

# Algorithmique et Programmation (TALA330A L3 INALCO)

Crédits : G. Galisot - Université de Tours

---

Caroline Parfait  
caroline.parfait@sorbonne-universite.fr

2021-2022

Obtic, Sorbonne Center for Artificial Intelligence  
STIH EA 4509, Sorbonne Université

# Plan de la présentation

1. Un peu d'histoire
2. Programmation : Suivre la recette
3. Langages de programmation et Environnements

## Un peu d'histoire

---

Le mot algorithme provient du nom latinisé du mathématicien perse **Al-Khawarizmi** (780-850).

- Origine du mot algèbre :  
*Abrégé du calcul par la restauration et la comparaison*
- Utilisation des chiffres arabes au Moyen-Orient et en Occident :  
*Traité du système de numération des Indiens*

# Qu'est ce que l'algorithmique ?

## Résolution informatique d'un problème

- Le problème (la tâche à résoudre)
- L'algorithme (la méthode de résolution)
- Le langage de programmation (langage dans lequel est écrit le programme)
- L'ordinateur (la machine qui exécute le programme)

# Histoire de la programmation

- 1801 : le métier à tisser de Jacquard (machine dont les possibilités changent quand on modifie son programme)

**Figure 1:** Métier à tisser de Joseph-Marie Jacquard.



- 1834 : La machine analytique (analytical engine en anglais) de Charles Babbage, mathématicien anglais.

# Frame Title

- 1842-1843 : Ada Lovelace traduction du traité sur la machine de Babbage.
- 1843 + : Ada réalise le premier véritable programme informatique.

**Figure 2:** «Programme» de calcul des nombres de Bernoulli dans la note **G** d'Ada Lovelace (1843).

Diagram for the responses by the Engines of the Stratons at Bermuda. See Table G, page 277 of app.

# Histoire de la programmation

- 1896 + : Herman Hollerith fonde la *Tabulating Machine Co.*
- 1917 : « IBM »3 *l'International Business Machines Corporation* .
- 1896 + : Amélioration des machines d'IBM développement de la carte perforée d'IBM3.

**Figure 3:** Carte Hollerith.

1	2	3	4	5	6	7	8	9	10	On	S	A	C	E	a	c	e	g		EB	SB	Ch	Sy	U	Sh	Hk	Br	Rm
2	2	4	1	3	E	15	Off	IS	B	D	F	b	d	f	h					SY	X	Fp	Cn	R	X	Al	Cg	Kg
3	0	0	0	0	W	20			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	1	1	1	1	0	25	A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	2	2	2	2	5	30	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
C	3	3	3	3	0	3	C	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	4	4	4	4	1	4	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
E	5	5	5	5	2	C	E	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
F	6	6	6	6	A	D	F	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
G	7	7	7	7	B	E	G	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
H	8	8	8	8	A	F	H	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
I	9	9	9	9	b	c	I	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

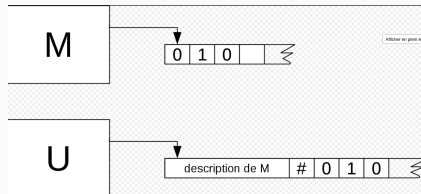


# Histoire de la programmation

- 1936 : Alan Turing, publication de l'article fondateur de la science informatique *On Computable Numbers with an Application to the Entscheidungsproblem*
- 1936 : La machine de Turing, premier calculateur universel programmable.

Introduction des concepts de programmation et de programme.

**Figure 4:** Fonctionnement de la machine de Turing



# Histoire Technique de la programmation

- Premiers programmes = fer à souder + tubes à vide (transistors).
- Une seule erreur = le programme entier inutilisable.
- Progrès des supports de données = cartes perforées la liste des instructions en code binaire
- OUI MAIS spécifique à un type d'ordinateur particulier.
- Les langages de programmation = du texte pour suites de 0 et de 1
- Les langages de programmation font de plus en plus abstraction du matériel sur lequel devraient tourner les programmes.
- Gains :
  - ces langages sont plus faciles à apprendre
  - un programmeur peut produire du code plus rapidement
  - Les programmes peuvent tourner sur différents types de machines.

