

# Enseigner la pensée informatique

*Et puis quoi encore ?*

V. Englebert, Prof.

SETT'2019 @Namur

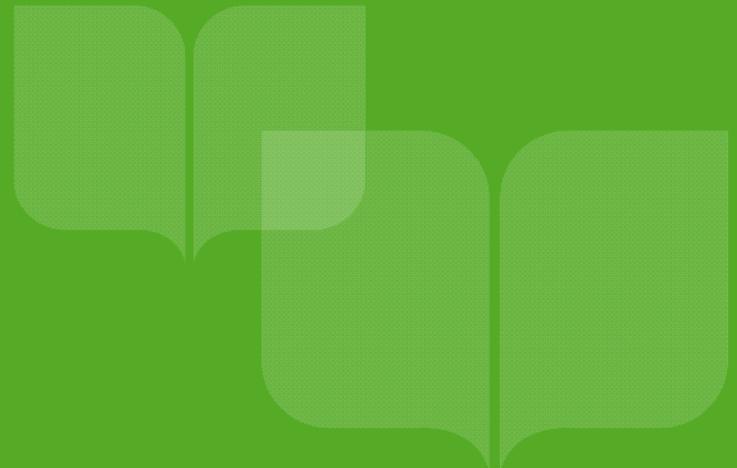
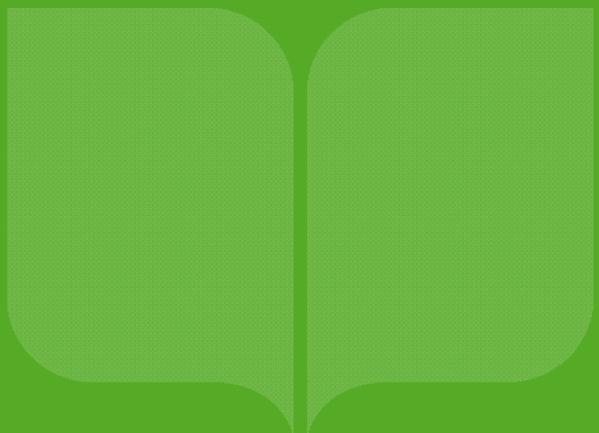


FACULTÉ  
D'INFORMATIQUE

# Thèse

Les élèves devraient être confrontés à la « Pensée Informatique » dès le plus jeune âge, à chaque étape pertinente de l'enseignement obligatoire par des cours spécifiques.





Mais pourquoi ?

## **1) Perspectives Professionnelles**

# L'emploi ?

- 373.000 postes d'informaticiens à pourvoir en Europe en 2015.
- **530.000 postes informaticiens en 2020 !**
  - 226.000 dans la gestion des TIC
- Le nombre d'offres augmentait de 3% par an et est maintenant de **10% depuis 4 ans !**
- Emploi garanti avant la fin des études pour les universitaires depuis ... très très longtemps.

## FOR EUROPE, A POTENTIAL OF MORE THAN 750,000 NEW ICT JOBS UNTIL 2020

The demand for e-skills keeps growing at a tremendous pace. In 2020 the European labour market is projected to grow by more than 670,000 new jobs, but it could absorb another 756,000 ICT practitioners if only sufficient supply were in sight.

For Europe, a potential of more than 750,000 new ICT jobs until 2020

The demand for e-skills keeps growing at a tremendous pace. In 2020 the European labour market is projected to grow by more than 670,000 new jobs, but it could absorb another 756,000 ICT practitioners if only sufficient supply were in sight.

These are the key messages of a recent working paper by empirica which published research results showing that demand for e-skills keeps growing at a tremendous pace. The trend in the number of core ICT jobs has been steady for more than a decade, with growth rates averaging at 3% p.a., and the average annual growth in ICT management and analysis jobs even reaching 10% in the last four years. At the same time demand for medium level skilled jobs is declining. As a net effect, Europe has added a significant number of new jobs in ICT year after year. Despite the crisis, vacancy rates in ICT remain high and there is a need to continuously increase the quality and the relevance of e-skills. As graduate figures stagnate, supply from universities does not seem to keep pace.

For the first time empirica analysed a massive set of online vacancy data provided through jobfeed ([www.jobfeed.com](http://www.jobfeed.com)) to estimate the number of open posts for ICT professionals. Based on this big data analysis of job postings in Europe, empirica is now in a position to estimate vacancy figures on a much broader basis than with previous approaches which were survey based. Jobfeed has been developed by Textkernel BV ([www.textkernel.com/](http://www.textkernel.com/)) as a Big Data tool for jobs and provides a database of real time and historical online job data.

Together with Jobfeed, empirica analysed the online job postings that mapped to the definition of ICT jobs and identified 373,000 open posts in October 2015. The excess demand or shortage (calculated as the number of open posts) according to our projections will reach 756,000 in 2020. This figure can best be described as 'demand potential' or 'job potential' for ICT jobs. It should be seen as a hypothetical figure describing the demand potential for new ICT jobs which could additionally be created in Europe due to e-skills demand likely to occur in the next five years until 2020 if demand were not hampered by supply bottlenecks. Of these 756,000 there are 530,000 potential additional jobs in ICT practitioner occupations and around 226,000 at ICT management level. Source: <http://eskills-lead.eu/news/single-view/for-europe-a-potential-of-more-than-750000-new-ict-jobs-until-2020.html>



## Emplois IT par Fonction ▼

Développeur 377

Analyste Programmeur 365

System Engineer 191

Business Analyst 163

Functional Analyst 159

Consultant informatique 149

Helpdesk 137

Process Analyst 115

Chef de projet informatique 111

Technical Analyst 107

Maintenance Engineer 103

Team Leader 86

Application Architect 69

Project Coordinator 65

Operator 60

Data Analyst 59

Infrastructure Architect 58

Network Engineer 51

Software Administrator 47

Security Engineer 46

## Emplois IT par Compétence ▲



### Security Analyst / Consultant IAM

Approach Belgium

□ 02/04 Wallonie, Flandre, Bruxelles

IAM PAM Azure security SAML OAuth



### EasyVista Integration Expert

Mielabelo

□ 09/04 Wallonie

EasyVista, solution, integration



### Responsable de la Sécurité des Systèmes d'Information

afarax

□ 26/03 Wallonie

Sécurité



### Support Eng

EyeTech Solutions

□ 11/04

support enginner



### Data Scientists

Laco

□ 26/03 Anvers, Brabant Wallon, Liège, Bruxelles, Gantois

Python R SAS data scientist



### Business Analyst Energy Market

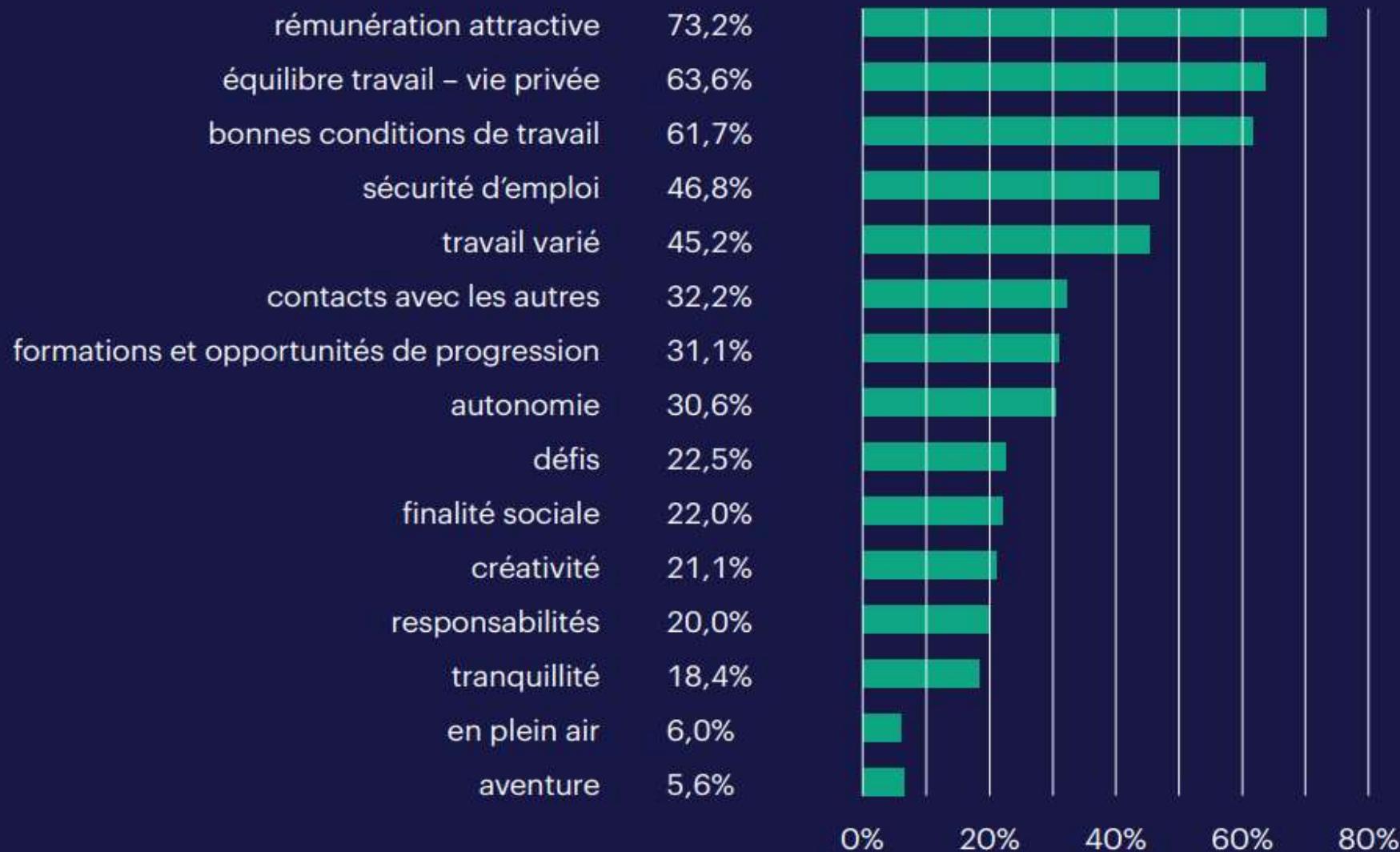
Altran Belgium

□ 19/04 Bruxelles, Malines, Liège

Business Analyst

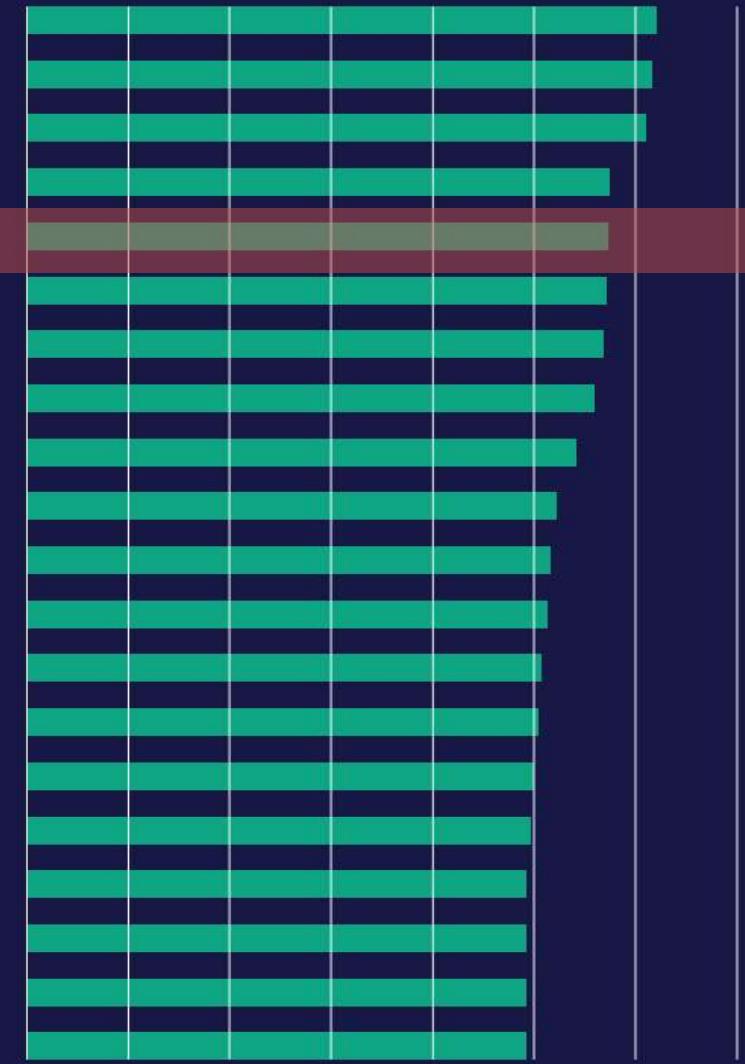
ICT Jobs en FWB  
Temps  
plein/CDI/Master+5  
(22/04/2019)

# L'importance des critères



# Attractivité

		2018
1	ingénieur civil/industriel	62,0%
2	pharmacien	61,6%
3	médecin	60,9%
4	architecte	57,3%
5	ingénieur informaticien	57,2%
6	vétérinaire	57,0%
7	chirurgien	56,8%
8	professeur d'université	55,9%
9	notaire	54,1%
10	kiné	52,1%
11	professeur*	51,6%
12	pharmacologue	51,2%
13	dentiste	50,6%
14	avocat	50,3%
15	expert juridique	49,9%
16	web designer	49,6%
17	support IT	49,2%
18	biotechnologue	49,2%
19	fonctionnaire**	49,1%
20	traducteur	49,1%



Source: Randstad « l'attractivité des professions en Belgique » Janvier 2019

# Les critères

		salaire	équilibre travail-privé	conditions de travail	variation	défi
1	ingénieur	8	29	7	27	15
2	pharmacien	15	19	3	86	96
3	médecin	4	103	55	11	10
4	architecte	11	34	9	5	8
5	ingénieur inormaticien	18	46	24	39	24
6	vétérinaire	14	91	51	10	12
7	chirurgien	6	108	54	16	2
8	professeur d'université	9	17	5	55	37
9	notaire	1	18	1	43	72
10	kiné	27	58	20	44	63

Source: Randstad « l'attractivité des professions en Belgique » Janvier 2019

# Au-delà de l'emploi

- ❑ Le manque d'informaticiens **grève l'innovation scientifique et technologique**
- ❑ La carence d'informaticiens handicape gravement le **développement économique**

## Par contre...

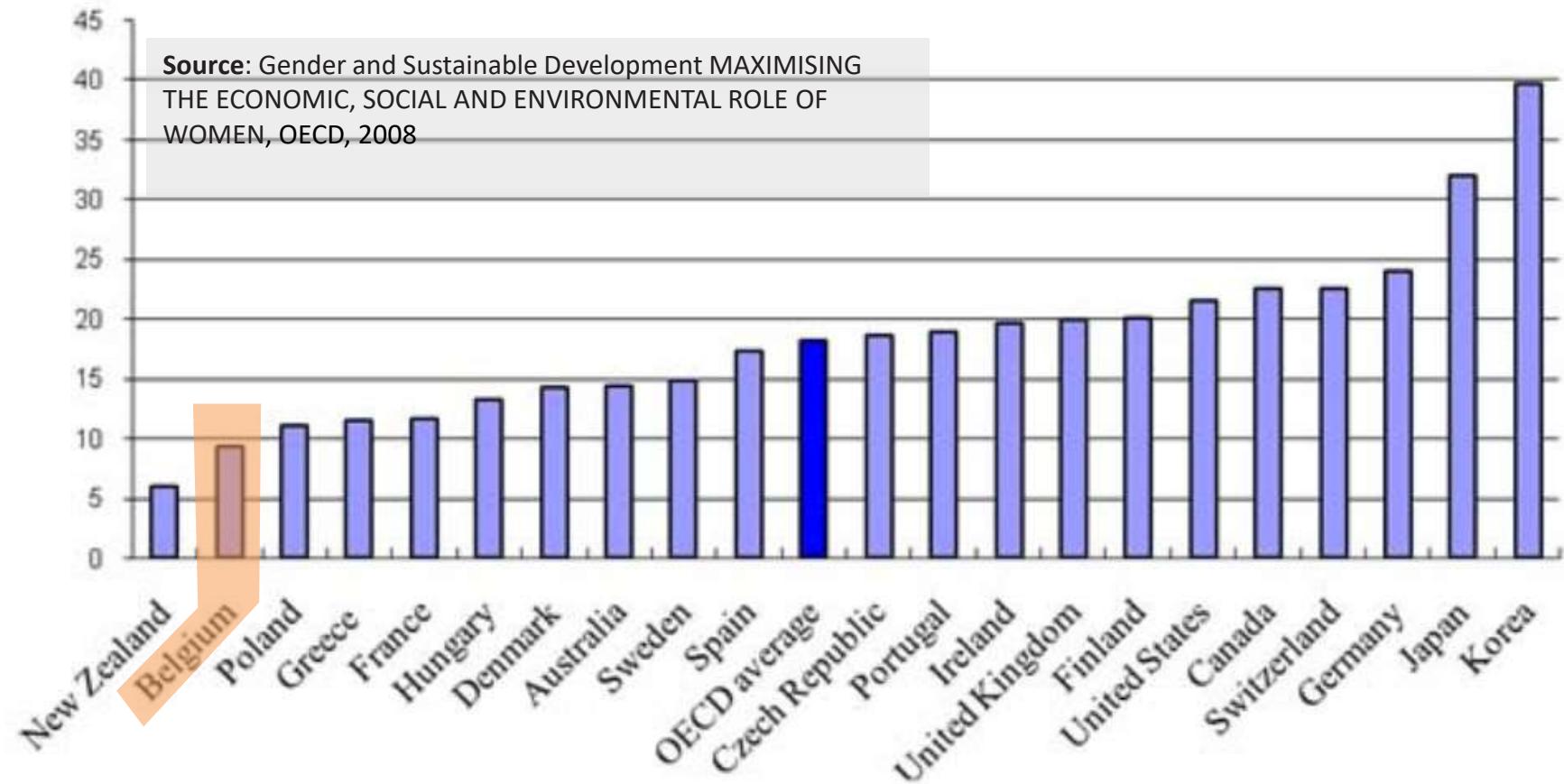


dans le top 10, la moitié des professions, tout au plus, sont encore clairement masculines : ingénieur, ingénieur informaticien, architecte, chirurgien et notaire.

Source: Randstad « l'attractivité des professions en Belgique » Janvier 2019

## **2) Fracture générée**

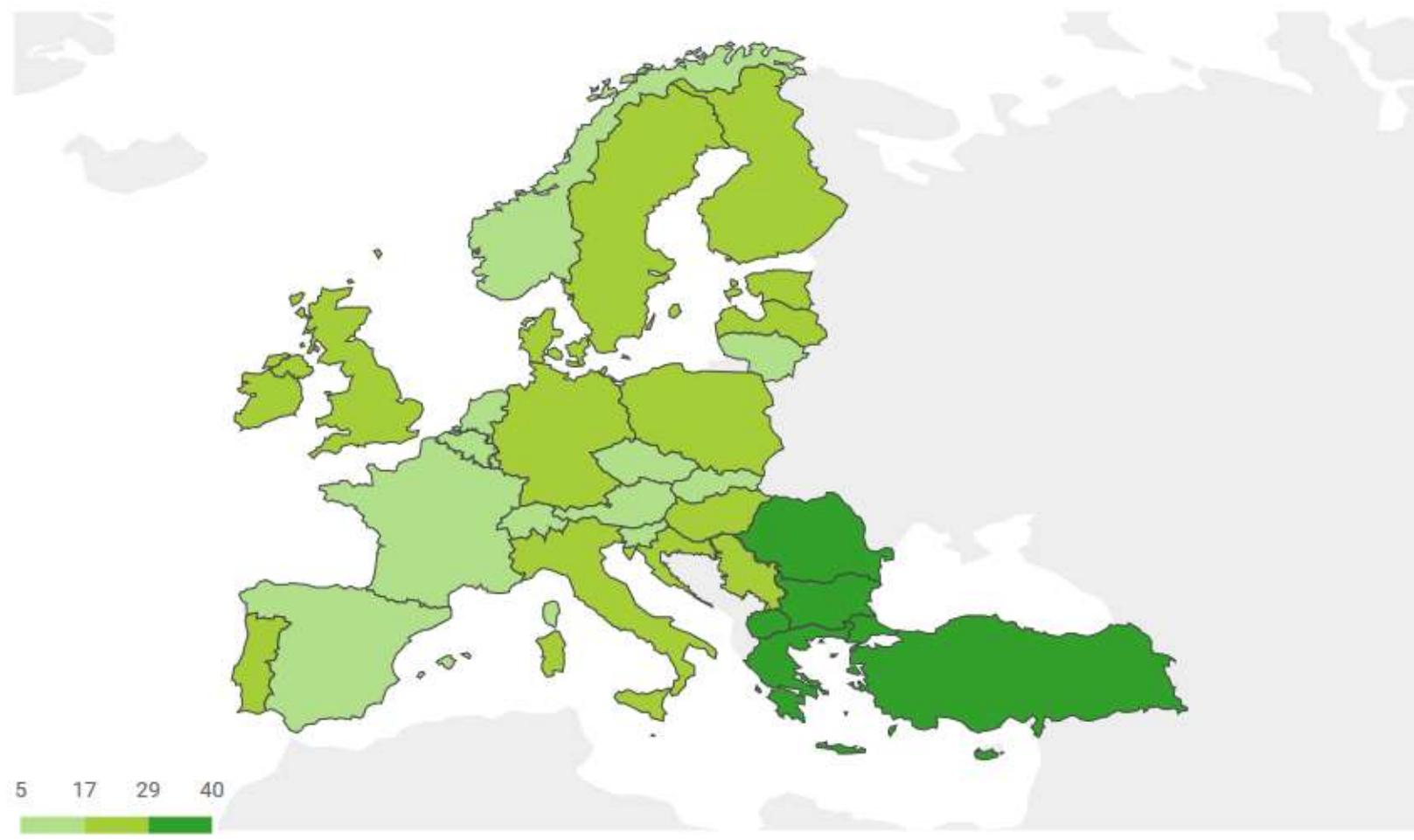
# L'informatique fait « mauvais genre »...



