Remboursement de soins
Objets 3D, articulations, trajectoires, modèles d'interactions	Calcul de positions, de vues, de bruits dans les jeux vidéo
Structure des chromosomes	Recherche de séquences proches

# Histoire simplifiée

# Les précurseurs



machine de Babbage(1792-1871)

machines de Pascal (1623-62)

- la carte perforée et les trieuses : Hollerith(1860-1929), Bull (1882-1925)
- les premiers algorithmes : Euclide (3e s. av. JC)
- l'algèbre de Boole (1815-64)
- Le premier programmeur est une programmeuse : Ada Lovelace (1815-1852)

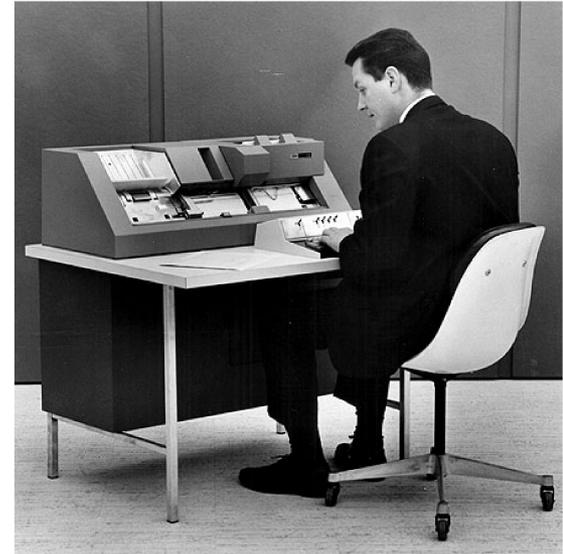


# Les générations d'ordinateurs

- Premier ordinateur : Z2 de Zuse (Berlin 1939), suivi de Eniac (Philadelphie 1943-46)
- Tubes à vide 1942
- Transistors, Shockley 1956
- Circuits imprimés 1963
- Circuits intégrés 1974
- Microprocesseurs 1974 (processeurs VLSI)

# Les modes d'utilisation : systèmes d'exploitation

- Machine « nue » (1950-60)
- Traitements par lots
  - Entrées sorties simultanées
  - Multiprogrammation
- Temps partagé (1970)
- Ordinateur individuel (1981)
- Calcul distribué sur grappes (2000)



# Les réseaux

- Les précurseurs
  - Télégraphe Chappe 1793
  - Télégraphe électrique 1840
- Réseaux à grande échelle
  - Arpanet 1969
  - France : Cyclades, puis Transpac 1979
  - Numéris 1987
- Réseaux locaux
  - Ethernet 1973,76
- Le “web” (internet + serveurs d’informations + hyperliens) : 1990