# **Programmation : Suivre la recette**

---

# Un Algorithme est une Recette

- Définition : Un algorithme est une suite finie et non ambiguë d'opérations permettant de donner la réponse à un problème.
- Exemples :
  - Faire un gâteau au chocolat
  - Changer une roue
  - Résoudre un système d'équations différentielles
  - Trier des informations selon une clé donnée
  - ...

# Un Algorithme est une Recette

## Ingrédients, outils, procédure



200 g de farine



30 g de sucre



3 oeufs



20 g de beurre



25 cl de lait



1 pincée de sel



1 appareil à gaufres



1 saladier



1 fouet

### Etape 1

Mettre la farine dans un saladier, y ajouter les jaunes d'œufs et le beurre ramolli.

### Etape 2

Délayer peu à peu le tout en y ajoutant le lait. Il ne doit pas y avoir de grumeaux.

### Etape 3

Battre les blancs en neige avec une pincée de sel, puis ajouter au restant en remuant délicatement.

### Etape 4

Cuire le tout dans un gaufrier légèrement beurré.

# Un Algorithme est une Recette

- Exercice : Écrivez la marche à suivre permettant d'accrocher un tableau au centre d'un mur pour cela,
- vous devez :

# Un Algorithme est une Recette

- Exercice : Écrivez la marche à suivre permettant d'accrocher un tableau au centre d'un mur pour cela,
- vous devez :
  - Définir les objets nécessaires à la résolution du problème

# Un Algorithme est une Recette

- Exercice : Écrivez la marche à suivre permettant d'accrocher un tableau au centre d'un mur pour cela,
- vous devez :
  - Définir les objets nécessaires à la résolution du problème
  - Établir la liste des opérations à effectuer



# Un Algorithme est une Recette

- Exercice : Écrivez la marche à suivre permettant d'accrocher un tableau au centre d'un mur pour cela,
- vous devez :
  - Définir les objets nécessaires à la résolution du problème
  - Établir la liste des opérations à effectuer
  - Ordonner cette liste

# Propriétés d'un algorithme

- Avoir un nombre fini d'étapes
- Avoir un nombre fini d'opérations par étape
- Se terminer après un nombre fini d'opérations
- Fournir au moins un résultat
- Chaque opération doit être :
  - définie rigoureusement et non ambiguë
  - effective, c'est-à-dire réalisable par une machine
- Un algorithme est indépendant du langage de programmation utilisé

# **Langages de programmation et Environnements**

---

- Définition
  - L'algorithme ou code source est exprimé en texte humain
  - Le code source pour être analysé par la machine subit une transformation  
= un programme ou code exécutable
- Objectif
  - Faire abstraction des mécanismes de **bas niveau** de la machine  
= le code source est écrit et compris par un être humain

- La programmation
  - Activité de rédaction du code source d'un programme.
  - Mise en œuvre de techniques d'écriture
  - Résolution d'algorithmes informatiques
- Un langage de programmation est composé :
  - **Vocabulaire** : Symboles/termes utilisés pour construire des expressions/opérations
  - **Syntaxe** : Règles définissant la bonne construction des expressions
  - **Sémantique** : Règles permettant d'associer un sens à une expression (signification d'une expression)