Dans les emplois ICT

## Women lag behind in ICT tertiary education

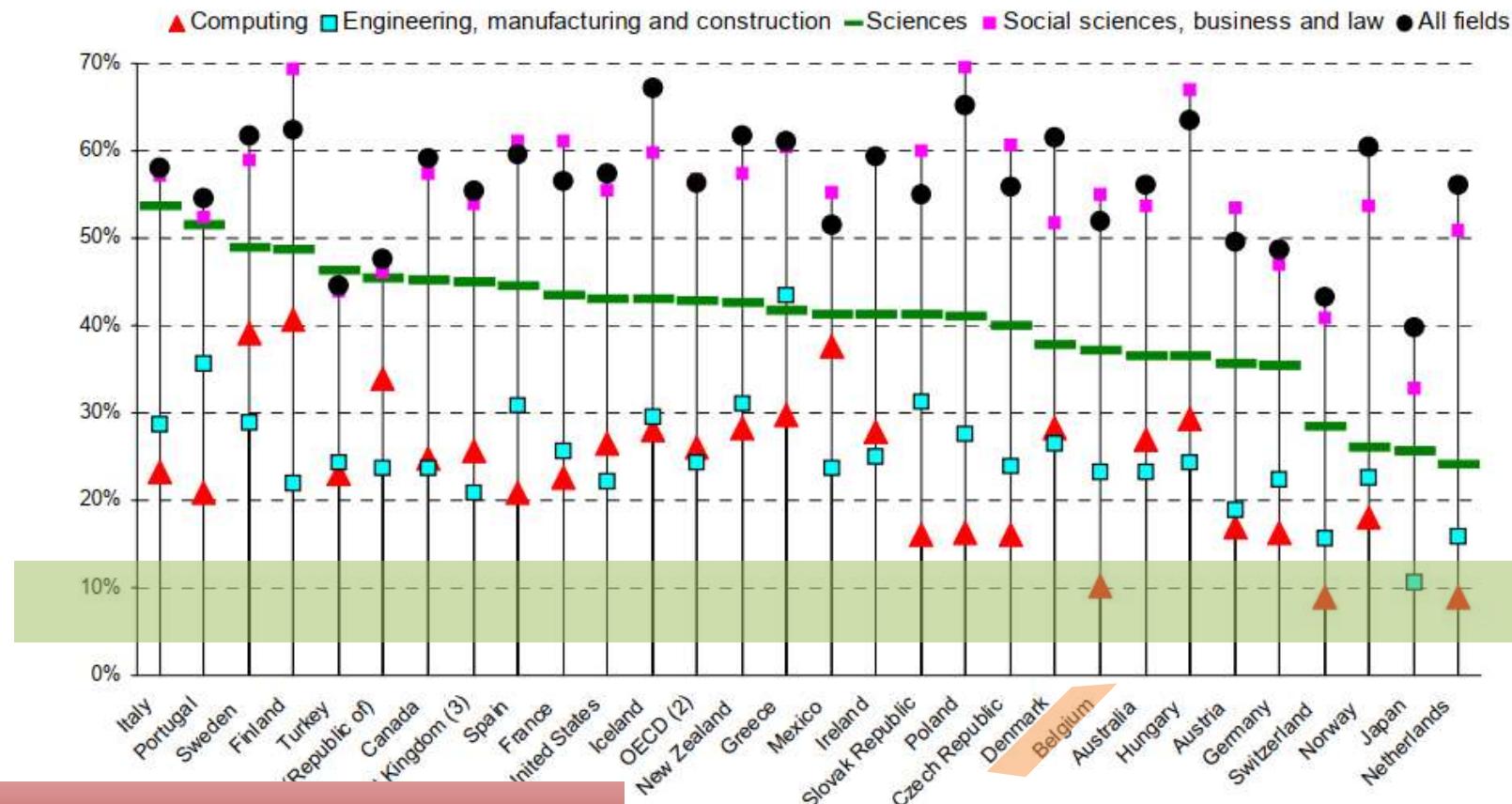
Percentage of women graduates in ICT tertiary education programmes according to the latest available data.



# L'informatique fait « mauvais genre »...

**Figure 5. Share of Female Graduates by Field**

2004 or most recent year



Dans les études ICT

Source: Gender and Sustainable Development MAXIMISING THE ECONOMIC, SOCIAL AND ENVIRONMENTAL ROLE OF WOMEN, OECD, 2008

### **3) Une compétence universelle**

# Une compétence universelle

- L'informatique est une discipline scientifique à part entière, au même titre que la chimie, la biologie, la physique ou les mathématiques.
  - Sa propre histoire
  - Des communautés fortes et distinctes des autres
  - Des objets d'étude intemporels
  - Qui contribue, comme et avec les autres disciplines, à la connaissance scientifique

## **4) Programmer ou être programmé**

Rendre la justice plus efficiente grâce à l'intelligence artificielle

CONTRIBUTION EXTERNE Publié le samedi 13 avril 2019 à 17h12 - Mis à jour le samedi 13 avril 2019 à 17h14

OPINION

Franceinfo:

vidéos | radio | jt | magazines | DIREC

20 WEEK-END



Du vendredi au dimanche à 20h

Faits-divers / Police

## Police : l'intelligence artificielle traquer la délinquance

Un logiciel de prédition d'éventuels délits est utilisé par la gendarmerie depuis u ffres de la criminalité ba

L'IA AU COEUR DE L'HUMAIN

## Intelligence artificielle: vers une autre médecine

Par Eric Favereau — 14 janvier 2019 à 11:42



Les Echos

À la une Idées Économie Élections Politique Monde Tech-Médias Entreprises Bourse Finance - Marchés Régions

Partager



2018 01:00



A lire également

SIEN Agenda du jeudi 25 avril 2019

Pourquoi Umicore trinque encore après sa dégringolade

Perquisition d'entraînement à l'Union belge blanchiment

## IA : faire des décideurs d'aujourd'hui les leaders de demain

Dès 2014, Deep Knowledge Ventures, une société de gestion basée à Hong Kong, a fait l'actualité en plaçant une Intelligence Artificielle à son board pour recommander ou écarter des investissements. Si ce phénomène précurseur a intéressé les médias, il reste toutefois isolé. En France, à quel stade en sommes-nous quant à l'intégration de l'IA dans la stratégie des entreprises ?



EXPRESS BUSINESS HOME POLITIQUE SCIENCE ECONOMIE TRAVEL & FOOD LIFESTYLE TECH

Le dilemme du Tramway : achèterez-vous la Tesla égoïste, ou la Tesla altruiste ?

Dominique Dewitte — 26 mars 2019 2 minutes read



EXPRESS BUSINESS HOME POLITIQUE SCIENCE ECONOMIE TRAVEL & FOOD LIFESTYLE TECH

Le dilemme du Tramway : achèterez-vous la Tesla égoïste, ou la Tesla altruiste ?

Dominique Dewitte — 26 mars 2019 2 minutes read



En Estonie, une intelligence artificielle va rendre des décisions de justice

Par Harold Grand — Mis à jour le 05/04/2019 à 15:57 / Publié le 01/04/2019 à 18:20



LE FIGARO PREMIUM

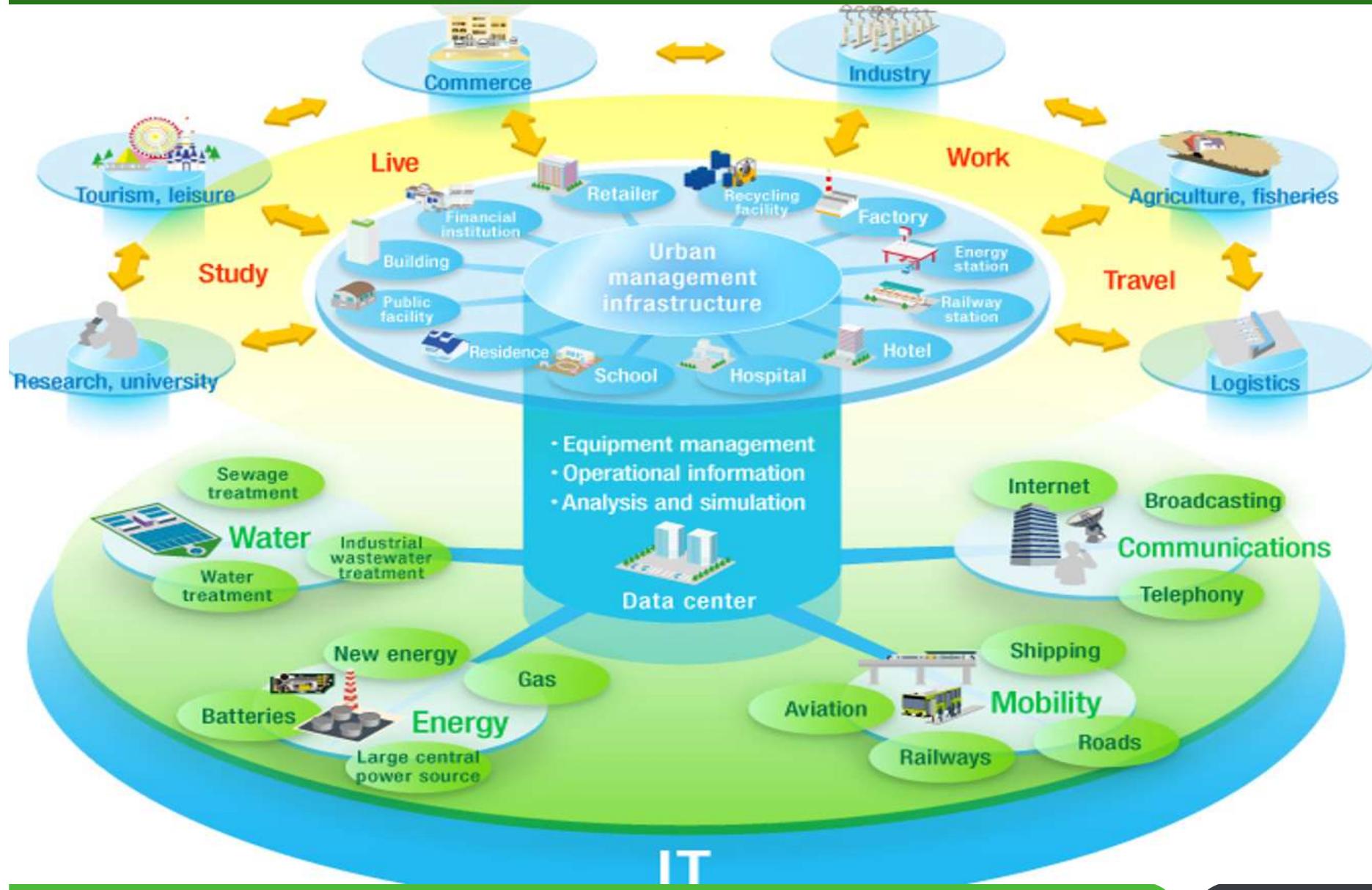
> 1€ le premier mois

89 commentaires



Le gouvernement estonien développe une intelligence artificielle qui devra

# L'IT au cœur de nos vies





# La chasse aux vilains canards

# Canard n°1

- « *On n'enseigne pas la mécanique alors que nous conduisons tous une voiture. Pourquoi enseigner l'informatique ?* »
  
- On enseigne le latin, la physique, et l'histoire alors que nous ne deviendrons pas tous latinistes, physiciens ou historiens.



# Canard n°2

□ « Il est inutile d'apprendre l'informatique, demain, les ordinateurs programmeront à notre place »



DEMAIN SERA VERTIGINEUX, PAR  
**LAURENT ALEXANDRE**

## Apprendre à coder ? Non !

**L**e code informatique est la base de la révolution informatique. Doit-on généraliser son apprentissage ? La question est bien difficile, tant il existe un consensus pour répondre oui... C'est en apparence, une proposition logique et pleine de bon sens qui est en réalité absurde. Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former : a-t-on envisagé que 100 % des jeunes deviennent électriciens en 1895, lorsque l'électricité s'est généralisée et est devenue la base de la deuxième révolution industrielle ?

Avoir une culture générale informatique est, bien sûr, essentiel pour être un citoyen au fait des enjeux numériques et capable de participer au débat politique sur le monde qui vient. Mais les plans de formation massifs de développeurs informatiques bac + 2 conduiront à beaucoup de désillusions. Le code informatique bas de gamme sera entièrement automatisé grâce à l'intelligence artificielle (IA), qui sera quasi gratuite. Ainsi, le patron de Nvidia, l'un des plus importants fabricants de microprocesseurs dédiés à l'IA, explique : « L'IA va manger le logiciel. »

**B**ien sûr, il y a un immense besoin d'architectes informatiques et de *data scientists*. Mais ces métiers ne seront accessibles qu'à des esprits multidisciplinaires de très haut niveau. Codeur informatique bas de gamme sera un passeport pour Pôle emploi ! Le spécialiste de l'IA Serge Abiteboul résume, dans *Le Monde*, l'horrible réalité d'une phrase : « Il faut bien sûr pour politiquer être correcte ». Data scientist est un métier qui demande énormément de neurones. « La métaphore de la langue étrangère (le code serait la nouvelle langue étrangère) est inadéquate : tout le monde peut devenir bilingue par immersion linguistique précoce, scelle une minorité – hélas – pourra maîtriser le code informatique pour en faire un métier.

Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former

Il faut bien sûr donner une culture numérique de base à tous les enfants pour les aider à se mouvoir dans le futur, mais il est plus crucial encore de former leur esprit critique, ce qui les protégera de la concurrence de l'IA ! Pour un gamin normalement doué, il est tout à fait possible d'apprendre à savoir lire, résumer et critiquer un texte qu'à apprendre quelques banalités en HTML. Dans un monde où l'informatique, devenue IA, va piloter les voitures, les avions, réguler les pacemakers et les coeurs artificiels, augmenter nos cervaeaux avec les implants d'Elon Musk, décider du traitement des cancéreux et protéger les arsenaux nucléaires contre les hackers, il serait naïf de penser qu'il y aura une grosse niche pour les bidouilleurs de code.

**L**e code de 2030 sera à la convergence de multiples savoirs : droiture, sécurité, gestion de la complexité, neurosciences. Ce sera davantage un bouton pour diplômés de plus Harvard Law School que d'amateurs bien intentionnés. Xavier Niel, le premier entrepreneur français à avoir décrypté le futur, est un précurseur de la formation des jeunes au code informatique avec la création de l'école 42 : c'est une admirable réussite, qui commence à éessimier. A côté de Station F, le plus grand incubateur de start-up au monde, c'est un modèle d'intégration pour beaucoup de jeunes, y compris pour ceux qui ont décroché le système scolaire. Mais Xavier Niel lui-même reconnaît un peu d'ignorance historique et philosophique... – au programme de l'école 42 : « Savoir l'avenir à long terme de ses étudiants ! L'enjeu est aussi politique : ne multiplions pas les sources de déceptions et d'aliénages pour les jeunes générations, qui deviendraient de futures recrues potentielles pour les idéologues populistes, en déployant, en 2020, une idée qui aurait été adaptée à l'information des années 1990. »

**L**e code informatique est la base de la révolution informatique. Doit-on généraliser son apprentissage ? La question est bien difficile, tant il existe un consensus pour répondre oui... C'est, en apparence, une proposition logique et pleine de bon sens qui est en réalité absurde. Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former : a-t-on envisagé que 100 % des jeunes deviennent

électriciens en 1895, lorsque l'électricité s'est généralisée et est devenue la base de la deuxième révolution industrielle ?

*Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former*

neurosciences, diplômés entrepreneurs | entreprenariat

neurosciences, diplômés entrepreneurs | entreprenariat



DEMAIN SERA VERTIGINEUX, PAR

## LAURENT ALEXANDRE

### Apprendre à coder ? Non !

**L**e code informatique est la base de la révolution informatique. Doit-on généraliser son apprentissage ? La question est bien difficile, tant il existe un consensus pour répondre oui... C'est, en apparence, une proposition logique et pleine de bon sens qui est en réalité absurde. Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former : a-t-on envisagé que 100 % des jeunes deviennent électriciens en 1895, lorsque l'électricité s'est généralisée et est devenue la base de la deuxième révolution industrielle ? Avoir une culture générale informatique est, bien sûr, essentiel pour être un citoyen au fait des enjeux numériques et capable de participer au débat politique sur le monde qui vient. Mais les plans de formation massifs de développeurs informatiques bac + 2 conduiront à beaucoup de désillusions. Le code informatique bas de gamme sera entièrement automatisé grâce à l'intelligence artificielle (IA), qui sera quasi gratuite. Ainsi, le patron de Nvidia, l'un des plus importants fabricants de microprocesseurs dédiés à l'IA, explique : « L'IA va manger le logiciel. »

**B**ien sûr, il y a un immense besoin d'architectes informatiques et de *data scientists*. Mais ces métiers ne seront accessibles qu'à des esprits multidisciplinaires de très haut niveau. Codeur informatique bas de gamme sera un passeport pour Pôle emploi ! Le spécialiste de l'IA Serge Abiteboul résume, dans *Le Monde*, l'horrible réalité d'une phrase assassine et bien peu politiquement correcte : « Data scientist est un métier qui demande énormément de neurones. » La métaphore de la langue étrangère (le code serait la nouvelle langue étrangère) est inadaptée : tout le monde peut devenir bilingue par immersion linguistique précoce, seule une minorité – hélas – pourra maîtriser le code informatique pour en faire un métier.

*Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former*

Il faut bien sûr donner une culture numérique de base à tous les enfants pour les aider à se mouvoir dans le futur, mais il est plus crucial encore de former leur esprit critique, ce qui les protégera de la concurrence de l'IA ! Pour un gamin normalement doué, il est mille fois plus important d'apprendre à savoir lire, résumer et critiquer un texte qu'à annoncer quelques banalités en HTML. Dans un monde où l'informatique, devenue IA, va piloter les voitures, les avions, réguler les pacemakers et les cœurs artificiels, augmenter nos cerveaux avec les implants d'Elon Musk, décider du traitement des cancéreux et protéger les arsenaux nucléaires contre les hackers, il serait naïf de penser qu'il y aura une grosse niche pour les bidouilleurs de code.

