



Méthodologie de la programmation

Chapitre 0 Présentation du cours



Pablo Rauzy <pr@up8.edu>
pablo.rauzy.name/teaching/mp

Présentation du cours

- ▶ Qu'est ce que l'informatique ?
- un outil,
 - une technologie,
 - une science.

- ▶ Qu'est ce que l'informatique ?
 - un outil,
 - une technologie,
 - une science.

- ▶ La programmation est au cœur de ces trois aspects.

- ▶ La *programmation* est l'activité centrale du *développement* de logiciel.
- ▶ Elle consiste à écrire le *code source* d'un *programme*.
- ▶ Ce code source est ensuite *interprété* ou *compilé* pour obtenir du code *exécutable*.

- ▶ **1842** Premier programme au monde.



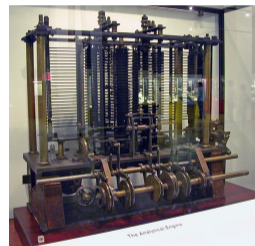
Ada Lovelace

- ▶ **1842** Premier programme au monde.



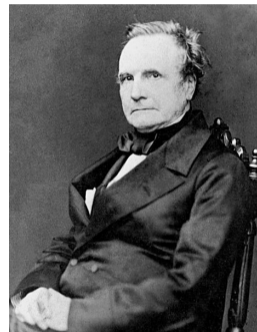
Cartes perforées

- ▶ **1842** Premier programme au monde.



Machine analytique

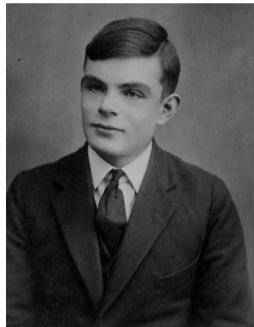
- ▶ **1842** Premier programme au monde.



Charles Babbage

Aperçu historique

- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".



Alan Turing

Aperçu historique

- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".
- ▶ **1952** Premier compilateur (A-0).



Grace Hopper

- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".
- ▶ **1952** Premier compilateur (A-0).
- ▶ **1950s** Premiers langages haut niveau (Fortran, Lisp).



John Backus

Aperçu historique

- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".
- ▶ **1952** Premier compilateur (A-0).
- ▶ **1950s** Premiers langages haut niveau (Fortran, Lisp).



John McCarthy

Aperçu historique

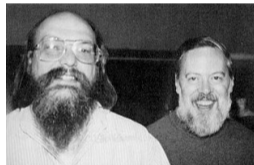
- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".
- ▶ **1952** Premier compilateur (A-0).
- ▶ **1950s** Premiers langages haut niveau (Fortran, Lisp).
- ▶ **1964** BASIC.



John Kemeny
Thomas Kurtz

Aperçu historique

- ▶ **1842** Premier programme au monde.
- ▶ **1936** Turing publie "*On Computable Numbers with an Application to the Entscheidungsproblem*".
- ▶ **1952** Premier compilateur (A-0).
- ▶ **1950s** Premiers langages haut niveau (Fortran, Lisp).
- ▶ **1964** BASIC.
- ▶ **1972** C.



Denis Ritchie

► Objectifs :

- Apprendre à (apprendre à) programmer.
- Connaître les outils utiles pour programmer.
- Voir un peu de théorie de la programmation.
- Acquérir un socle solide de culture générale sur la programmation.

→ Avoir les moyens de suivre le plus efficacement possible vos études d'informatique.

Langages et paradigmes de programmation

- ▶ Aborder plusieurs *langages* et différents *paradigmes* de programmation.

Langages et paradigmes de programmation

- ▶ Aborder plusieurs *langages* et différents *paradigmes* de programmation.
- ▶ Paradigmes de programmation :
 - concurrente,
 - déclarative,
 - fonctionnelle,
 - impérative,
 - logique,
 - objet,
 - par contrainte,
 - synchrone,
 - événementielle,
 - ...
- ▶ Langages de programmation :
 - Assembleur,
 - Bash,
 - C,
 - C++,
 - CSS,
 - Erlang,
 - HTML,
 - Java,
 - LaTeX,
 - Lustre,
 - OCaml,
 - PHP,
 - Prolog,
 - Python,
 - Racket,
 - SQL,
 - ...

- ▶ Ce cours aborde l'informatique au travers de la programmation dans différents langages :
 - Bash
 - Python
 - L^AT_EX
 - C

- ▶ Chacun des langages que nous allons voir vient avec son lot d'outils que nous apprendront également à utiliser :
 - interpréteurs,
 - compilateurs,
 - moteurs de production,
 - gestionnaires de contrôle de version,
 - gestionnaire de paquets,
 - débogueurs,
 - profileurs,
 - documentations.

- ▶ Votre évaluation pour ce cours prendra en compte (dans l'ordre d'importance) :
 - le projet,
 - les TPs (potentiellement tous les TPs seront notés !)
 - un exam écrit sur la partie théorique.

- ▶ La propreté du code (nommage, indentation, organisation) est importante est sera prise en compte autant pour les TP que pour le projet.

Ce qui est attendu de vous

- ▶ Correctement écrire du code :
 - indentation propre,
 - nommage cohérent,
 - organisation modulaire des fichiers.

- ▶ Savoir gérer un projet seul ou à plusieurs :
 - utiliser git pour gérer les versions du code source et la collaboration,
 - utiliser Make pour gérer la compilation modulaire.

- ▶ Savoir utiliser les bons outils :
 - connaître les forces et faiblesses de différents langages et paradigmes de programmation,
 - savoir chercher efficacement dans une documentation,
 - utiliser un débogueur voire un profileur.

- ▶ Savoir communiquer sur vos projets :
 - utiliser $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pour écrire des rapports.