# Langage machine (bas niveau)

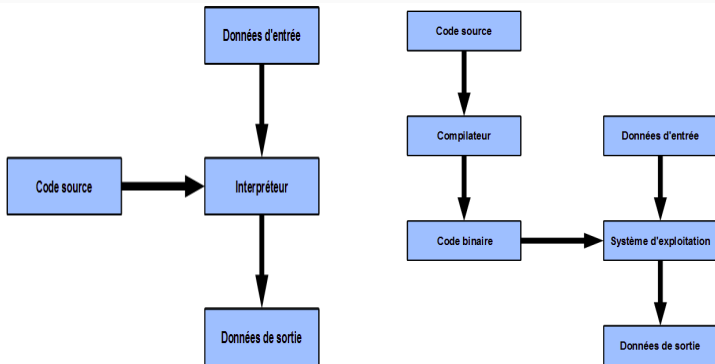
- Un ordinateur ne comprend que le langage machine
- Comment traduire un code source en langage machine ?
- Plusieurs méthodes :
  - Interprétation
  - Compilation
  - Méthodes intermédiaires
- Langage Machine :
  - Langage natif d'une famille de processeurs
  - Composé d'instructions et de données à traiter codées en binaire
  - Chaque processeur possède son propre langage machine donc un code machine ne peut s'exécuter que sur la famille de machine pour laquelle il a été préparé.

- Un programme écrit en langage interprété est converti en instructions **directement exécutables** par la machine au moment de son exécution.
- **Interpréteur** : programme qui traduit le langage de programmation en langage machine.
- **Interpréteur** : il doit être en fonctionnement sur la machine où l'on veut lancer un programme interprété.

- Le programme est traduit **une fois pour toute en code machine** pour être **ensuite exécuté** sur la machine cible
- Le code machine obtenu est appelé exécutable (.exe)
- Un compilateur est un programme informatique qui traduit un programme en langage source vers le langage machine (cible)
  - Le langage source est de **haut niveau** = destiné à l'humain
  - Le langage cible est de **bas niveau** = destiné à la machine



# Langages Interprétés vs Compilés



(a) Langage Interprété

(b) Langage Compilé

**Figure 5:** Fonctionnement des langages Interprétés et Compilés.

<http://www.france-ioi.org/algo/course.php?idChapter=561&idCourse=2368>

# Langages Interprétés vs Compilés

	Interprétés	Compilés
Langages	Python, Shell, BASIC, PHP, Javascript, Ruby, Perl	C, C++, Java
Avantages	simplicité  Code source compatible sur chaque machine ayant un interpréteur	rapidité et optimisation du code
Inconvénients	lenteur	complexité, Code machine spécifique à chaque type de machines et SE

# Environnement de développement intégré - EDI

IDE en anglais, pour *integrated development environment*

- Ensemble d'outils destinés à programmer dans un langage donné<sup>1</sup>
- Fonctionnalités primaires : Analyser, écrire et déboguer un programme.
- Un IDE comporte :
  - une interface graphique pour lancer les différents outils,
  - un éditeur pour le code source,
  - un compilateur,
  - un débogueur : permet d'exécuter le programme ligne par ligne et vérifier l'état des variables.

environnements de développement populaires : Visual Studio, Spyder, Eclipse et IntelliJ IDEA, etc.

1. Les EDI sont souvent dédiés à un langage donné, même si certains peuvent gérer plusieurs langages (Visual Studio).

- Qu'avez-vous retenu ?
- Quelles différences entre algorithme et langage de programmation ?
- Quelles différences entre langages compilés et interprétés ?

Ce support de cours a été construit à partir des sources suivantes :

- Support de cours Licence 1 « Algorithmique et langages du Web » UE 102 - 2011-2012. Auteur : Thomas Devogèle
- Informatique et sciences du numérique. Spécialité ISN en Terminal S. Gilles Dowek. Eyrolles.
- Site OpenClassrooms.com. Tutoriel Python. Sept 2013  
[http ://fr.openclassrooms.com/informatique/python/cours](http://fr.openclassrooms.com/informatique/python/cours)
- Site Débuter avec Python au lycée. Sept 2013.  
[http ://python.lycee.free.fr/](http://python.lycee.free.fr/)
- Gérard Swinnen Apprendre Python 3 -  
[http ://inforef.be/swi/python.htm](http://inforef.be/swi/python.htm)
- Cours sur Python 3 - Robert Cordeau
- M-Anne Moreaux, Licence LLCER, Parcours TAL – 2020/2021
- Wikipédia ...