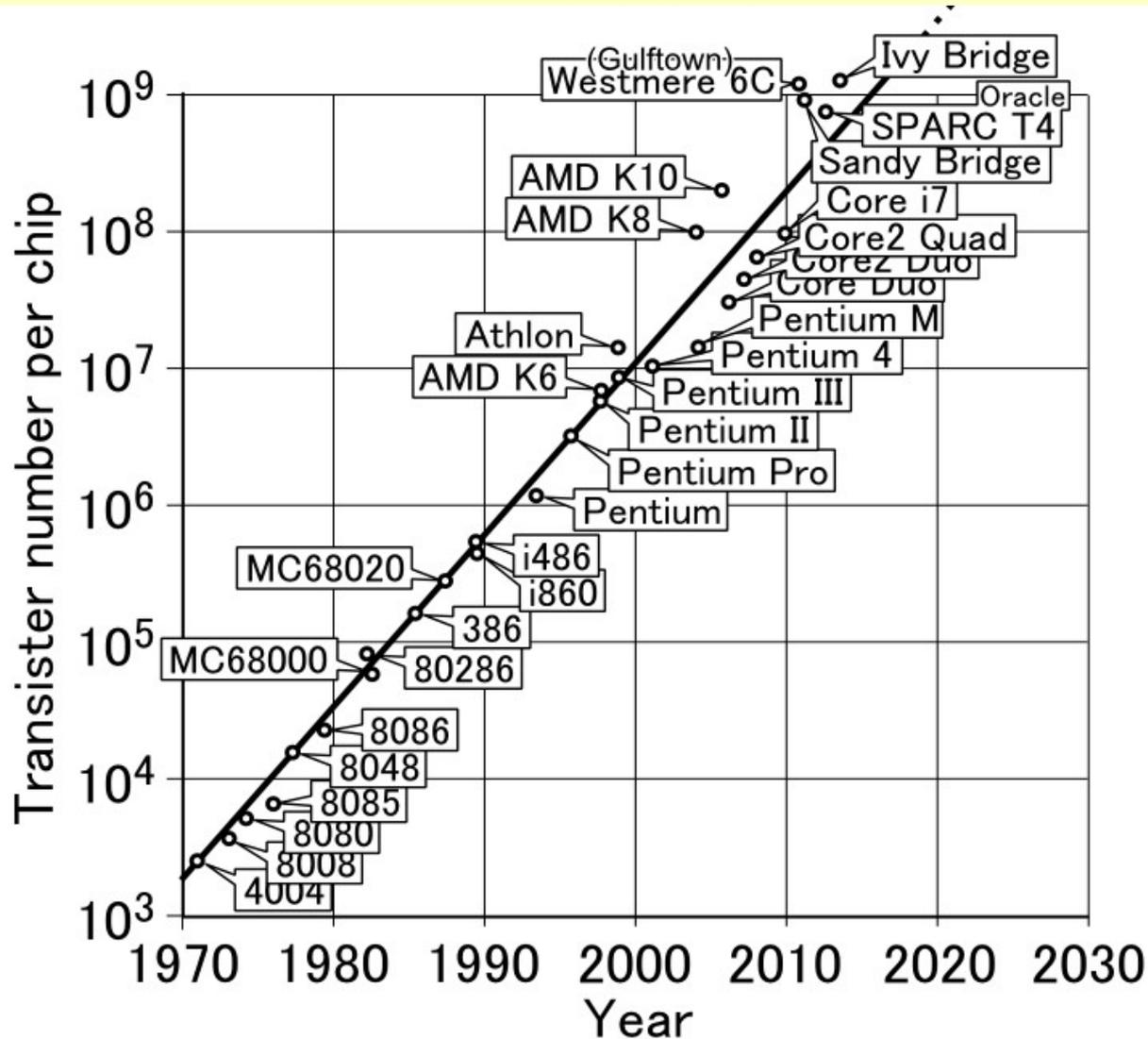
# Situation actuelle

- De l'informatique partout, y compris dans les parcmètres ! 1/3 du prix d'un Airbus !
- Tendance à l'interconnexion
  - Des ordinateurs
  - Des appareils domestiques
- Problèmes de sécurité, de confidentialité
- Des emplois variés (et intéressants) dans **tous** les secteurs: l'information est omniprésente