**L**e codeur de 2030 sera à la convergence de multiples savoirs : droit, sécurité, gestion de la complexité, neurosciences. Ce sera davantage un boulot pour diplômés de X plus Harvard Law School que d'entrepreneurs bien intentionnés. Xavier Niel, le premier entrepreneur français à avoir décrypté le futur, est un précurseur de la formation des jeunes au code informatique avec la création de l'école 42 : c'est une admirable réussite, qui commence à essaimer. A côté de Station F, le plus grand incubateur de startup au monde, c'est un modèle d'intégration pour beaucoup de jeunes, y compris pour ceux qui ont décroché du système scolaire. Mais Xavier Niel lui-même doit mettre un peu d'humanités – histoire et philosophie... – au programme de l'école 42 pour assurer l'avenir à long terme de ses étudiants ! L'enjeu est aussi politique : ne multiplions pas les sources de déceptions et d'âgrefours pour les jeunes générations, qui deviendraient de futures recrues potentielles pour les idéologues populistes, en déployant, en 2020, une idée qui aurait été adaptée à l'informatique des années 1990. ■

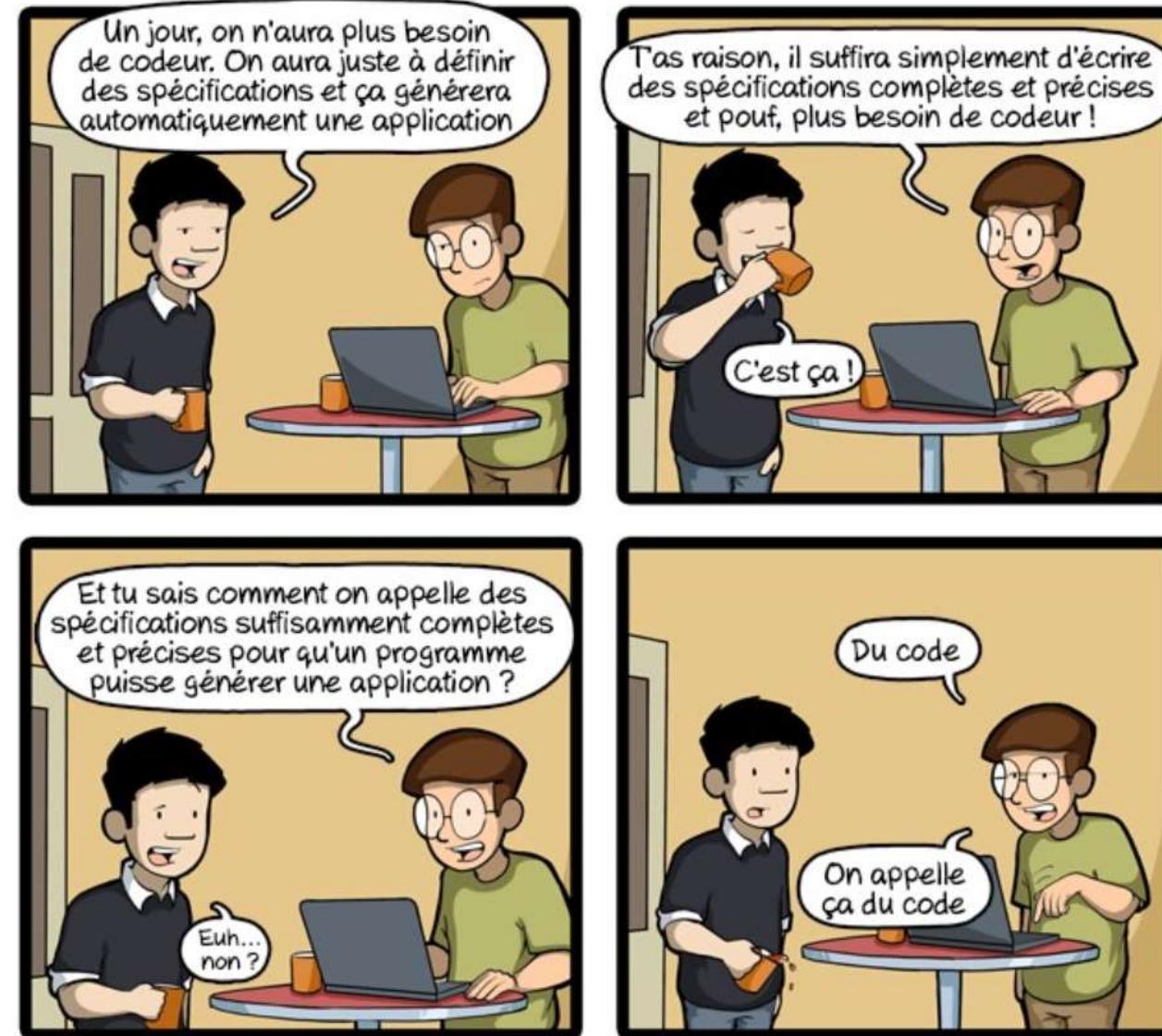
**L**e code informatique est la base de la révolution informatique. Doit-on généraliser son apprentissage ? La question est bien difficile, tant il existe un consensus pour répondre oui... C'est, en apparence, une proposition logique et pleine de bon sens qui est en réalité absurde. Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former : a-t-on envisagé que 100 % des jeunes deviennent électriciens en 1895, lorsque l'électricité s'est généralisée et est devenue la base de la deuxième révolution industrielle ? Avoir une culture générale informatique est, bien sûr, essentiel pour être un citoyen au fait des enjeux numériques et capable de participer au débat politique sur le monde qui vient. Mais les plans de formation massifs de développeurs informatiques bac + 2 conduiront à beaucoup de désillusions. Le code informatique bas de gamme sera entièrement automatisé grâce à l'intelligence artificielle (IA), qui sera quasi gratuite. Ainsi, le patron de Nvidia, l'un des plus importants fabricants de microprocesseurs dédiés à l'IA, explique : « L'IA va manger le logiciel. »

**L**e code informatique bas de gamme sera entièrement automatisé grâce à l'intelligence artificielle (IA), qui sera quasi gratuite. Ainsi, le patron de Nvidia, l'un des plus importants fabricants de microprocesseurs dédiés à l'IA, explique : « L'IA va manger le logiciel. »

Il f  
de bas  
voir da  
formel  
concu  
doué, i  
savoir  
ner qu  
où l'in

*Ce n'est pas parce qu'une technologie est ubiquitaire que tout le monde doit s'y former*

neuro:  
diplône  
teurs |  
entrep  
... ...



CommitStrip.com

## Canard n°3

- « Pourquoi rendre l'informatique obligatoire ? Tout le monde ne deviendra pas informaticien demain »



# Tout le monde informaticien demain ?

- Tout le monde n'a pas vocation à devenir informaticien demain !
- **MAIS** tout le monde sera confronté au numérique demain dans sa **vie privée et professionnelle**
  - Enseignant, Infirmière, Ouvrier, Employé, Médecin, Économistes, Traducteurs, Chauffeur, Pharmacien, Notaire, Marketing, Banquier, ...
  - Et bien sûr mais surtout les demandeurs d'emploi !
- **Avec des prérequis qui vont au-delà des usages !**
  - Compréhension des enjeux, des limites
  - Esprit critique
  - Co-conception participative

## Canard n°5

- Les adolescents de demain seront des « digital natives » qui en connaîtront déjà bien plus que leurs parents ou instituteurs !

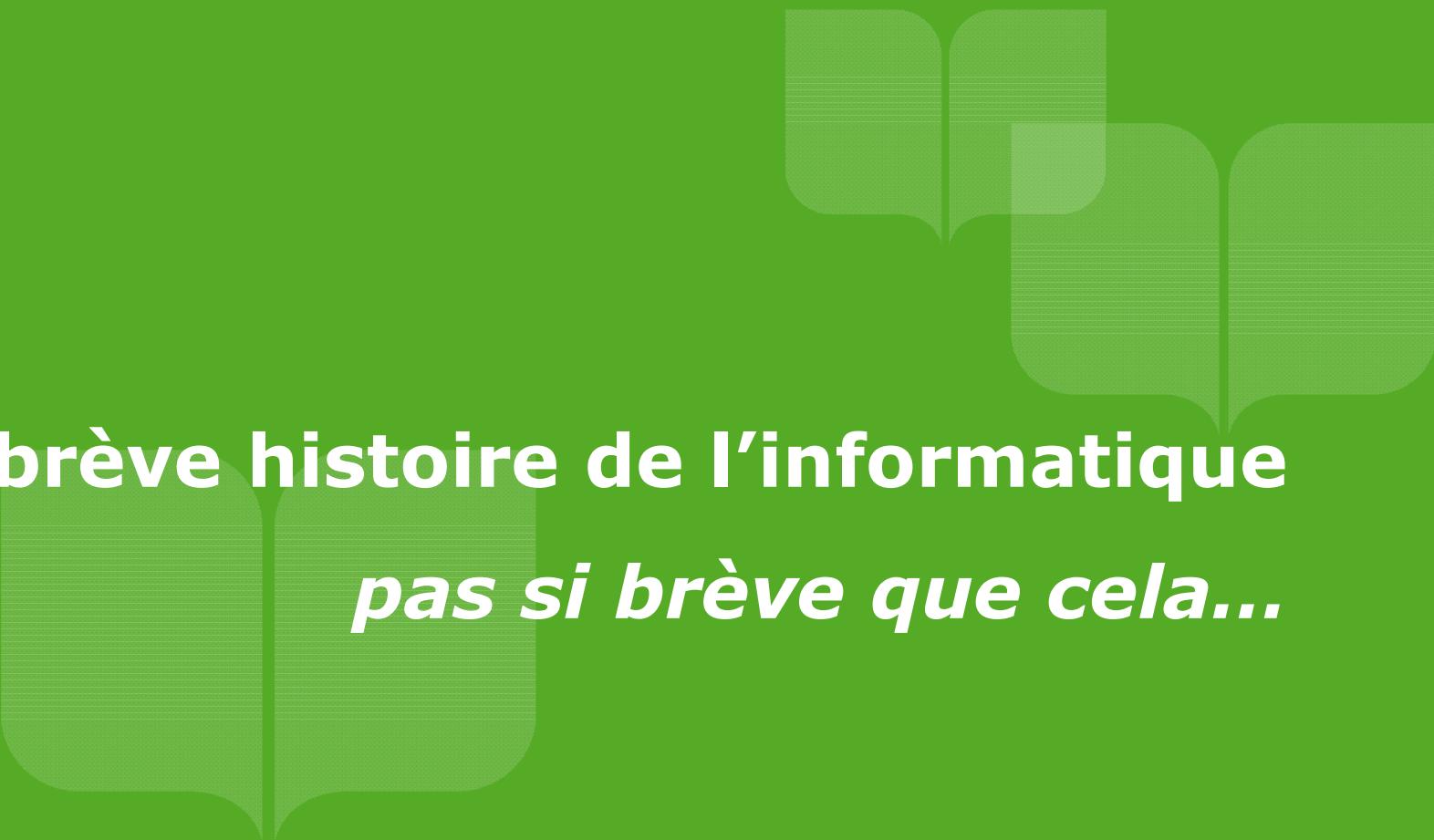
- Se bagarrer dans la cours de récréation ne fait pas des enfants des sportifs rompus à l'art de la boxe !



# Canard n°6

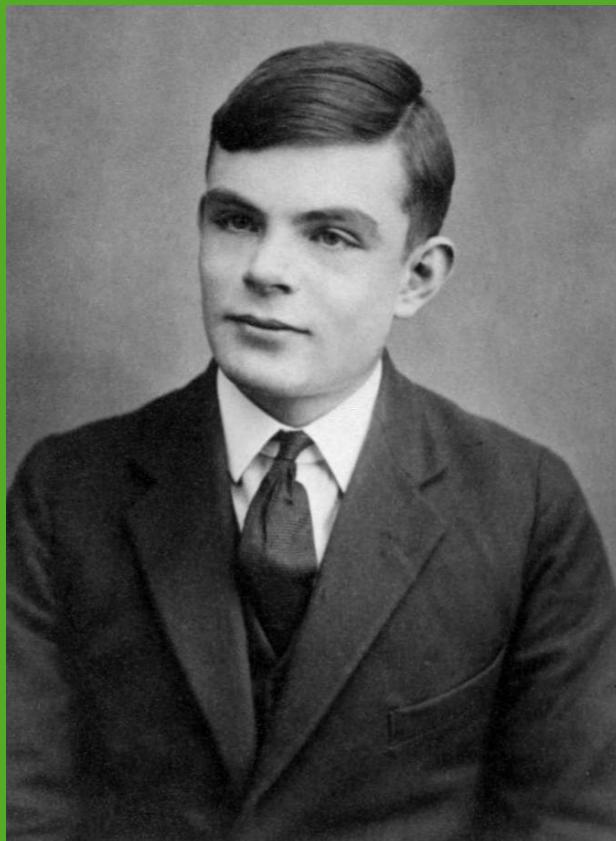
- « D'une certaine manière, **coder n'est qu'une technique de notre époque**. Et je pense que ce serait une grave erreur que cet outil soit enraciné. Vous l'enseignez à des enfants de trois ans et, jusqu'à ce qu'ils aient leur diplôme, ils vous demanderont de leur rappeler ce qu'était **outil sera bientôt obsolète** »
  - Andreas Schleicher, Directeur de l'Education et des Compétences au Secrétariat Général de l'OCDE.  
World Innovation Summit for Education, Fév. 2019.
- « [...] **je serais beaucoup plus enclin à enseigner la science des données ou la pensée informatique**\* qu'à enseigner une technique très spécifique d'aujourd'hui. »
  - Andreas Schleicher, Directeur de l'Education et des Compétences au Secrétariat Général de l'OCDE.  
World Innovation Summit for Education, Fév. 2019.
  - \*« Computational thinking » dans le texte original.





# **Un brève histoire de l'informatique**

***pas si brève que cela...***

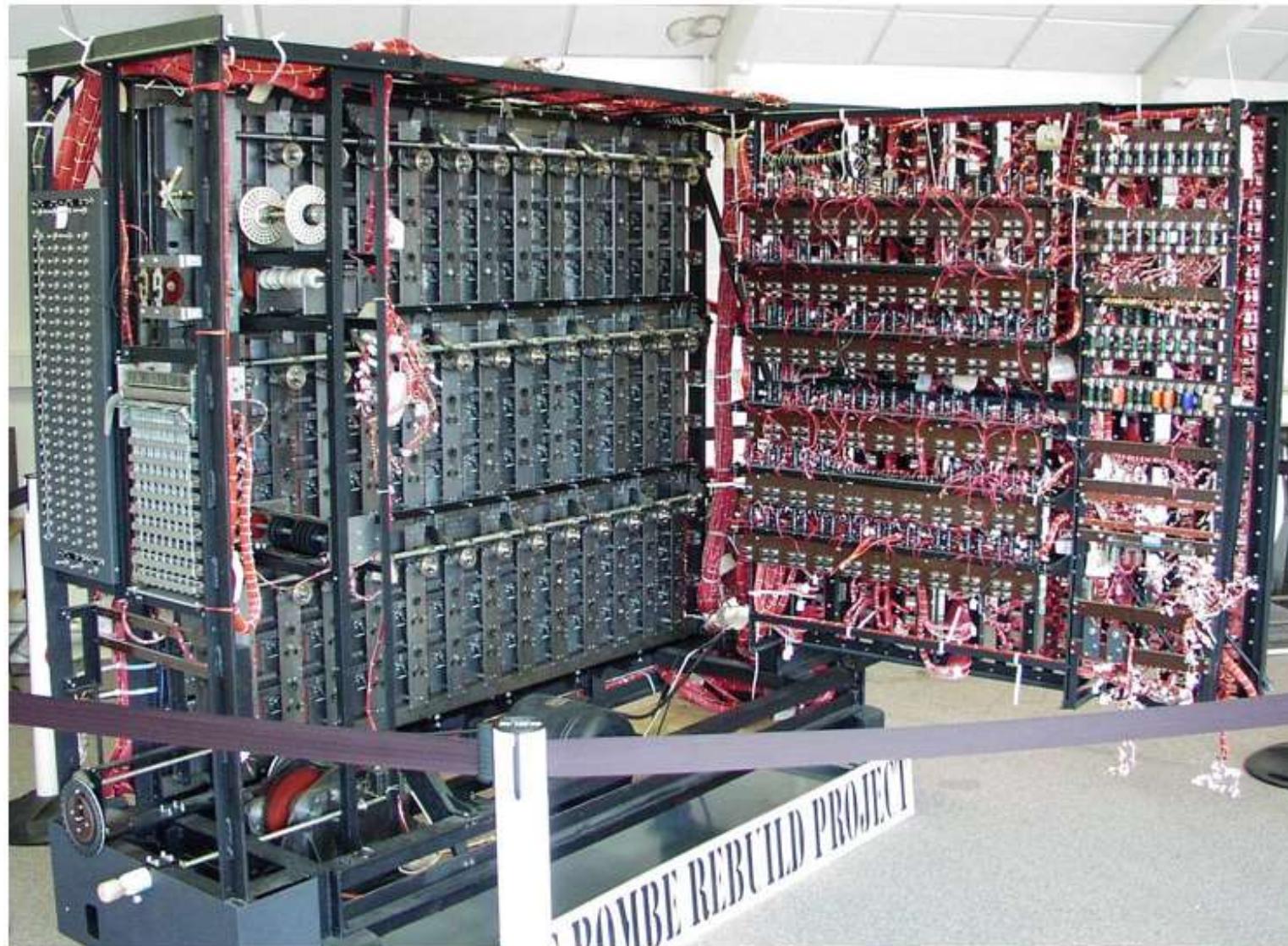


# Turing

# Imitation Game (avec Benedict CUMBERBATCH)

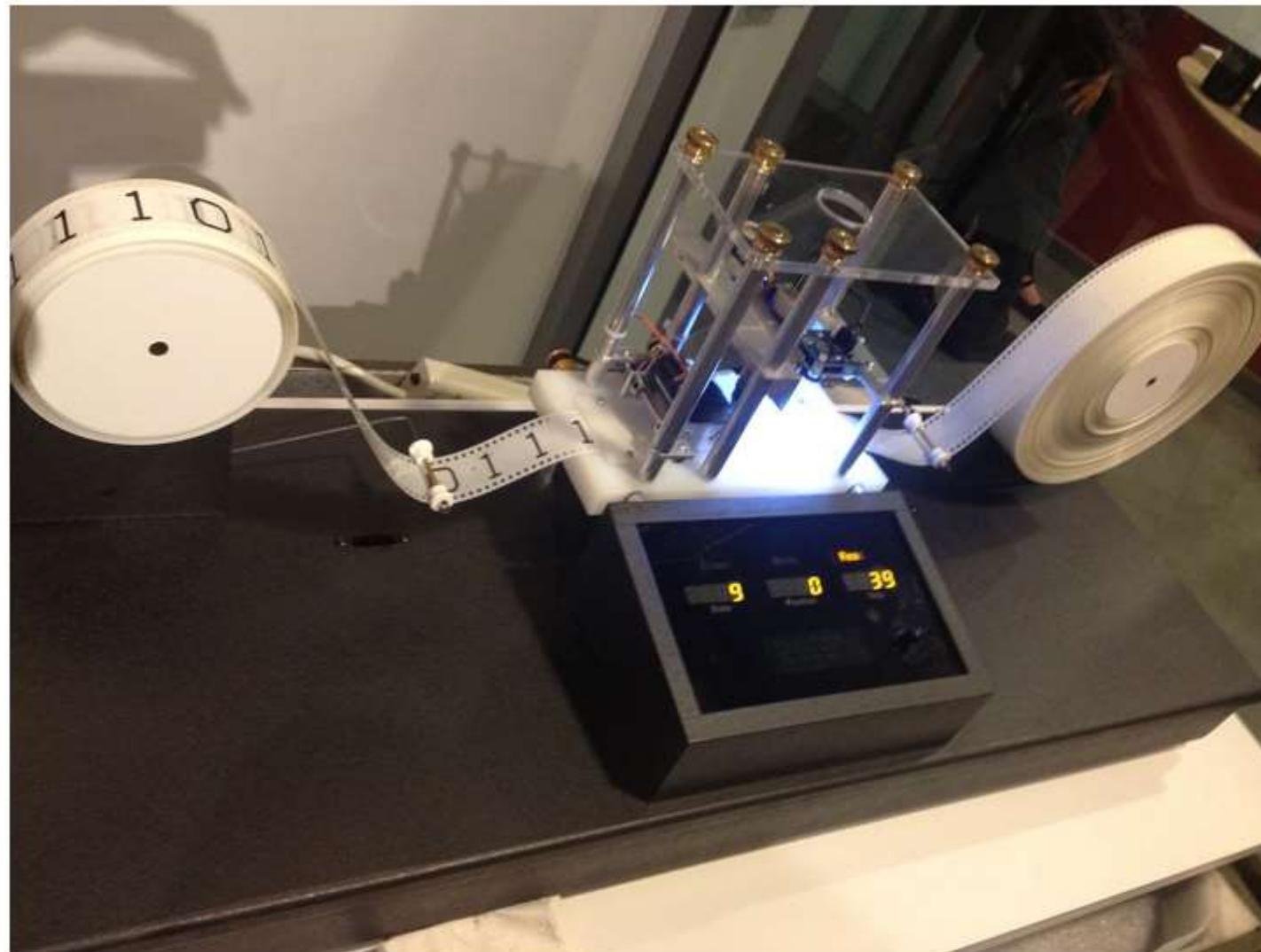


1940→

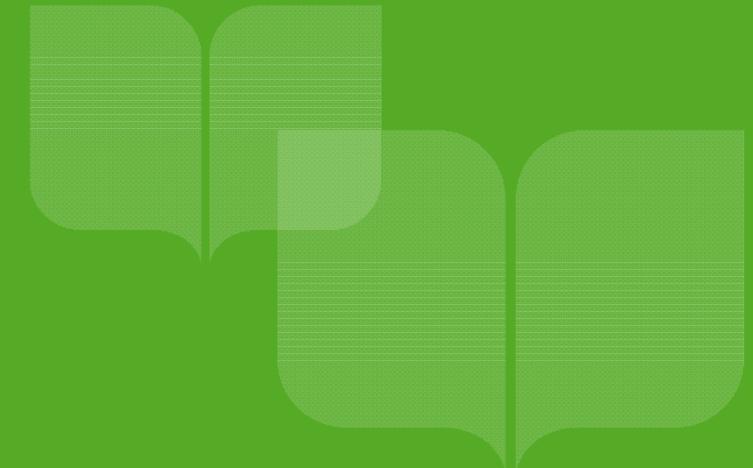
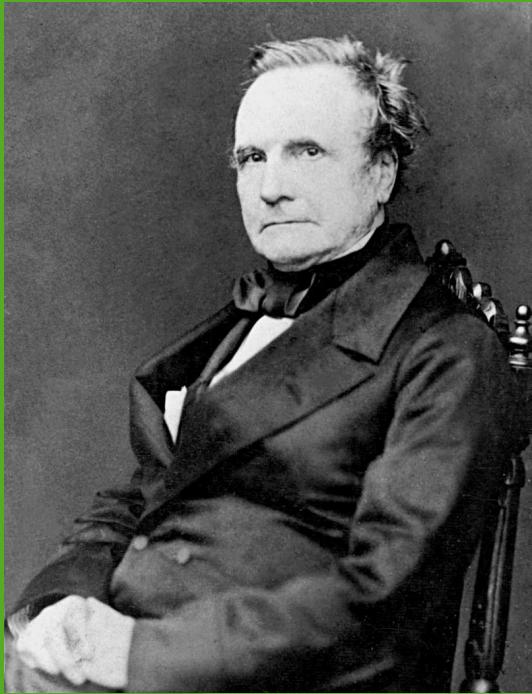


CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=528200>

# 1936



By GabrielF - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=26270095>



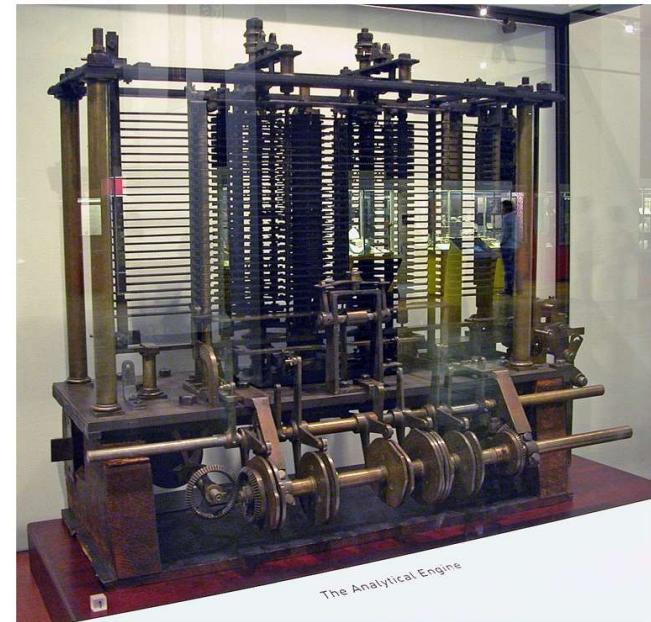
**1834-1836**  
**Babbage**

# La machine à différence

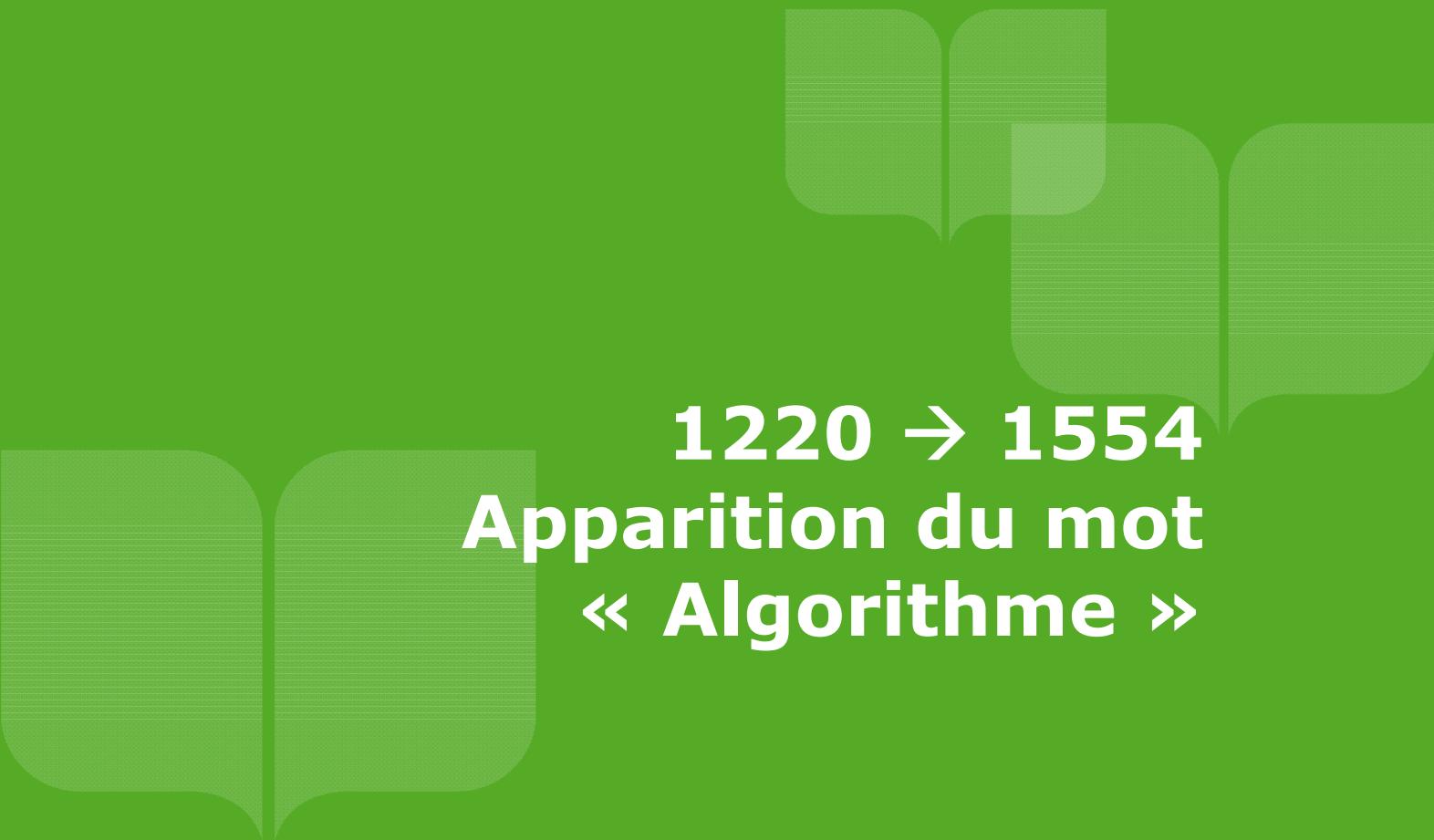
- Il conçoit une machine capable d'exécuter des **algorithmes codés sur des cartes perforées**.
- La machine ne fut pas fabriquée fautes de moyens techniques



Par Margaret Carpenter — Art Work Details page. Original upload was at English wikipedia at enFile:Ada\_Lovelace.jpg. Domaine public, <https://commons.wikimedia.org/w/index.php?curid=354077>



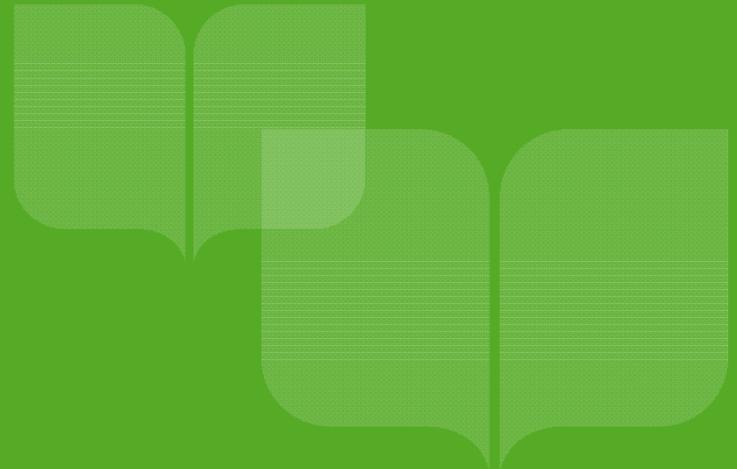
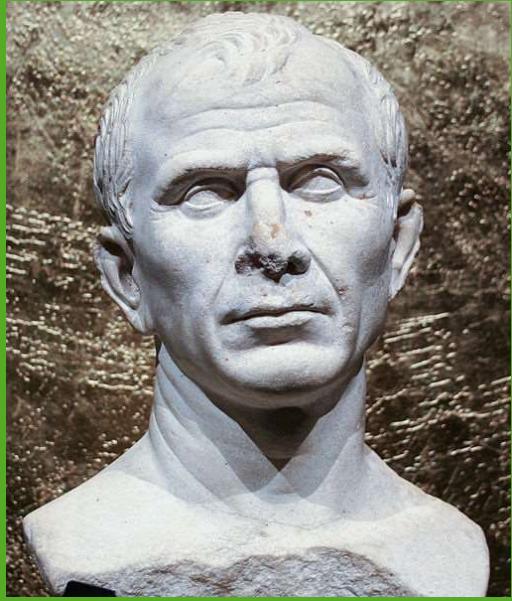
Par Bruno Barral (ByB), CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=6839854>



**1220 → 1554**  
**Apparition du mot**  
**« Algorithme »**

**Étymol. ET HIST.** – 1. Ca 1220-30 **augorisme** « procédé de calcul utilisant les chiffres arabes » (G. de Coinci, *Miracles narratifs de Notre-Dame* ds Gdf. : Or ai tant fait par **augorisme** que cifre ai fait de moi meisme); étant donné le rôle du zéro dans l'arithmétique arabe, on trouve un emploi part. dans l'expr. *être chiffré en algorisme*, c.-à-d. « être comme un zéro, n'avoir aucune valeur », **augorisme** (G. de Coinci, *ibid.* : Or ai tant fait par moi meisme que chifres suis en **augorismes**); xiii<sup>es</sup>. **algorisme** « *id.* » (*Comput*, f°15 ds Littré : Cette senefiance est appelée **algorisme** de le [la] quele nous usons de tels figures : 9, 8, 7, 6, 5, 4, 3, 2, 1); 2. 1534 **algorisme** « l'arithmétique, l'art du calcul en général » (Rabelais, *Gargantua*, 12 ds Hug. : Ces enfans deviendront grands en **algorisme**); 1554 **algoritme** (J. Peletier, *Algèbre* ds DG); 3. 1845 (Besch. : **Algorithme** [...] Mathém. Méthode et notation de toute espèce de calcul). Empr. à l'a. esp. *alguarismo*, attesté au sens de « art de compter, arithmétique », dep. 1256-76, *Libros del Saber de Astronomia* ds Cor., s.v. *guarismo*. L'esp. *alguarismo* est issu de l'ar. *Al Ḥuwārizmī*, littéralement « celui de Huwārizm [territoire de l'Asie Centrale] », surnom du mathématicien ar. du ix<sup>es</sup>. Abdallāh Muhammad ibn Mūsā, dont les trad. ont introduit l'arithm. dans l'Europe médiév. (cf. domaine germ. xii<sup>es</sup>., *Anonymi algorismus Salenensis*, p. 3, 18 ds *Mittellat. W.* s.v., 446, 7 : *prema species algorizmi additio licitur*). La forme *algorithme* est le résultat d'un croisement de *algorisme* avec le gr. αριθμός, nombre (cf. esp. *algoritmo*); FEW t. 19, s.v. *Huwārizm*.

**Source:** CNRTL



-50

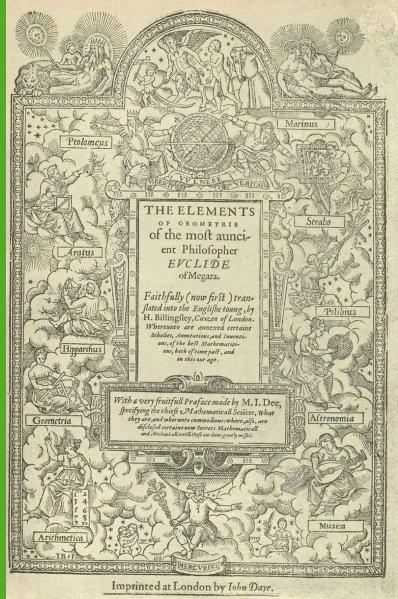
## Le codage de César



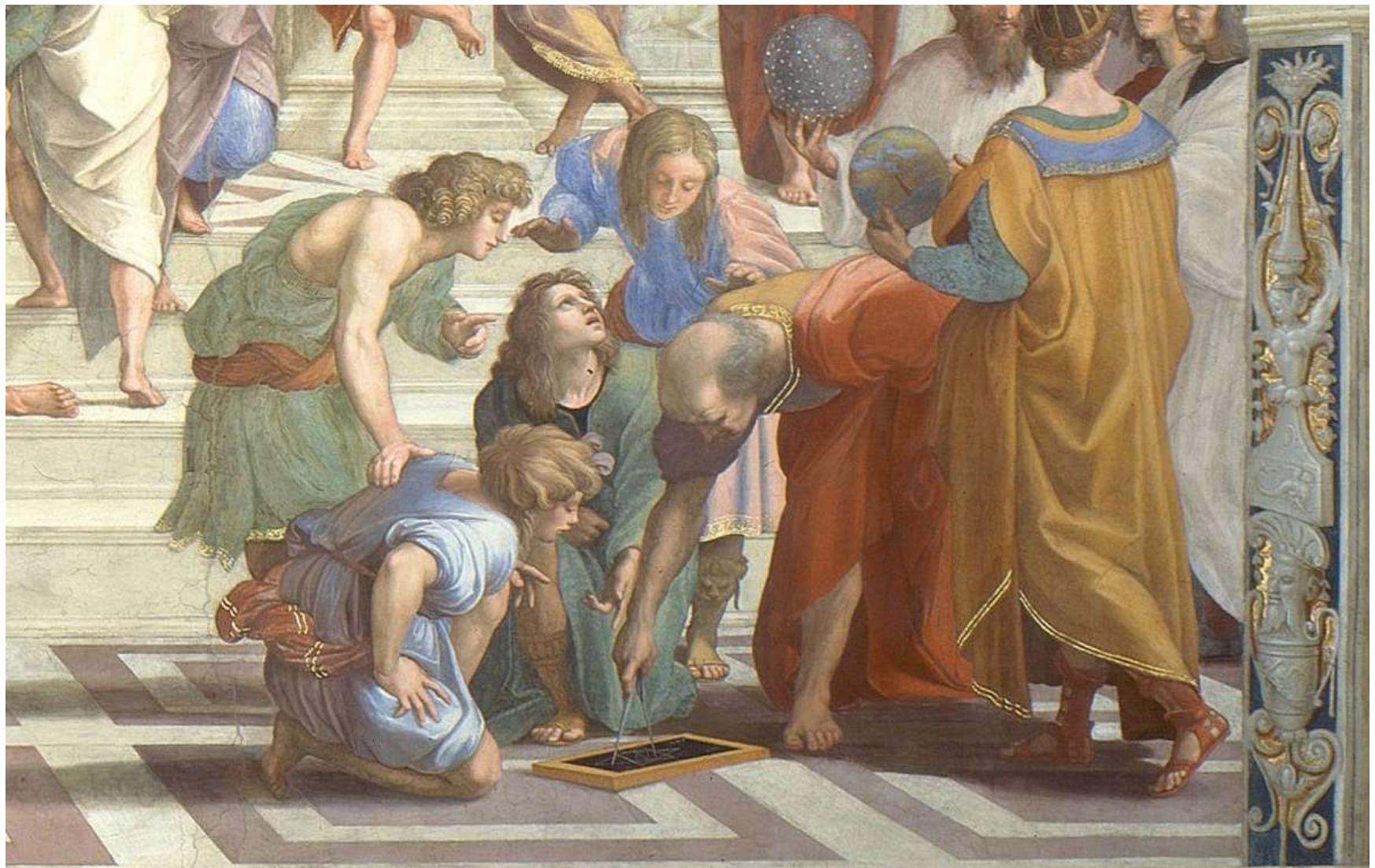
□ « Il y employait, pour les choses tout à fait secrètes, une espèce de chiffre qui en rendait le sens inintelligible (les lettres étant disposées de manière à ne pouvoir jamais former un mot), et qui consistait, je le dis pour ceux qui voudront les déchiffrer, à changer le rang des lettres dans l'alphabet, en écrivant la quatrième pour la première, c'est-à-dire le d pour le a, et ainsi de suite. »

— Suétone, Vie des douze Césars, Livre I, paragraphe 56.

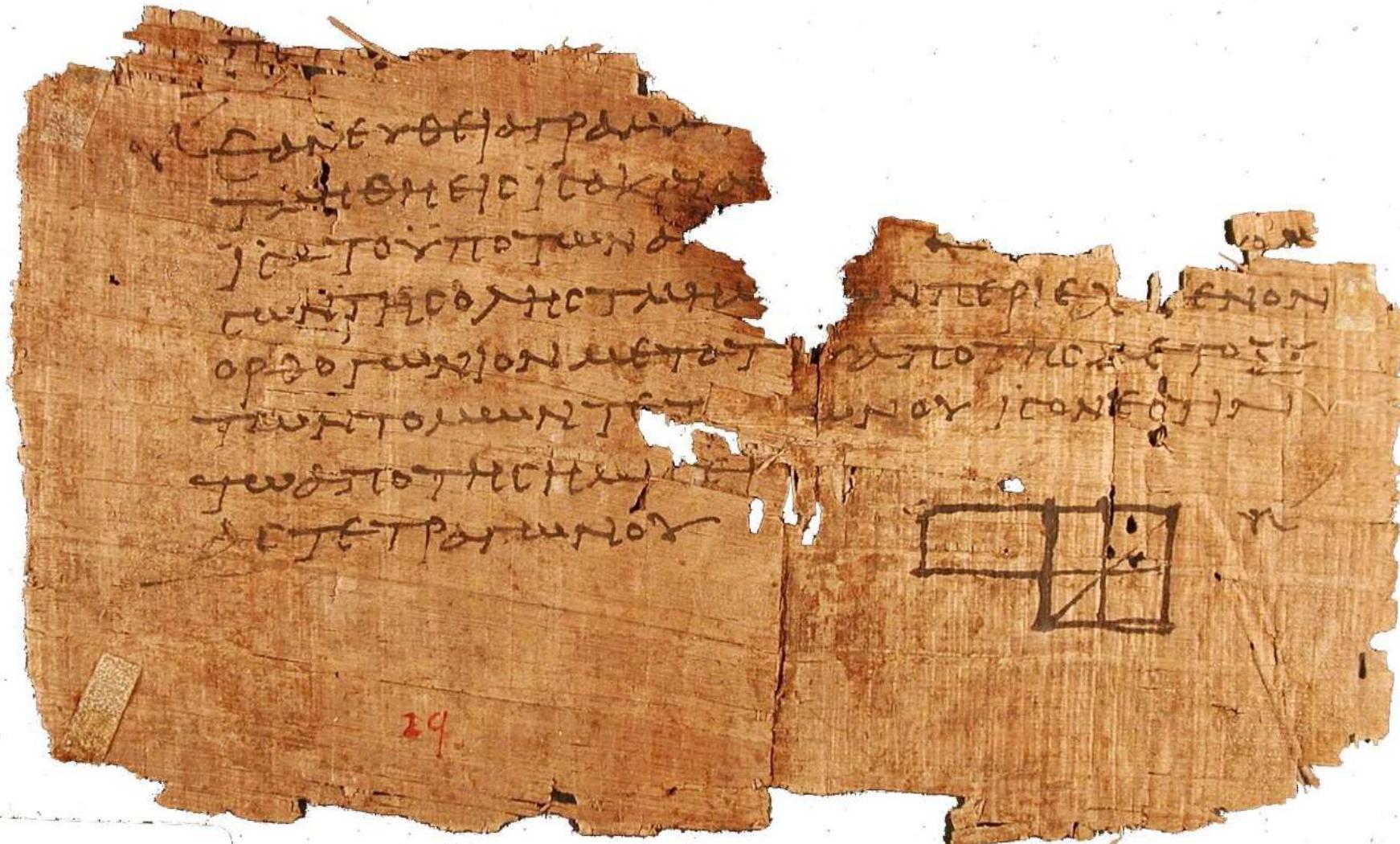
- (1523-1596) Le chiffre de Vigenère
- Cassé en 1863 par Friedrich Kasiski