# Une explosion continue depuis 1960

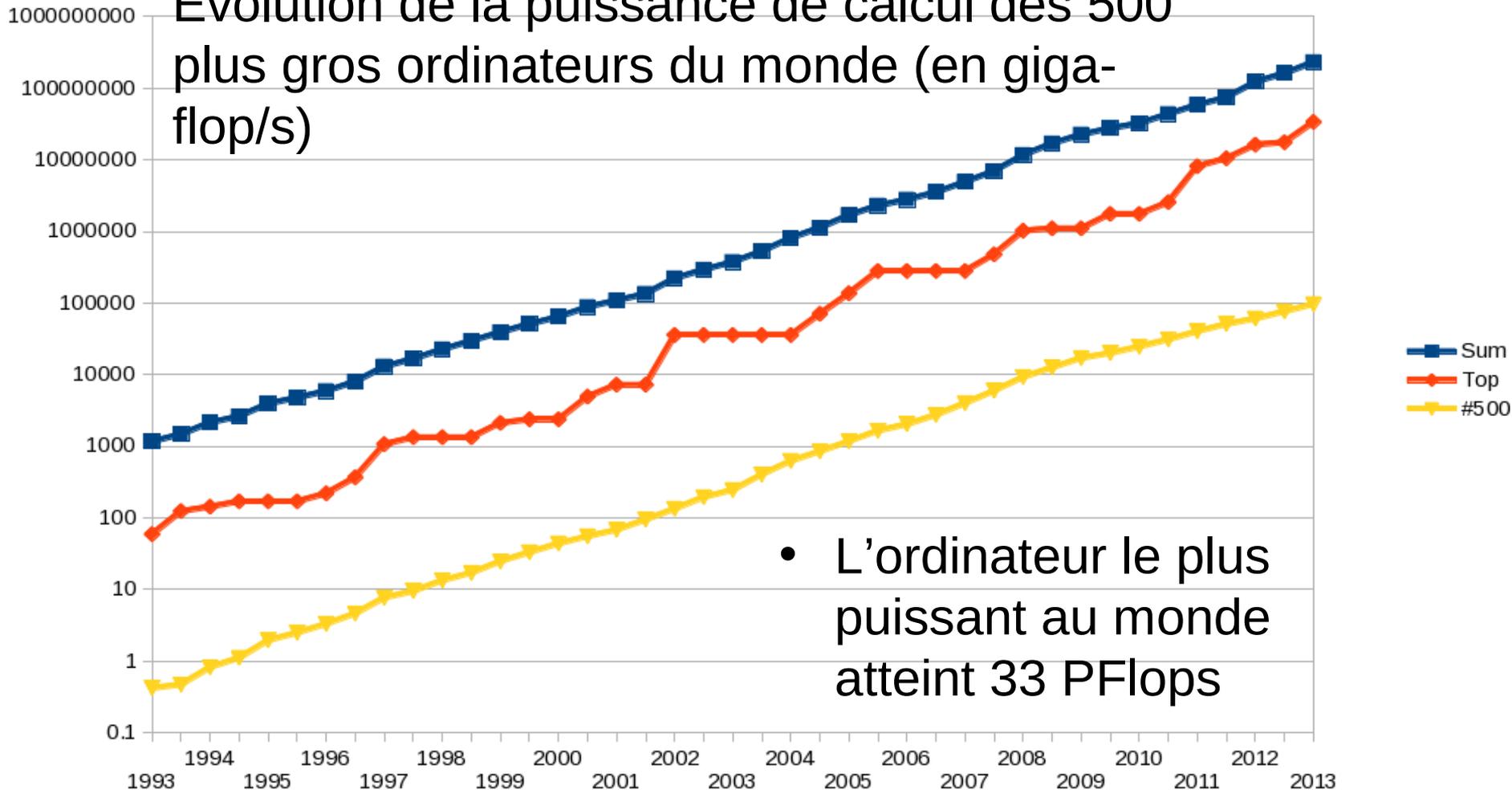
- Due à la puissance de calcul (typique:  $10^7$ - $10^8$  opérations/secondes) et qui continue de croître (loi de Moore: doublement du nombre de transistors tous les deux ans)
- Due au volume d'information manipulable (grosse machine:  $>10^{10}$  caractères (octets))
- Due à la vitesse des transmissions (autoroute:  $10^{10}$  bits/seconde)
- Due à la variété des applications: de la voiture (régulation de vitesse, diagnostique, trajet) aux finances (estimation de risque, prévision, gestion de comptes)

# Loi de Moore pour l'intégration de circuits



# Loi de Moore pour la puissance de calcul

Évolution de la puissance de calcul des 500 plus gros ordinateurs du monde (en giga-flop/s)



- L'ordinateur le plus puissant au monde atteint 33 PFlops

# Langages de programmation

- Calculer 2+2 pour un processeur Intel (langage machine, compréhensible par le processeur) :

```
B80200000083C002
```

➔ Bon, on ne va pas faire comme ça ...

- Le même programme, en langage d'assemblage (ou langage assembleur)

```
movl $2, %eax
```

```
addl $2, %eax
```

➔ Mieux, mais on ne va pas faire comme ça non plus !

- En Python :

```
>>> 2+2
```

```
4
```

# Le langage Python, introduction

- À la craie et au tableau !