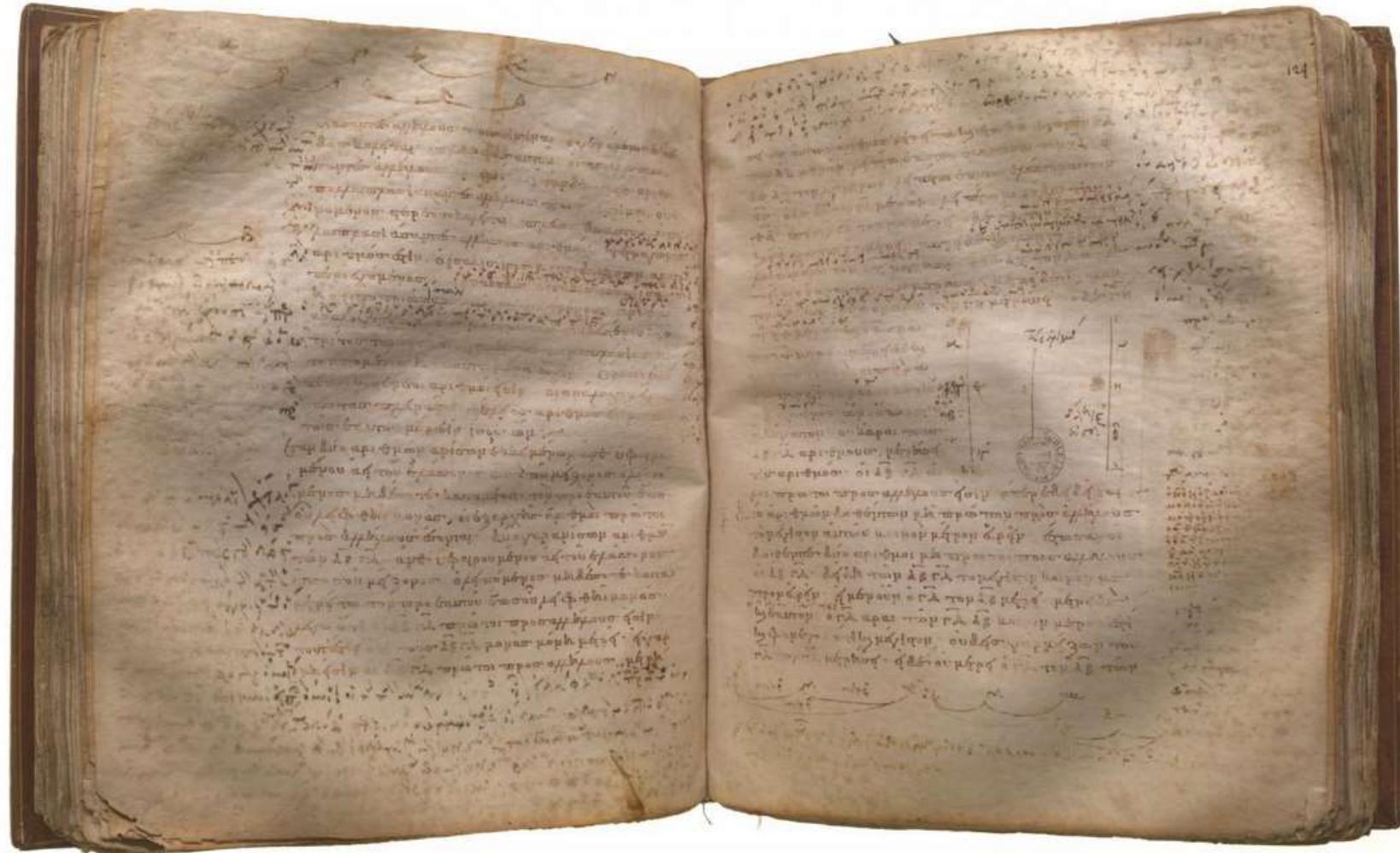
-300  
L'algorithme d'Euclide



Euclid in Raphael's [School of Athens](#) (Wikimedia Commons)



One of the oldest surviving fragments of Euclid's *Elements*, found at Oxyrhynchus and dated to circa AD 100

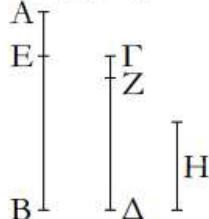


Δύο ἀριθμῶν δοθέντων μὴ πρώτων πρὸς ἄλληλους τὸ μέγιστον αὐτῶν κοινὸν μέτρον εὑρεῖν. Έστωσαν οἱ δοθέντες δύο ἀριθμοὶ μὴ πρῶτοι πρὸς ἄλληλους οἱ ΑΒ, ΓΔ. δεῖ δὴ τὸν ΑΒ, ΓΔ τὸ μέγιστον κοινὸν μέτρον εὑρεῖν. Εἰ μὲν οὖν ὁ ΓΔ τὸν ΑΒ μετρεῖ, μετρεῖ δὲ καὶ ἔαυτόν, ὁ ΓΔ ἄρα τὸν ΓΔ, ΑΒ κοινὸν μέτρον ἐστίν. καὶ φανερόν, ὅτι καὶ μέγιστον: οὐδεὶς γάρ μείζων τοῦ ΓΔ τὸν ΓΔ μετρήσει. Εἰ δὲ οὐ μετρεῖ ὁ ΓΔ τὸν ΑΒ, τὸν ΑΒ, ΓΔ ἀνθυφαιρουμένον ἀεὶ τοῦ ἐλάσσονος ἀπὸ τοῦ μείζονος λειφθῆσεται τις ἀριθμός, δις μετρήσει τὸν πρὸ ἔαυτοῦ. μονάς μὲν γάρ οὐ λειφθῆσεται: εἰ δὲ μή, ἔσονται οἱ ΑΒ, ΓΔ πρῶτοι πρὸς ἄλληλους: δπερ οὐχ ὑπόκειται. λειφθῆσεται τις ἄρα ἀριθμός, δις μετρήσει τὸν πρὸ ἔαυτοῦ. καὶ ὁ μὲν ΓΔ τὸν ΒΕ μετρῶν λειπέτω ἔαυτοῦ ἐλάσσονα τὸν ΕΑ, ὁ δὲ ΕΑ τὸν ΔΖ μετρῶν λειπέτω ἔαυτοῦ ἐλάσσονα τὸν ΖΓ, ὁ δὲ ΓΖ τὸν ΑΕ μετρείτω. ἐπεὶ οὖν ὁ ΓΖ τὸν ΑΕ μετρεῖ, ὁ δὲ ΑΕ τὸν ΔΖ μετρεῖ, καὶ ὁ ΓΖ ἄρα τὸν ΔΖ μετρήσει: μετρεῖ δὲ καὶ ἔαυτόν: καὶ ὅλον ἄρα τὸν ΓΔ μετρήσει. ὁ δὲ ΓΔ τὸν ΒΕ μετρεῖ: καὶ ὁ ΓΖ ἄρα τὸν ΒΕ μετρεῖ: μετρεῖ δὲ καὶ τὸν ΕΑ: καὶ ὅλον ἄρα τὸν ΒΑ μετρήσει: μετρεῖ δὲ καὶ τὸν ΓΔ: ὁ ΓΖ ἄρα τοὺς ΑΒ, ΓΔ μετρεῖ. ὁ ΓΖ ἄρα τὸν ΑΒ, ΓΔ κοινὸν μέτρον ἐστίν. λέγω δή, ὅτι καὶ μέγιστον. εἰ γάρ μή ἐστιν ὁ ΓΖ τὸν ΑΒ, ΓΔ μέγιστον κοινὸν μέτρον, μετρήσει τις τοὺς ΑΒ, ΓΔ ἀριθμοὺς ἀριθμὸς μείζων ὥν τοῦ ΓΖ. μετρείτω, καὶ ἔστω ὁ Η. καὶ ἐπεὶ ὁ Η τὸν ΓΔ μετρεῖ, ὁ δὲ ΓΔ τὸν ΒΕ μετρεῖ, καὶ ὁ Η ἄρα τὸν ΒΕ μετρεῖ: μετρεῖ δὲ καὶ ὅλον τὸν ΒΑ: καὶ λοιπὸν ἄρα τὸν ΑΕ μετρήσει. ὁ δὲ ΑΕ τὸν ΔΖ μετρεῖ: καὶ ὁ Η ἄρα τὸν ΔΖ μετρήσει: μετρεῖ δὲ καὶ ὅλον τὸν ΔΓ: καὶ λοιπὸν ἄρα τὸν ΓΖ μετρήσει ὁ μείζων τὸν ἐλάσσονα: δπερ ἐστὶν ἀδύνατον: οὐκ ἄρα τοὺς ΑΒ, ΓΔ ἀριθμοὺς τις μετρήσει μείζων ὥν τοῦ ΓΖ: ὁ ΓΖ ἄρα τὸν ΑΒ, ΓΔ μέγιστον ἐστι κοινὸν μέτρον: [δπερ ἔδει δεῖξαι].

Given two numbers not prime to one another, to find their greatest common measure. Let AB, CD be the two given numbers not prime to one another. Thus it is required to find the greatest common measure of AB, CD. If now CD measures AB—and it also measures itself—CD is a common measure of CD, AB. And it is manifest that it is also the greatest; for no greater number than CD will measure CD. But, if CD does not measure AB, then, the less of the numbers AB, CD being continually subtracted from the greater, some number will be left which will measure the one before it. For an unit will not be left; otherwise AB, CD will be prime to one another, which is contrary to the hypothesis. Therefore some number will be left which will measure the one before it. Now let CD, measuring BE, leave EA less than itself, let EA, measuring DF, leave FC less than itself, and let CF measure AE. Since then, CF measures AE, and AE measures DF, therefore CF will also measure DF. But it also measures itself; therefore it will also measure the whole CD. But CD measures BE; therefore CF also measures BE. But it also measures EA; therefore it will also measure the whole BA. But it also measures CD; therefore CF measures AB, CD. Therefore CF is a common measure of AB, CD. I say next that it is also the greatest. For, if CF is not the greatest common measure of AB, CD, some number which is greater than CF will measure the numbers AB, CD. Let such a number measure them, and let it be G. Now, since G measures CD, while CD measures BE, G also measures BE. But it also measures the whole BA; therefore it will also measure the remainder AE. But AE measures DF; therefore G will also measure DF. But it also measures the whole DC; therefore it will also measure the remainder CF, that is, the greater will measure the less: which is impossible. Therefore no number which is greater than CF will measure the numbers AB, CD; therefore CF is the greatest common measure of AB, CD.

$\beta'$ .

Δύο ἀριθμῶν διοικέντων μή πρώτων πρὸς ἄλληλους τὸ μέγιστον κοινὸν μέτρον εὑρεῖν.



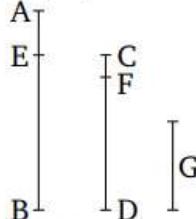
Ἐστιν οὖν οἱ δύο ἀριθμοὶ μή πρῶτοι πρὸς ἄλληλους οἱ ΑΒ, ΓΔ. δεῖ δὴ τὸν ΑΒ, ΓΔ τὸ μέγιστον κοινὸν μέτρον εὑρεῖν.

Εἰ μὲν οὖν ὁ ΓΔ τὸν ΑΒ μετρεῖ, μετρεῖ δὲ καὶ ἔστι τόν, ὁ ΓΔ ἅρα τὸν ΓΑ, ΑΒ κοινὸν μέτρον ἔστιν. καὶ φανερόν, δῆτα καὶ μέγιστον οὐδεὶς γάρ μείζων τοῦ ΓΔ τὸν ΓΔ μετρήσει.

Εἰ δὲ οὐ μετρεῖ ὁ ΓΔ τὸν ΑΒ, τὸν ΑΒ, ΓΔ ἀντιμετρουμένου ἀεὶ τοῦ ἐλάσσονος ἀπὸ τοῦ μείζονος λειφθῆσται τις ἀριθμός. δημητρήσει τὸν πρὸς ἔστιν. μονάς μὲν γάρ οὐ λειφθῆσται· εἰ δὲ μή, ἔσονται οἱ ΑΒ, ΓΔ πρῶτοι πρὸς ἄλληλους διὰ τὸ οὐκ ὑπόκειται λειφθῆσται τις ἄριθμος· διὰ μετρήσει τὸν πρὸς ἔστιν. καὶ οἱ μὲν ΓΔ τὸν ΒΕ μετρῶν λειπόντα ἔστιν τὸν ΕΑ, ὁ δὲ ΕΑ τὸν ΔΖ μετρῶν λειπόντα ἔστιν τὸν ΖΓ, ὁ δὲ ΓΖ τὸν ΑΕ μετρεῖται. ἐπεὶ οὖν ὁ ΓΖ τὸν ΑΕ μετρεῖ, ὁ δὲ ΑΕ τὸν ΔΖ μετρεῖ, καὶ ὁ ΓΖ ἅρα τὸν ΔΖ μετρήσει, μετρεῖ δὲ καὶ ἔστιν· καὶ ὁλὸν ἅρα τὸν ΓΔ μετρήσει. ὁ δὲ ΓΔ τὸν ΒΕ μετρεῖ· καὶ ὁ ΓΖ ἅρα τὸν ΒΕ μετρεῖ· μετρεῖ δὲ καὶ τὸν ΕΑ· καὶ ὁλὸν ἅρα τὸν ΓΔ μετρήσει· μετρεῖ δὲ καὶ τὸν ΓΔ· καὶ ὁλὸν ἅρα τὸν ΑΒ· ΓΔ μετρεῖ. ὁ ΓΖ ἅρα τὸν ΑΒ, ΓΔ κοινὸν μέτρον ἔστιν. λέγω δὴ, δῆτα καὶ μέγιστον. εἰ γάρ μὴ ἔστιν ὁ ΓΖ τὸν ΑΒ, ΓΔ μέγιστον κοινὸν μέτρον, μετρήσει τις τοὺς ΑΒ, ΓΔ ἀριθμούς ἀριθμός μείζων ὃν τοῦ ΓΖ, μετρήσει, καὶ ἔστω ὁ Η. καὶ ἐπεὶ ὁ Η τὸν ΓΔ μετρεῖ, ὁ δὲ ΓΔ τὸν ΒΕ μετρεῖ, καὶ ὁ Η ἅρα τὸν ΒΕ μετρεῖ· μετρεῖ δὲ καὶ ὁλὸν τὸν ΒΑ· καὶ λοιπὸν ἅρα τὸν ΑΕ μετρήσει. ὁ δὲ ΑΕ τὸν ΔΖ μετρεῖ· καὶ ὁ Η ἅρα τὸν ΔΖ μετρήσει· μετρεῖ δὲ καὶ ὁλὸν τὸν ΔΓ· καὶ λοιπὸν ἅρα τὸν ΓΖ μετρήσει ὁ μείζων τὸν ἐλάσσονα· διὰ τὸ ἔστιν ἀδύνατον οὐκ ἅρα τοὺς ΑΒ, ΓΔ ἀριθμούς ἀριθμός τις μετρήσει μείζων ὃν τοῦ ΓΖ· ὁ ΓΖ ἅρα τὸν ΑΒ, ΓΔ μέγιστον ἔστι κοινὸν μέτρον [διὰ τὸ δεῖται δεῖξαι].

### Proposition 2

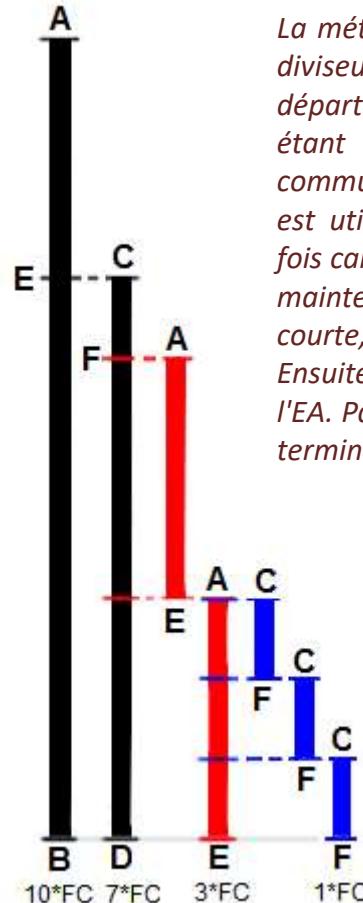
To find the greatest common measure of two given numbers (which are) not prime to one another.



Let  $AB$  and  $CD$  be the two given numbers (which are) not prime to one another. So it is required to find the greatest common measure of  $AB$  and  $CD$ .

In fact, if  $CD$  measures  $AB$ ,  $CD$  is thus a common measure of  $CD$  and  $AB$ , (since  $CD$ ) also measures itself. And (it is) manifest that (it is) also the greatest (common measure). For nothing greater than  $CD$  can measure  $CD$ .

But if  $CD$  does not measure  $AB$  then some number will remain from  $AB$  and  $CD$ , the lesser being continually subtracted, in turn, from the greater, which will measure the (number) preceding it. For a unit will not be left. But if not,  $AB$  and  $CD$  will be prime to one another [Prop. 7.1]. The very opposite thing was assumed. Thus, some number will remain which will measure the (number) preceding it. And let  $CD$  measuring  $BE$  leave  $EA$  less than itself, and let  $EA$  measuring  $DF$  leave  $FC$  less than itself, and let  $CF$  measure  $AE$ . Therefore, since  $CF$  measures  $AE$ , and  $AE$  measures  $DF$ ,  $CF$  will thus also measure  $DF$ . And it also measures itself. Thus, it will also measure the whole of  $CD$ . And  $CD$  measures  $BE$ . Thus,  $CF$  also measures  $BE$ . And it also measures  $EA$ . Thus, it will also measure the whole of  $BA$ . And it also measures  $CD$ . Thus,  $CF$  measures (both)  $AB$  and  $CD$ . Thus,  $CF$  is a common measure of  $AB$  and  $CD$ . So I say that (it is) also the greatest (common measure). For if  $CF$  is not the greatest common measure of  $AB$  and  $CD$  then some number which is greater than  $CF$  will measure the numbers  $AB$  and  $CD$ . Let it (so) measure ( $AB$  and  $CD$ ), and let it be  $G$ . And since  $G$  measures  $CD$ , and  $CD$  measures  $BE$ ,  $G$  thus also measures  $BE$ . And it also measures the whole of  $BA$ . Thus, it will also measure the remainder  $AE$ . And  $AE$  measures  $DF$ . Thus,  $G$  will also measure  $DF$ . And it also measures the whole of  $DC$ . Thus, it will also measure the remainder  $CF$ , the greater (measuring) the lesser. The very thing is impossible. Thus, some number which is greater than  $CF$  cannot measure the numbers  $AB$  and  $CD$ . Thus,  $CF$  is the greatest common measure of  $AB$  and  $CD$ . [(Which is) the very thing it was required to show].



Euclid's example

La méthode d'Euclide pour trouver le plus grand diviseur commun (GCD) de deux longueurs de départ  $BA$  et  $DC$ , toutes deux définies comme étant des multiples d'une longueur "unitaire" commune. La longueur  $DC$  étant plus courte, elle est utilisée pour "mesurer"  $BA$ , mais une seule fois car le reste  $EA$  est inférieur à  $DC$ . L' $EA$  mesure maintenant (deux fois) la longueur  $DC$  plus courte, le reste  $FC$  étant plus court que l' $EA$ . Ensuite, le  $CF$  mesure (trois fois) la longueur de l' $EA$ . Parce qu'il n'y a pas de reste, le processus se termine avec  $FC$  étant le GCD.

```

def pgcd(m,n) {
    if (n == 0) {
        return m
    } else {
        while (reste(m,n) != 0) {
            t=n;
            n=reste(m,n);
            m=t
        }
        return n
    }
}

def pgcdR(m, n) {
    if (n == 0) {
        return m
    } else {
        if (reste(m,n) == 0) {
            return n
        } else {
            return pgcdR(n, reste(m,n))
        }
    }
}

```

(-780 → -850)

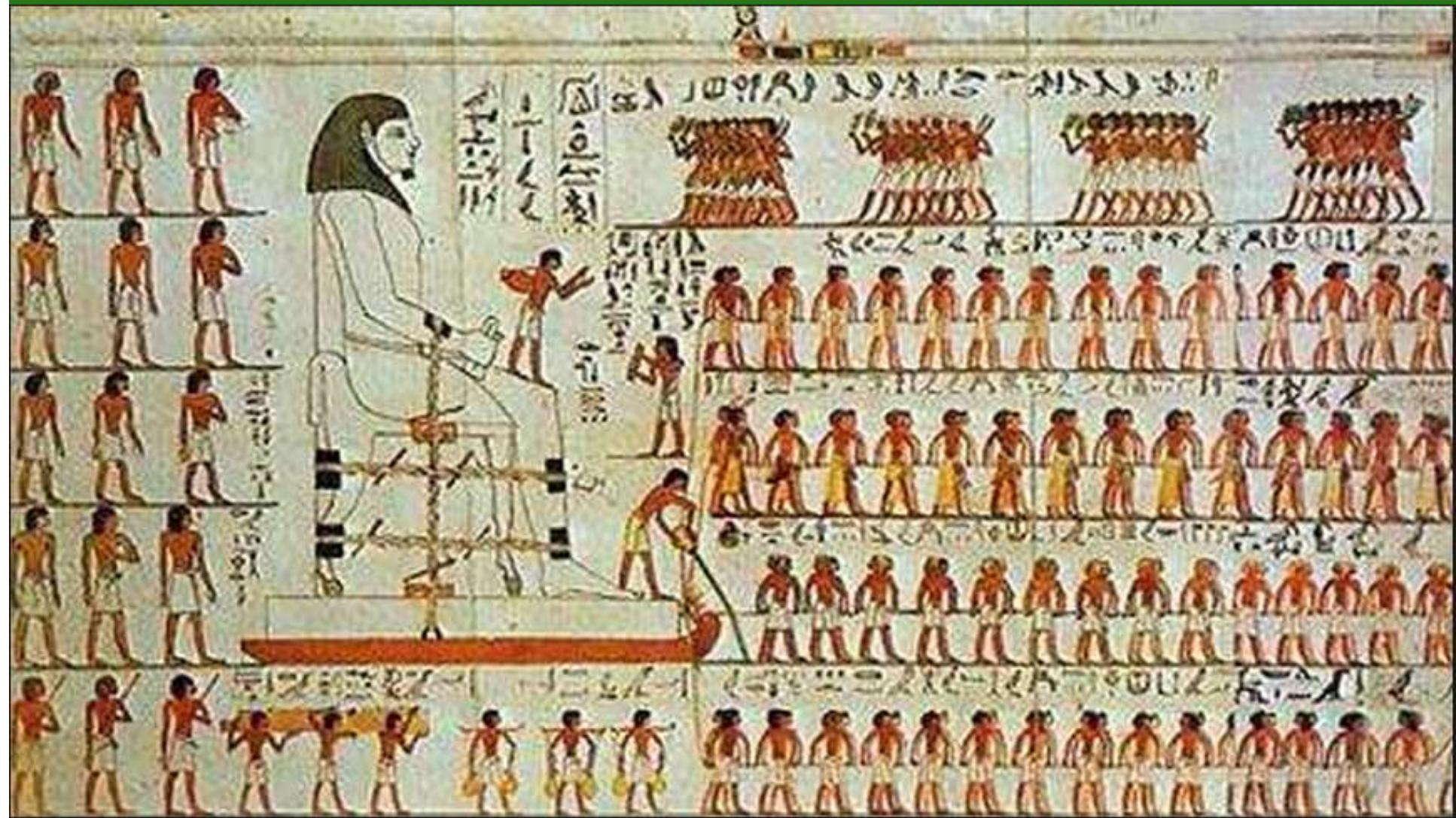
# Muhammad ibn Musa al-Khwarizmi

- Méthode pour calculer la racine carrée d'un nombre





-2700→  
**La construction des pyramides**



Fresque découverte dans la tombe de Djehutihotep (-1900)



-30.000  
**L'Homme de Neandertal**



*« L'animal était soit pris dans un piège (une fosse) puis frappé à la tête, soit tiré à l'arc avec des flèches à pointes de silex »*

Late Mesolithic hunting of a small female aurochs in the valley of the River Tjorger (the Netherlands) in the light of Mesolithic aurochs hunting in NW Europe  
Journal of Archaeological Science, Volume 38, Issue 7, July 2011, Pages 1456-1467 Wietske Prummel, Marcel J. L. Th. Niekus

« Le prototype de l'automatisme est sans doute *le piège* inventé par le chasseur primitif, las de courir après l'animal et s'exposer à ses attaques meurtrières. En quoi le piège est-il un automatisme ? Le trou creusé sur le chemin qu'emprunte l'animal pour aller boire, tapissé de pieux effilés et recouvert de feuillages, dispense le chasseur d'être présent [...] et fonctionne de lui-même, jusqu'à la capture de l'animal. »

Histoire de l'informatique. Philippe Breton. La découverte. 1987.

→ Appréhension d'une série de comportements récurrents par un modèle, et transformation de ce modèle pour concevoir le piège.

# Canard n°6

- « l'informatique n'intéresse pas les filles »
- « c'est un métier d'homme »
- « les filles sont moins compétentes que les garçons »



- ❑ Les Femmes ont eu un rôle prépondérant dans l'histoire de l'informatique.
- ❑ Tout enseignant en informatique démentira ces propos.
- ❑ Dans plusieurs pays asiatiques, il y a plus de femmes que d'hommes dans les études universitaires en informatique
- ❑ Et certaines facultés d'informatique ne comportent que des femmes.

# Des pistes intéressantes



## Is it realistic for both women and men to enter computer science and related fields?

Yes. [Harvey Mudd College](#) demonstrated that women are as capable as men in computer science. They introduced CS courses but changed the names, Introductory Java became Creative Problem Solving in Science and Engineering Using Computational Approaches. And focused on removing the intimidation factor, stemming from lack of prior exposure. They changed the classes to become collaborative and team oriented (which appealed to women) and challenged the stereotype of the loner geek programmer. The percentage of women in CS increased from 10% to 50% and Carnegie Mellon had similar success increasing the women pursuing CS from 7% to 40%. Part of the solution is to remove the stigma that coding is for men and create an environment where women can flourish.

Source: <https://www.cio.com/article/3278274/why-aren-t-more-women-in-computer-science.html>

# mais

- Introduire la pensée informatique dans l'enseignement obligatoire ne suffira pas à réduire la fracture générée si l'on ne prête pas attention aux spécificités de chaque genre.
  - Les garçons et les filles semblent avoir des atouts et difficultés qui leurs sont spécifiques

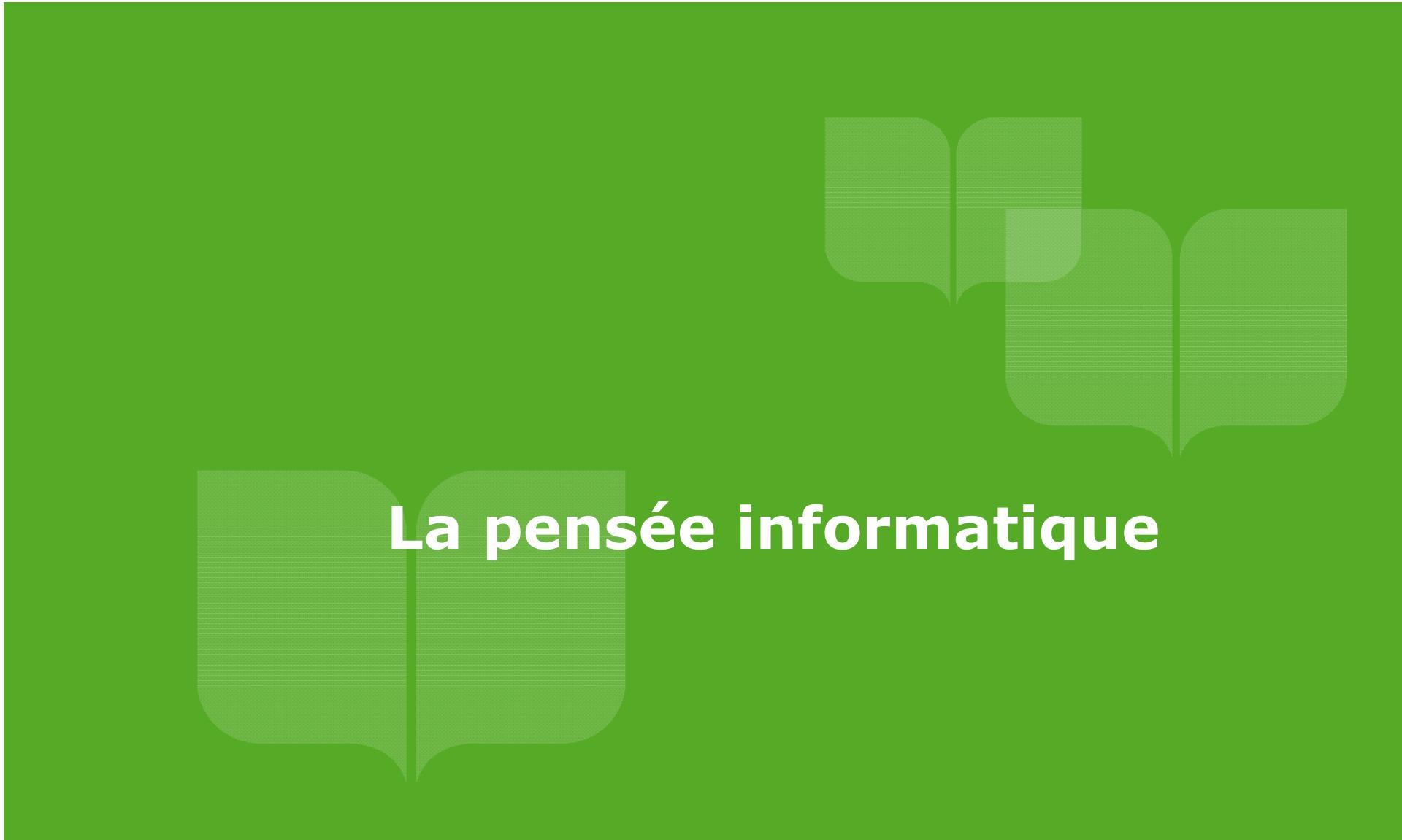


# **Que retenir ?**

# Que retenir ?

1. La pensée informatique est une science à part entière qui s'intéresse à l'information et aux traitements de l'information.
2. La pensée informatique n'est pas liée à l'objet « ordinateur »
  - *La pensée informatique n'est pas (qu') une « technologie »*
3. C'est une affaire de Femmes et d'Hommes.
4. C'est aussi une question de survie...
5. **La pensée informatique est un trait intrinsèque de l'Humanité et de son intelligence**





# La pensée informatique

# L'informatique



*« L'informatique est la science et la technique de la représentation de l'information d'origine artificielle ou naturelle, ainsi que des processus algorithmiques de collecte, stockage, analyse, transformation, communication et exploitation de cette information, exprimés dans des langages formels ou des langues naturelles et effectués par des machines ou des êtres humains, seuls ou collectivement. »*

Société Informatique de France

Source: « L'informatique : la science au cœur du numérique » <http://binaire.blog.lemonde.fr/2014/01/19/l'informatique-la-science-au-coeur-du-numerique-1/>

# L'informatique



« *L'informatique est la science et la technique de la représentation de l'information d'origine artificielle ou naturelle, ainsi que des processus algorithmiques de collecte, stockage, analyse, transformation, communication et exploitation de cette information, exprimés dans des langages formels ou des langues naturelles et effectués par des machines ou des êtres humains, seuls ou collectivement.* »

Société Informatique de France

Source: « *L'informatique : la science au cœur du numérique* » <http://binaire.blog.lemonde.fr/2014/01/19/l'informatique-la-science-au-coeur-du-numerique-1/>

# Et la « pensée informatique »

**« La pensée informatique est un ensemble d'attitudes et d'acquis universellement applicables que tous, et pas seulement les informaticiens, devraient apprendre et maîtriser. »**

## □ Computational Thinking, What it is, and isn't:

1. « Conceptualizing, not programming »
2. « Fundamental, not rote skill »
3. « A way that humans, not computers, think »
4. « Complements and combines mathematical and engineering thinking »
5. « Ideas, not artifacts »
6. « For everyone, everywhere »



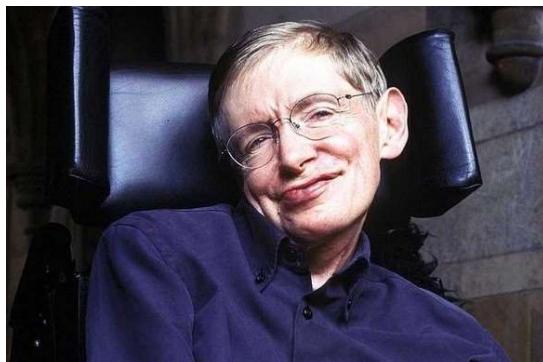
Source: « Computational Thinking » in Communications of the ACM, March 2006 / Vol 49 n°3 by Jeannette M. WING, President's Professor of Computer Science in and head of the Computer science department at Carnegie Mellon University, Pittsburgh, PA.

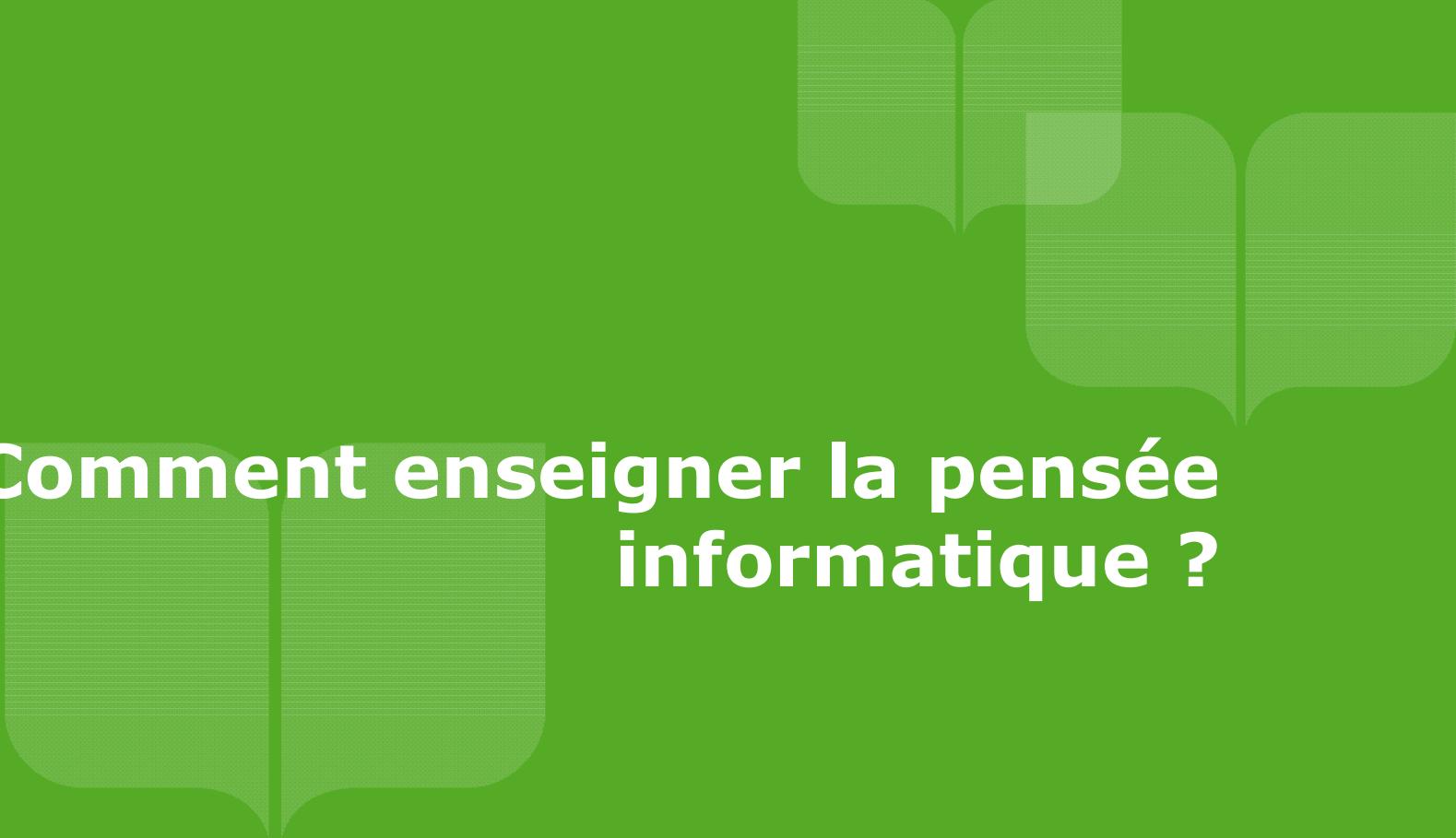
# Quels acquis d'apprentissage ?

- La rigueur**
- L'abstraction**
  - Pouvoir manipuler des concepts comme Variable, Fonction, Itération, État, ...
- La résolution de problèmes**
  - Décomposition, Modélisation, Généralisation, ...
- La créativité**
  - Inventer des solutions algorithmiques
- Le raisonnement logique**
  - L'algèbre booléenne, la logique, ...
- L'esprit critique**
- Les « soft skills »**
  - Expliquer, Argumenter, Savoir être, Collaborer, ...

*“Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn.”*

Stephen Hawking





# Comment enseigner la pensée informatique ?

# Problèmes / Contextes / Motivations

Échanger une photo

Trier un sac de mikado

Trouver le chemin le plus court

Remplir un camion de colis

Échanger un secret

Reconnaitre une forme

Transformer du texte en sens

Optimiser une recette de cuisine

Résoudre une énigme policière

Gérer les feux d'un circuit ferroviaire

Discuter les performances de solutions

etc

Interaction Personne/Machine

Méthodologie

Calculabilité Méta-informatique

Sécurité

Représentation de la l'information

Algorithmique Classique & IA

Abstraction

Communication

## Espace Solutions

Modalité « Déconnectée »



Modalité « Connectée »



Modalité « Incarnée »



# Modalité « Connectée »

SCRATCH Fils tendus créé par Daffodil\_alias\_Daffo Voir la page du projet Rejoindre Scratch Se connecter

Code Costumes Sons

Mouvement Apparence Son Événements Contrôle Capteurs Opérateurs Variables Mes Blocs Stylo

avancer de 10 pas tourner ⌂ de 15 degrés tourner ⌂ de 15 degrés aller à position aléatoire aller à x: 87 y: -84 glisser en 1 secondes à position aléatoire glisser en 1 secondes à x: 87 y: -84 s'orienter en direction de 90 s'orienter vers pointeur de souris ajouter 10 à x mettre x à 87 ajouter 10 à y mettre y à -84 rebondir si le bord est atteint fixer le sens de rotation: gauche-droite

quand flag est cliqué cacher aller à x: 0 y: 0 s'orienter en direction de -90 mettre taille à 0 relever le stylo effacer tout mettre l'intensité du stylo à 0 stylo en position d'écriture attendre 1 secondes répéter 71 fois répéter 3 fois mettre la taille du stylo à 1 avancer de taille pas tourner ⌂ de 360 / 3 degrés s'orienter en direction de direction - 120 attendre 0.01 secondes mettre la couleur du stylo à rose ajouter 0.2 à taille ajouter 0.8 à l'intensité du stylo répéter 3 fois

✓ Géométrie  
✓ Angles  
✓ Déplacement relation  
✓ Positionnement dans l'espace  
✓ Récurrence  
✓ Généralisation  
✓ Invariant  
✓ Etc...

The image shows the Scratch interface with a script in the script editor. The script uses the 'when green flag clicked' event. It starts by hiding the sprite, moving to (0,0), facing down, and setting the pen size to 0. It then enters a loop that repeats 71 times. Inside the loop, it enters another loop that repeats 3 times. In each iteration, it moves forward by a decreasing amount (1, then 0.2, then 0.8) and turns 360/3 degrees. After each inner loop iteration, it changes the pen color to pink. The preview window shows a fractal flower pattern with a pink center and several petal-like shapes, each composed of a series of segments.

Sprite Carres Afficher Taille Direction Scène

Carres Triangles... Spirale d... Spirale tri... Etoile 8 b... Fleur Croix Triangles... Rose

www.unamur.be 71

SCRATCH

Fichier Modifier Tutoriels mBot-race-simulation-st... créé par Wino76FR Voir la page du projet Rejoindre Scratch Se connecter

**Code** **Costumes** **Sons**

Mouvement

Apparence

Son

Événements

Contrôle

Capteurs

Opérateurs

Variables

Mes Blocs

quand est cliqué

aller à l' avant plan

aller à x: -213 y: -17

s'orienter en direction de 0

répéter indéfiniment

si couleur touche noir? et couleur touche noir? alors

avancer de 2 pas

si couleur touche noir? et couleur touche blanc? alors

tourner C° de 5 degrés

si couleur touche blanc? et couleur touche noir? alors

tourner C° de 5 degrés

si couleur touche blanc? et couleur touche blanc? alors

avancer de -2 pas

tourner C° de 3 degrés

jouer le son pop

ajouter 10 à x

mettre x à 171

ajouter 10 à y

mettre y à -125

rebondir si le bord est atteint

fixer le sens de rotation gauche-droite

A 0

Sprite mBot x: 26 y: 62

Afficher Taille 80 Direction 140

Scène

Arrière-plans 1

mBot Track Track2 Track3 Track4

Track5

# Modalité « Incarnée »



# Modalité « incarnée »

micro:bit Accueil Partager Blocs JavaScript Microsoft

The interface shows a Scratch-like programming environment for the micro:bit. On the left, there's a preview of the micro:bit board with red LED patterns and sensor status. The main workspace contains the following script:

```
au démarrage
  définir step à 0
toujours
  montrer nombre step
lorsque secouer
  changer step par 1
  arrêter l'animation
```

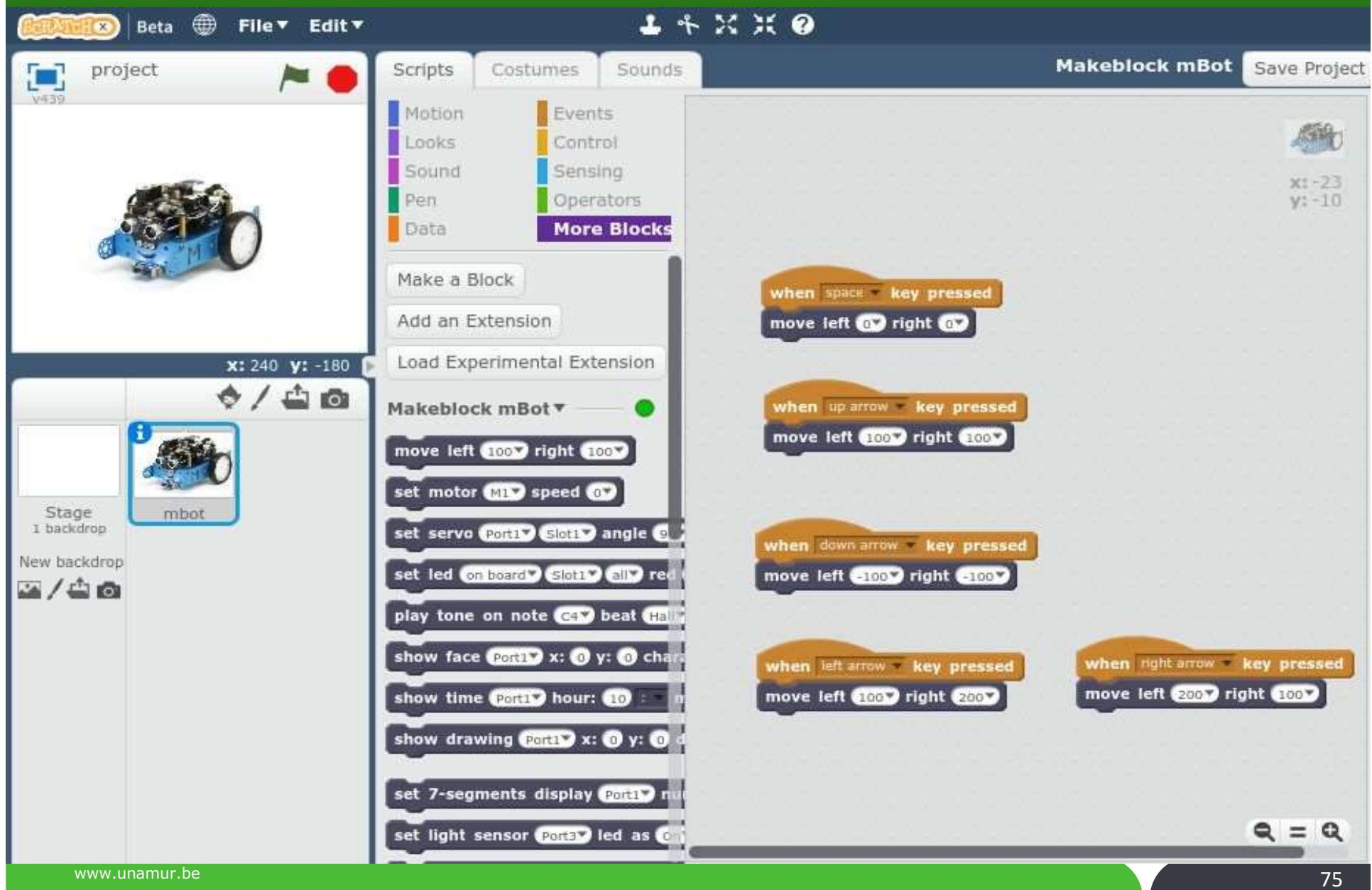
The script starts with an "au démarrage" (when started) hat block followed by a "définir step à 0" (define step as 0) control block. It then has a "toujours" (always) loop containing a "montrer nombre step" (show number step) control block. Finally, it has an "lorsque secouer" (when shake) sensor block that changes the step value by 1 and stops the animation.

A photograph of a smiling child holding a micro:bit board in front of their face, illustrating the concept of embodiment.

Télécharger Step counter

www.unamur.be 74

# Informatique Connectée



# Modalité « déconnectée »

- Exposer et utiliser les principes de la « pensée informatique » pour résoudre de vrais problèmes mais sans utiliser d'ordinateurs
- Se concentrer sur le raisonnement, la logique, l'abstraction, ...

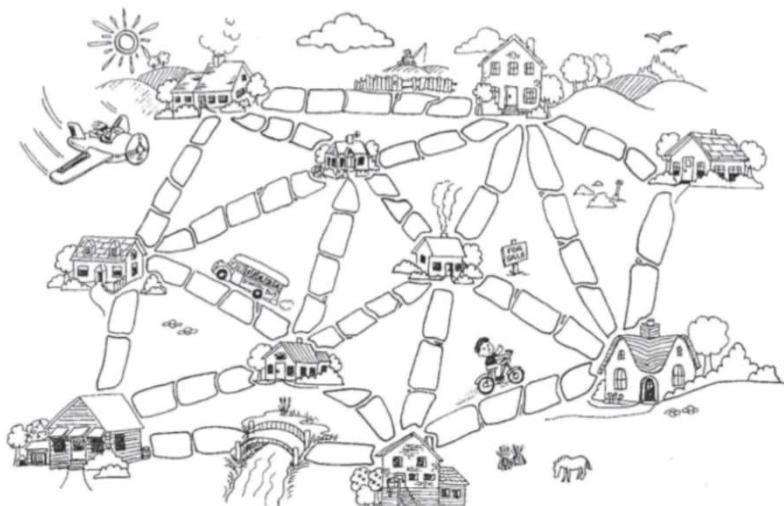
## Exercice : Le problème de la ville embourbée

Il était une fois une ville qui n'avait pas de rues. Il était très difficile de circuler dans la ville après de fortes pluies car le sol était boueux, les voitures s'embourbaient et les bottes des habitants étaient toutes crottées. Le maire de la ville décida de pavé certaines rues mais il ne voulait pas dépenser plus que nécessaire car il voulait également faire construire une piscine pour la ville. Le maire spécifia donc deux conditions :

1. Paver suffisamment de rues pour que chacun des habitants puisse se rendre de sa maison à n'importe quelle autre maison en empruntant des rues pavées.
2. Dépenser le moins d'argent possible pour pavé ces rues.

L'agencement de la ville est représenté ci-dessous. Le nombre de pavés entre chaque maison représente la dépense à engager pour pavé la route. Trouve le meilleur chemin pour relier toutes les maisons mais utilise le moins de jetons (pavés) possible.

Quelles stratégies as-tu utilisées pour résoudre le problème ?



### Data: the raw material—*Representing information*

Count the Dots—*Binary Numbers*

Colour by Numbers—*Image Representation*

You Can Say That Again! —*Text Compression*

Card Flip Magic—*Error Detection & Correction*

Twenty Guesses—*Information Theory*

### Putting Computers to Work—*Algorithms*

Battleships—*Searching Algorithms*

Lightest and Heaviest—*Sorting Algorithms*

Beat the Clock—*Sorting Networks*

The Muddy City—*Minimal Spanning Trees*

The Orange Game—*Routing and Deadlock in Networks*

Tablets of Stone—*Network Communication Protocols*

### Telling Computers What To Do—*Representing Procedures*

Treasure Hunt—*Finite-State Automata*

Marching Orders—*Programming Languages*

### Really hard problems—*Intractability*

The poor cartographer—*Graph coloring*

Tourist town—*Dominating sets*

Ice roads—*Steiner trees*

### Sharing secrets and fighting crime—*Cryptography*

Sharing secrets—*Information hiding protocols*

The Peruvian coin flip—*Cryptographic protocols*

Kid Krypto—*Public-key encryption*

### The human face of computing—*Interacting with computers*

The chocolate factory—*Human interface design*

Conversations with computers—*The Turing test*



# Problèmes / Contextes / Motivations

Échanger une photo

Trier un sac de mikado

Trouver le chemin le plus court

Remplir un camion de colis

Échanger un secret

Reconnaitre une forme

Transformer du texte en sens

Optimiser une recette de cuisine

Résoudre une énigme policière

Gérer les feux d'un circuit ferroviaire

Discuter les performances de solutions

etc

Interaction Personne/Machine

Méthodologie

Calculabilité Méta-informatique

Sécurité

Représentation de la l'information

Algorithmique Classique & IA

Abstraction

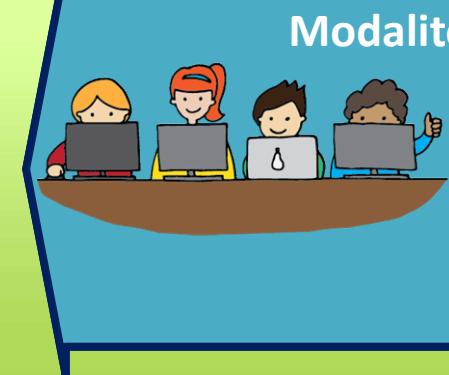
Communication

## Espace Solutions

Modalité « Déconnectée »



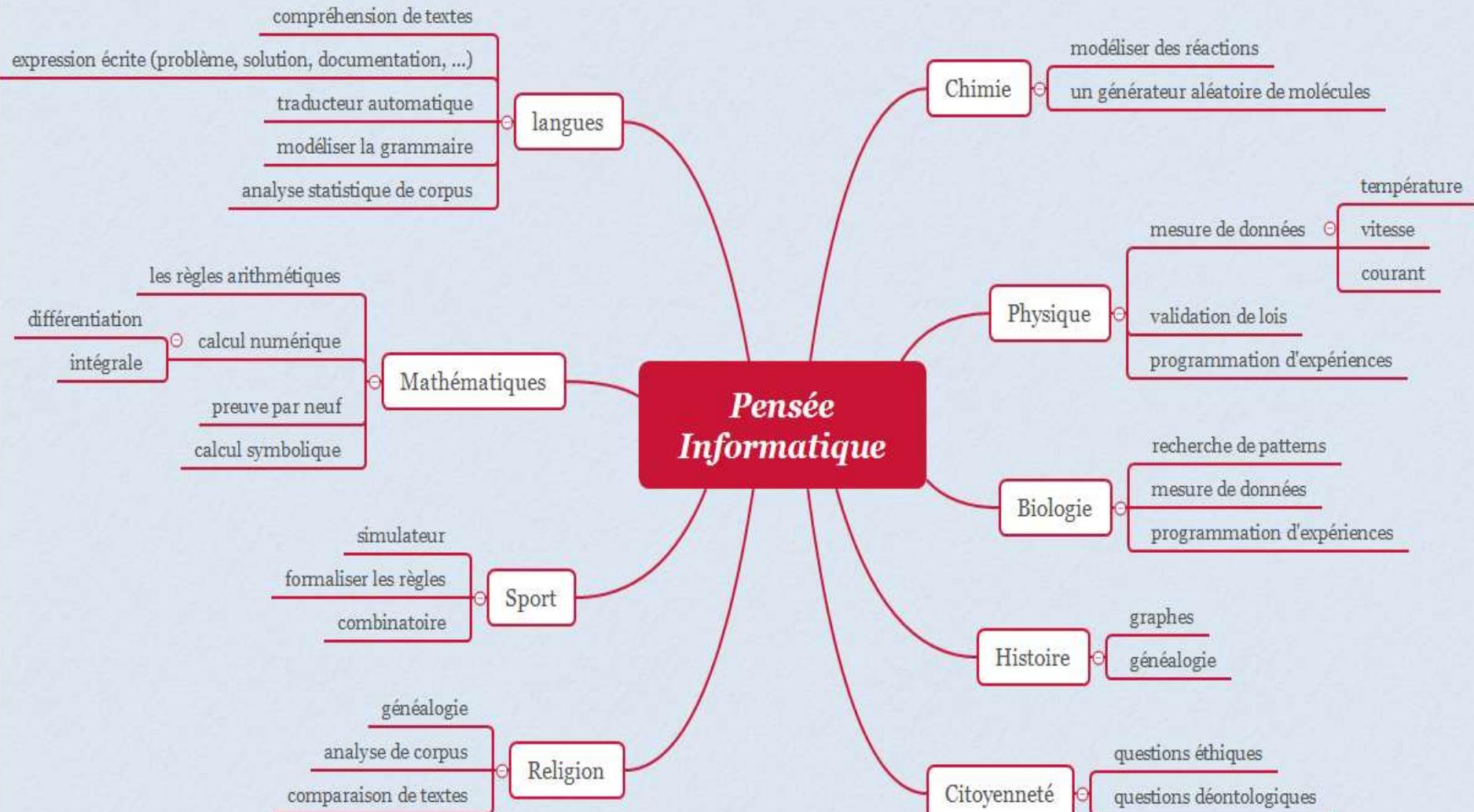
Modalité « Connectée »



Modalité « Incarnée »

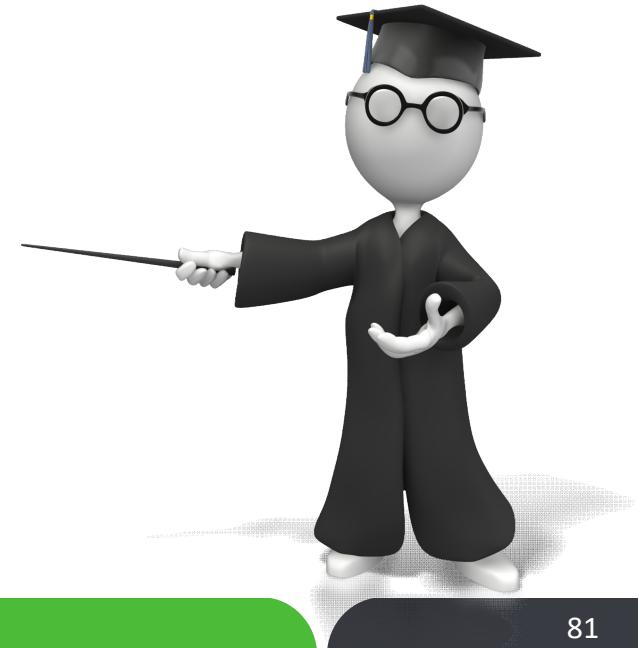


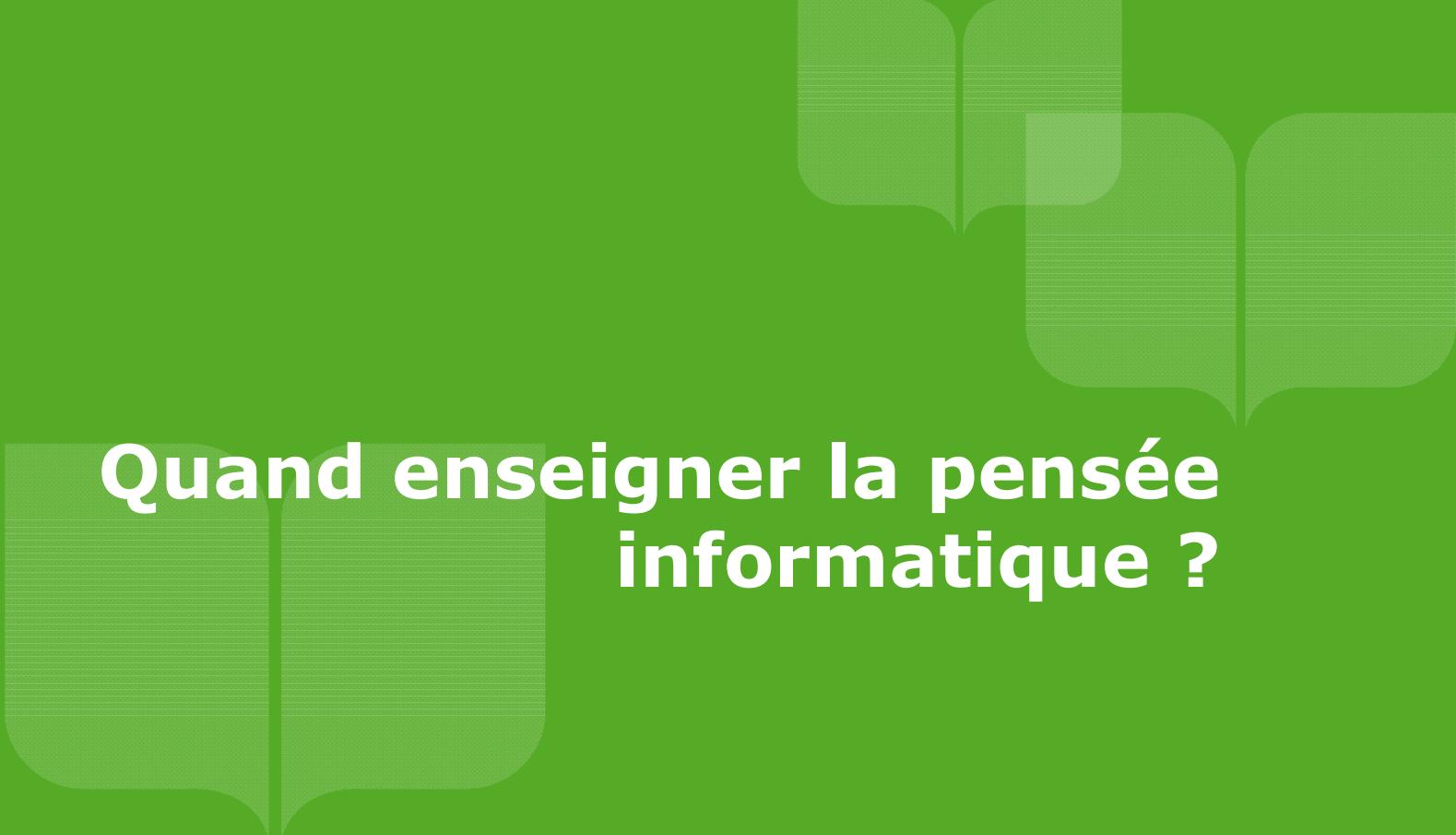
# La pensée informatique est transversale



# Le rôle de l'enseignant !!!

- Le rôle de l'enseignant est capital
  - Maîtrise de la discipline « informatique »
    - Motivations
    - Concepts
    - Méthodes
  - Maîtrise des « artefacts »
    - Logiciels
    - Matériels
  - Méthodes Pédagogies ad-hoc
- **La formation des enseignants ??**





# Quand enseigner la pensée informatique ?

□ A minima, une confrontation avec la pensée informatique par « période » de 2 ou 3 ans de la maternelle jusqu’au supérieur.

- Mittermeir, R. (2013). Algorithmics for Preschoolers—A Contradiction?. *Creative Education*, 4, 557-562. doi: 10.4236/ce.2013.49081.
- Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., & Huber, P. (2016, October). Artificial intelligence and computer science in education: From kindergarten to university. In *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1-9). IEEE

→ « *Il est possible (**sans difficultés**) d’enseigner le sens de l’abstraction à des classes d’enfants qui n’ont pas encore été confrontés à cette compétence par la lecture ou le calcul. (**enfants de 3 à 6 ans**)* »

- Mittermeir, R. (2013). **Algorithmics for Preschoolers**—A Contradiction?. *Creative Education*, 4, 557-562. doi: 10.4236/ce.2013.49081.

The sample is too small to generalize any gender differences out of the result. Nevertheless it is worth mentioning that all 4 girls presented a correct solution while both boys missed the target. Moreover, the girls had the result a little faster than the boys. However, all pupils worked concentrated and were able to describe their approach. Here again, two categories could be identified. One used rather an (almost) random approach, others worked according to a particular strategy.



Mittermeir, R. (2013). Algorithmics for Preschoolers—A Contradiction?. *Creative Education*, 4, 557-562. doi: 10.4236/ce.2013.49081.

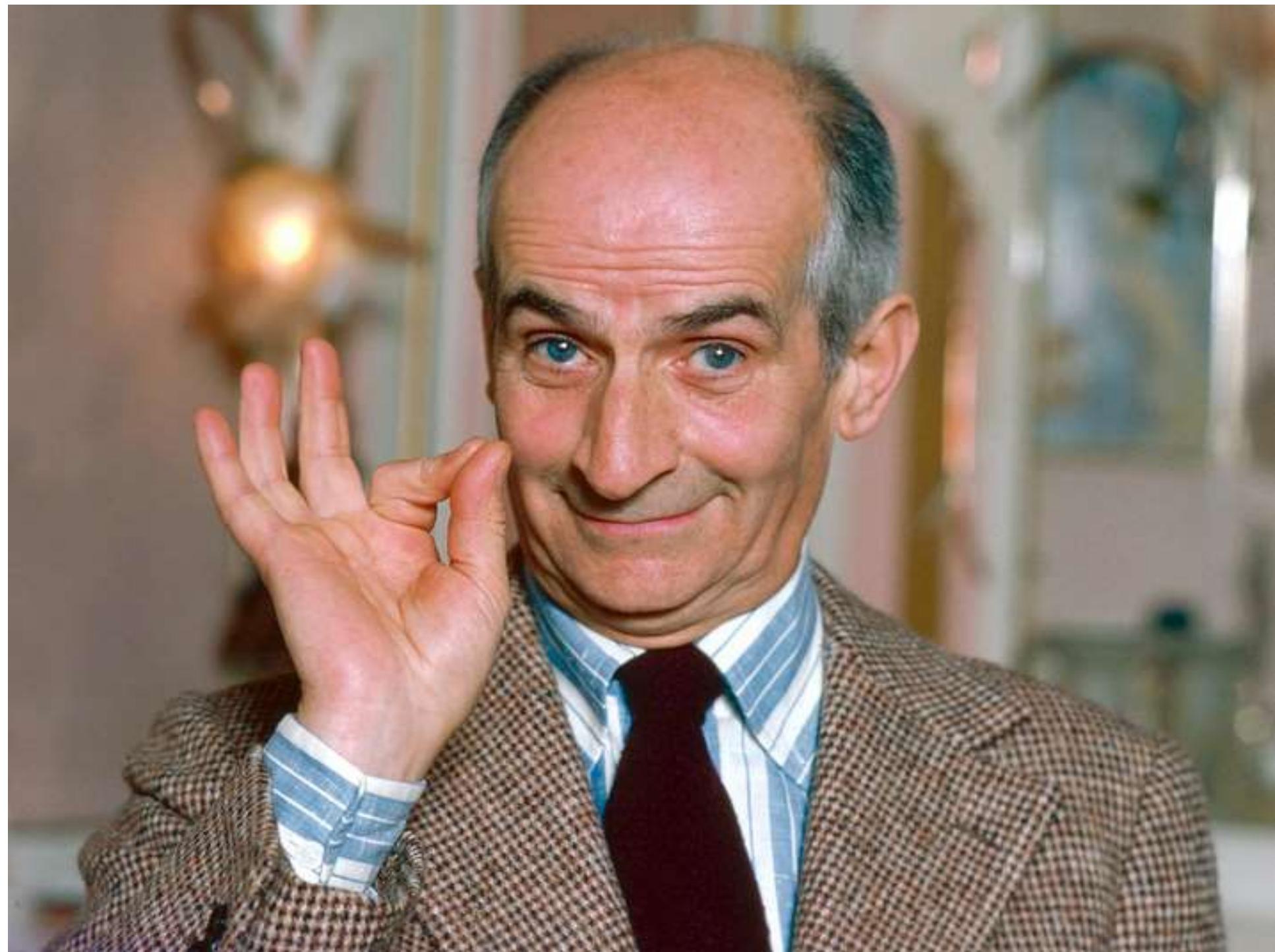
L'introduction de la pensée informatique dès le plus jeune âge pourrait contribuer à réduire l'inégalité Fille/Garçon dans leur rapport aux STEM.



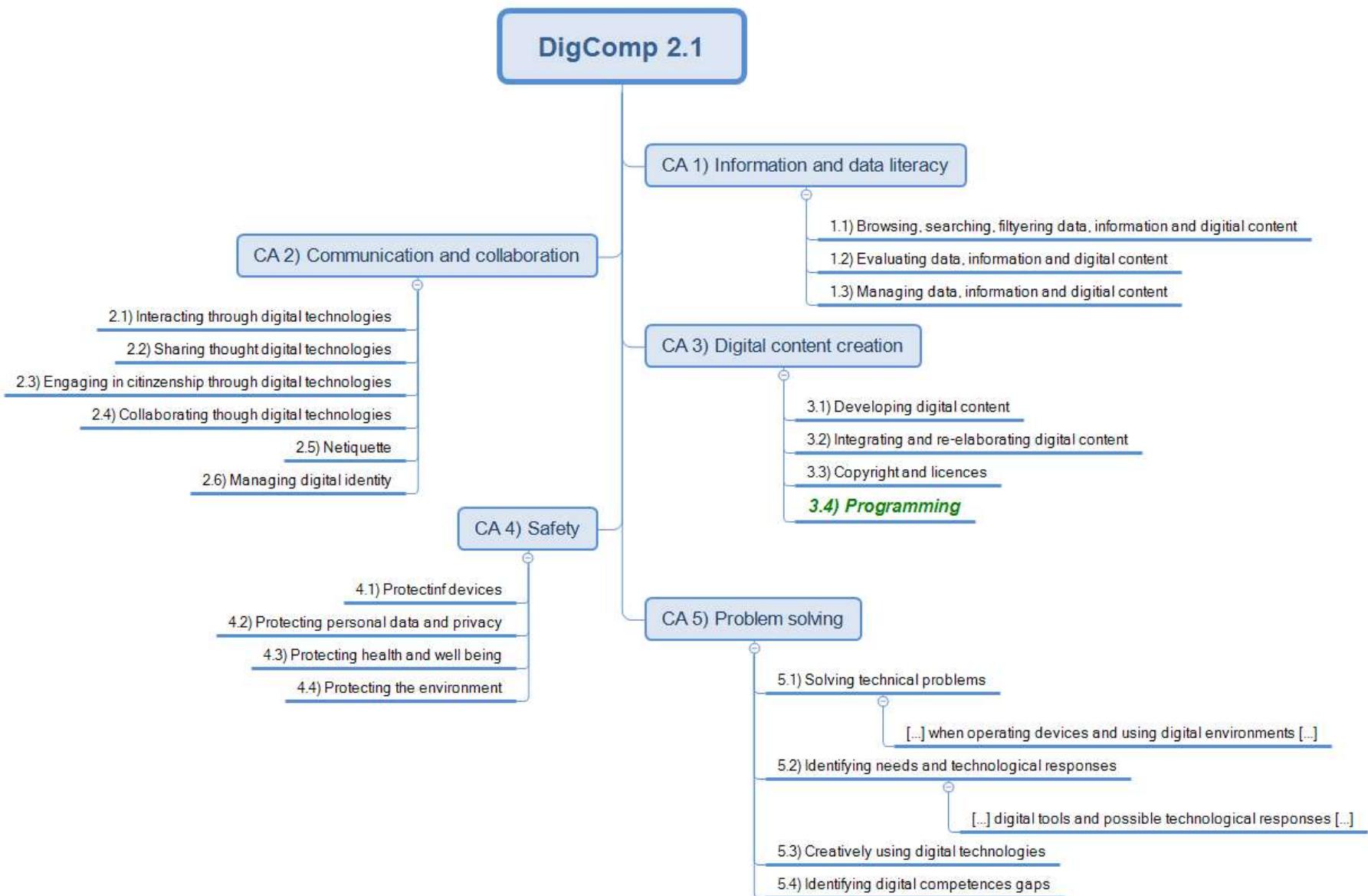
**Et le pacte d'excellence**

- Mme Marie-Martine Schyns, ministre de l'Éducation. – « Pour l'instant, les socles qui définissent les compétences de base attendues à quatorze ans ne contiennent en réalité aucun aspect relatif aux compétences informatiques ou numériques. L'intégration d'une formation numérique incluant des compétences essentielles à tous dans la formation de base de chaque citoyen, donc dans le tronc commun et dans ses socles de compétences, est un objectif important. Le groupe de travail dédié à la formation manuelle, technique, technologique et numérique est chargé d'établir les attendus relatif aux savoirs, savoir-faire et compétences numériques. Parmi ces compétences figurent celles liées à la programmation et à la pensée algorithmique, que certains désignent par le terme «coding». Dans la mesure où le code sera vraisemblablement bientôt généré par les machines elles-mêmes, l'apprentissage de sa logique est pertinent, bien que celui du codage lui-même n'est plus un objectif. C'est donc bien la logique, le système de pensée informatique qui doit être visé, comme souligné lors des auditions menées par le groupe de travail responsable du référentiel. »

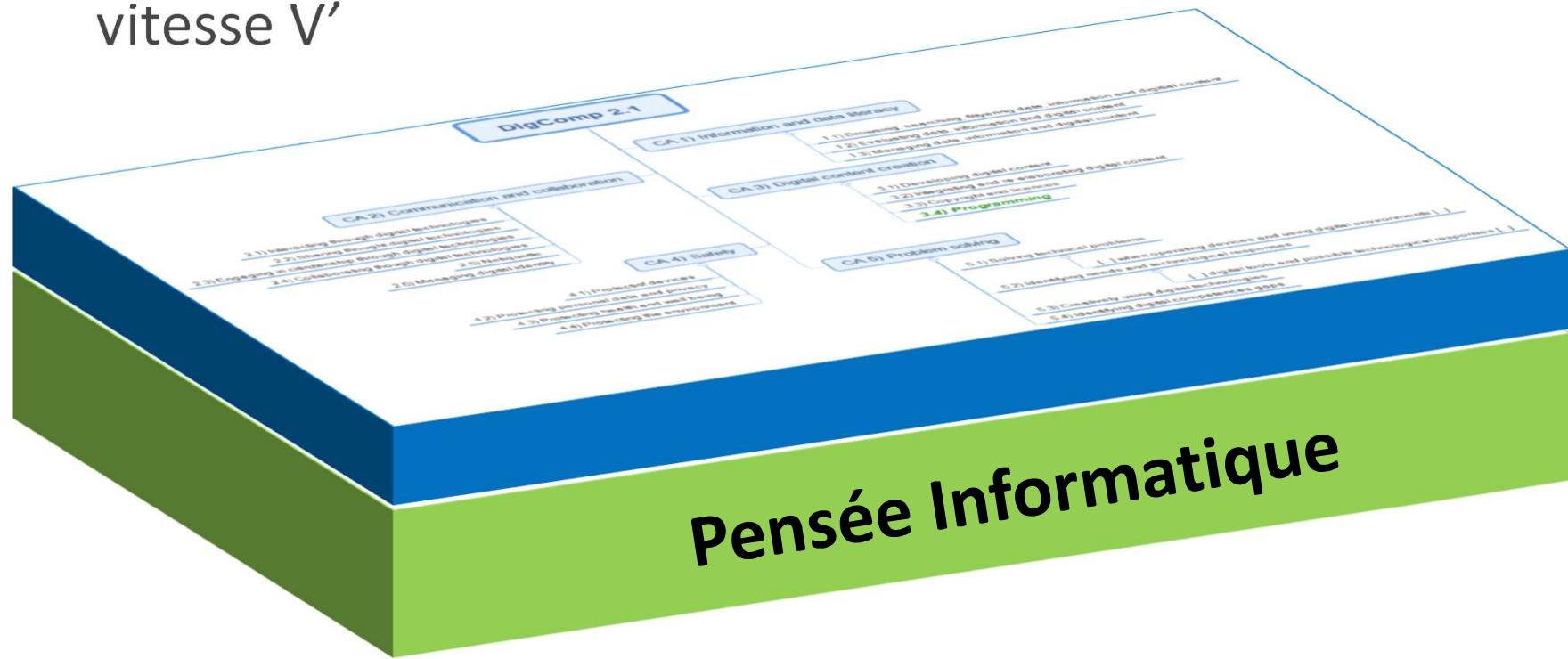
Séance du 20 mars 2018, commission de l'éducation du Parlement de la communauté française



# The Digital Competence Framework for Citizens 2.1



- Travailler ces compétences sans aborder la pensée informatique ne permettra pas aux élèves de contextualiser ces compétences et de les généraliser utilement alors que le champ numérique évolue à vitesse V'



- « Un GT pour la **formation manuelle, technique, technologique et numérique**. Ce groupe envisagera à la fois les apprentissages relevant spécifiquement de cours dédiés à la formation manuelle, technique, technologique et numérique (**incluant la sensibilisation à la pensée algorithmique**, en lien avec la formation mathématique) et ceux qui relèvent du caractère transversal de la dimension « polytechnique » et qui à ce titre peuvent trouver des portes d'entrée dans diverses disciplines (mathématiques, sciences, formation culturelle et artistique....). »

(Page 9)

SOCLE DE SAVOIRS ET DE COMPÉTENCES DU TRONC COMMUN  
CHARTE DES RÉFÉRENTIELS.

*Document approuvé par le gouvernement le 25 octobre 2017*

## OS 1.2 : METTRE EN PLACE UN TRONC COMMUN POLYTECHNIQUE ET PLURIDISCIPLINAIRE RENFORCÉ

- « En outre, le champ du *numérique* s'inscrira, pour une large part, de manière *transversale* au sein des disciplines existantes. A partir du moment où les outils numériques entrent dans la classe, l'éducation *au* numérique ne peut pas être totalement séparée de l'éducation *par* le numérique, car chaque domaine d'étude et de savoirs a une utilisation spécifique du numérique à laquelle les élèves doivent être initiés. »

Avis N° 3 du Groupe central, 7/3/2017, Page 51

## OS 1.2 : METTRE EN PLACE UN TRONC COMMUN POLYTECHNIQUE ET PLURIDISCIPLINAIRE RENFORCÉ

- « [...] tandis que l'éducation *au* numérique pourrait se réaliser au sein des ateliers dédiés aux techniques / technologies (l'initiation au numérique en particulier). **Le numérique ne nécessiterait dès lors pas la création d'un nouveau cours spécifique. On interrogera également l'opportunité - au-delà de la seule littératie numérique – de l'initiation aux « sciences informatiques » notamment algorithmique (en lien avec les mathématiques).** Les ateliers techniques liés au volet technologique pourraient également être l'occasion d'une mise en œuvre de réalisations techniques/numériques, dont la programmation. »

Avis N° 3 du Groupe central, 7/3/2017, Page 52



## **(1) COMPÉTENCES ET CONTENUS NUMÉRIQUES**

Réussir la transition numérique en matière d'éducation suppose tout d'abord de **définir les compétences et les contenus de la “société numérique”** actuelle. Les **compétences numériques** à développer sont des “savoirs, des savoir-faire et des savoir être en action” dont les composantes sont relatives aux aspects sociaux (p. ex : communiquer via les médias socionumériques), informationnels (p. ex. : rechercher de l'information pertinente sur le web) et techniques (p. ex. : connaître et utiliser les différentes fonctions de Twitter) impactés par la culture numérique. Ces compétences supposent tant l'éducation “au” numérique que “par” le numérique. **Les contenus numériques** font partie d'un nouveau domaine de savoirs. Ils sont les produits, les représentations, les opinions, les processus, etc. spécifiques à la société numérique.

En la matière, la distinction conceptuelle entre **éducation par le numérique et éducation au numérique est essentielle**. L'éducation *par* le numérique est le fait d'utiliser le numérique comme un moyen d'enseigner ou de faire apprendre (par ex., créer une leçon en mathématique dans laquelle l'information est présentée et traitée sur tableau blanc interactif). **L'éducation *au* numérique est le fait de considérer le numérique comme un objet d'apprentissage (par ex. : gérer son identité sur le Web, construire son espace d'apprentissage incluant les réseaux sociaux)**

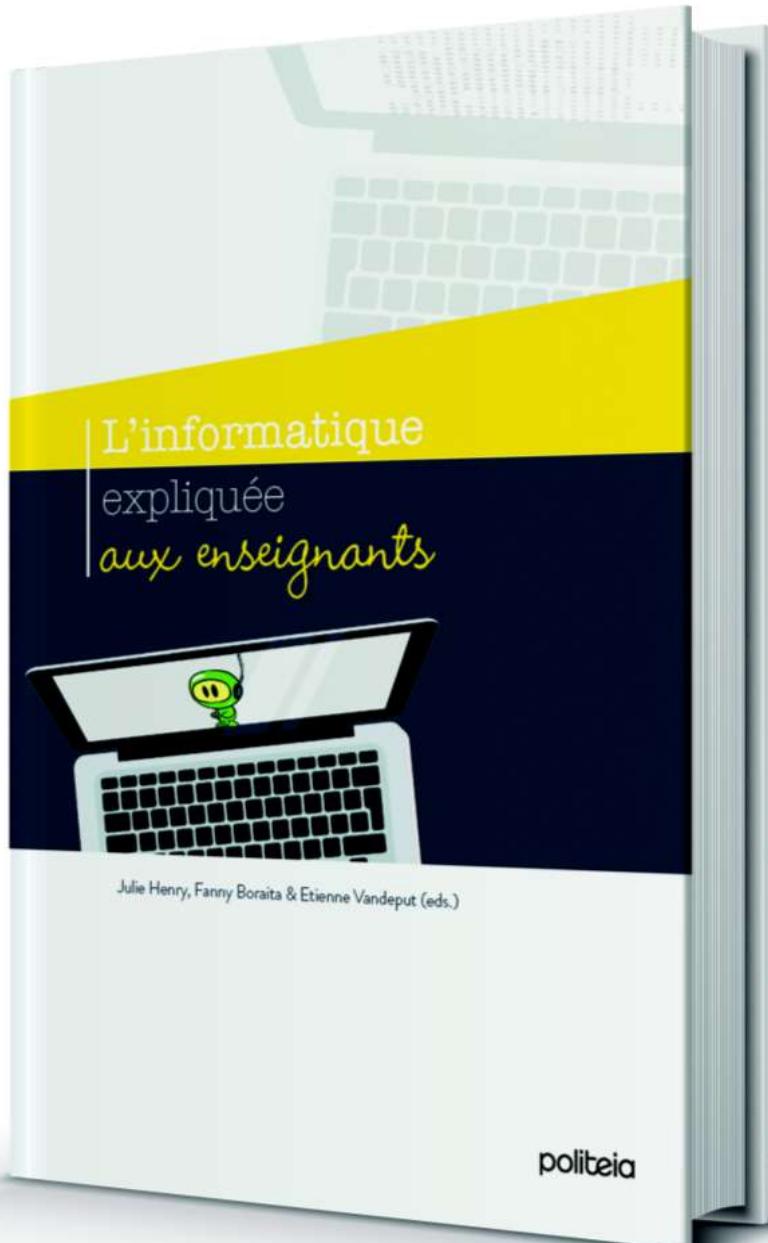
Avis N° 3 du Groupe central, 7/3/2017, Page 90

# Conclusion

- Enseigner la « pensée informatique » est une nécessité et requiert un cours spécifique
  - C'est une question de survie
  - C'est surtout une magnifique opportunité pour le développement de notre société
- Les enseignants doivent être formés de manière spécifique à cette science (et technologie) et la façon de l'enseigner.

# → Conférences

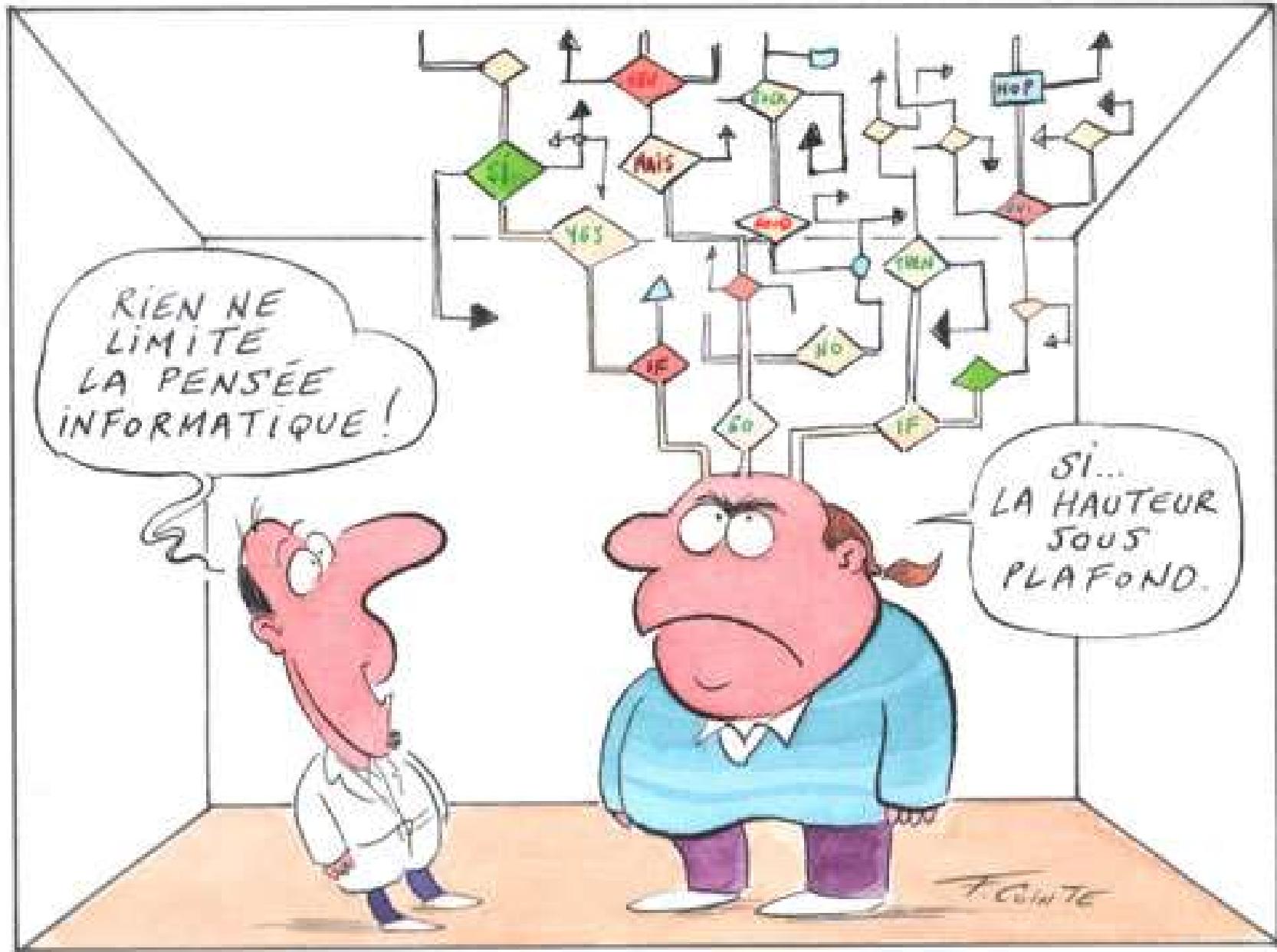
- Stratégie numérique pour l'éducation
- La Fabrique de Logiciels
- Regard sur les compétences développées avec des robots chez les jeunes enfants
- Quelles compétences numériques pour les professeurs de demain
- Développer la pensée informatique et algorithmique au fondamental : quoi de neuf ?
- Et si on faisait aussi de l'informatique... sans ordinateur ?
- Algorithme, robotique, IA... Déconstruire pour éduquer
- Démystifier l'IA : de la théorie à la pratique en classe
- Programmation : de la souris au clavier, une transition en douceur
- Initiation à la pensée informatique et au code avec Micro:Bit
- ...



# L'informatique expliquée aux enseignants

À paraître en  
Septembre 2019





© François Cointe

THANK  
YOU!



UNIVERSITÉ  
DE NAMUR

FACULTÉ  
D'INFORMATIQUE



Vincent ENGLEBERT, Prof.

[www.Englebert.eu](http://www.Englebert.eu)

University of Namur,  
Belgium