



Présentation / Intervention

2025

Open Access

This version of the publication is provided by the author(s) and made available in accordance with the copyright holder(s).

Didactique de l'informatique : panorama (partiel et subjectif) de quelques travaux récents et actuels. Esquisse de quelques perspectives

Jolivet, Sébastien

How to cite

JOLIVET, Sébastien. Didactique de l'informatique : panorama (partiel et subjectif) de quelques travaux récents et actuels. Esquisse de quelques perspectives. In: Journée de formation et d'échanges de la SSIE. Genève. 2025.

This publication URL: <https://archive-ouverte.unige.ch/unige:184593>

SV!A / SS!E / SS!I

Journée de formation et d'échanges, 4 et 5 avril 2025

**Didactique de l'informatique : panorama
(partiel et subjectif) de quelques travaux
récents et actuels.**

Esquisse de quelques perspectives

Sébastien Jolivet

sebastien.jolivet@unige.ch



**UNIVERSITÉ
DE GENÈVE**

INSTITUT UNIVERSITAIRE DE
FORMATION POUR L'ENSEIGNEMENT

Remerciements*

Aux **organisatrices et organisateurs de cette journée** pour m'avoir proposé d'animer cette première demi-journée ;

Aux **collègues (formatrices et formateurs, chercheuses et chercheurs)** pour tous les échanges, travaux menés, en cours et à venir, qui ont largement contribué à nourrir les réflexions présentées ici ;

Aux **enseignantes et enseignants d'informatique en formation**, actuellement ou les années précédentes, dont l'accompagnement permet un lien maintenu avec les réalités des classes et dont les pratiques, les questionnements, et l'imagination sont à la fois sources de réflexion, de questionnements et d'inspiration.

**La responsabilité des éventuelles "bonnes" idées et réflexions intéressantes est partagée avec toutes ces personnes, la responsabilité des erreurs ou imprécisions m'incombe entièrement.*

Qui suis-je ?

Chargé d'enseignement en didactique de l'informatique à l'IUFE - UNIGE. Chercheur associé au TECFA (UNIGE) et au LIP6 (Sorbonne Université).



Recherche

Thème et cadres théoriques

- ❖ Didactique des mathématiques et de l'informatique (théorie anthropologique du didactique, approche instrumentale)
- ❖ Environnements Informatiques pour l'Apprentissage Humain (EIAH)
- ❖ Ressources (conception, indexation, analyse)
- ❖ Pensée informatique et pensée mathématique

Activités et projets actuels

- ❖ Encadrement d'une thèse sur la décision adaptative des rétroactions dans un environnement d'apprentissage de Python (rétroactions épistémiques, apprentissage par renforcement)
- ❖ Environnement de *distributed pair programming*
- ❖ Modélisation du savoir dans le domaine de la programmation
- ❖ Computational thinking and mathematics
- ❖ CompréhenMath (didactique du français, des mathématiques et informatique)
- ❖ Encadrement de différents mémoires de master / stage licence

Enseignements

Avant

- ❖ Enseignant de mathématiques, secondaire I français
- ❖ Formateur d'enseignant.es du primaire et du secondaire (mathématiques, numérique éducatif, méthodologie de la recherche...)
- ❖ Enseignements dans différents Master (didactique ; EIAH)

Actuellement

- ❖ Formation des enseignant.es d'informatique du secondaire à Genève (cours et séminaire recherche de didactique de l'informatique ; suivi de stage)
- ❖ Master EdTech, Sorbonne Université (UE SHS pour les EdTech ; UE scénarisation ; encadrement de stages)

Organisation de cette présentation

- ❖ Introduction : une analyse du contexte
- ❖ Panorama de quelques travaux
- ❖ Focus sur un de mes travaux et ses applications : modéliser l'activité de programmation
- ❖ Esquisse de quelques perspectives
- ❖ Ressources : bibliographie commentée et rendez-vous à venir

Organisation de cette présentation

- ❖ Introduction : une analyse du contexte
- ❖ Panorama de quelques travaux
- ❖ Focus sur un de mes travaux et ses applications : modéliser l'activité de programmation
- ❖ Esquisse de quelques perspectives
- ❖ Ressources : bibliographie commentée et rendez-vous à venir

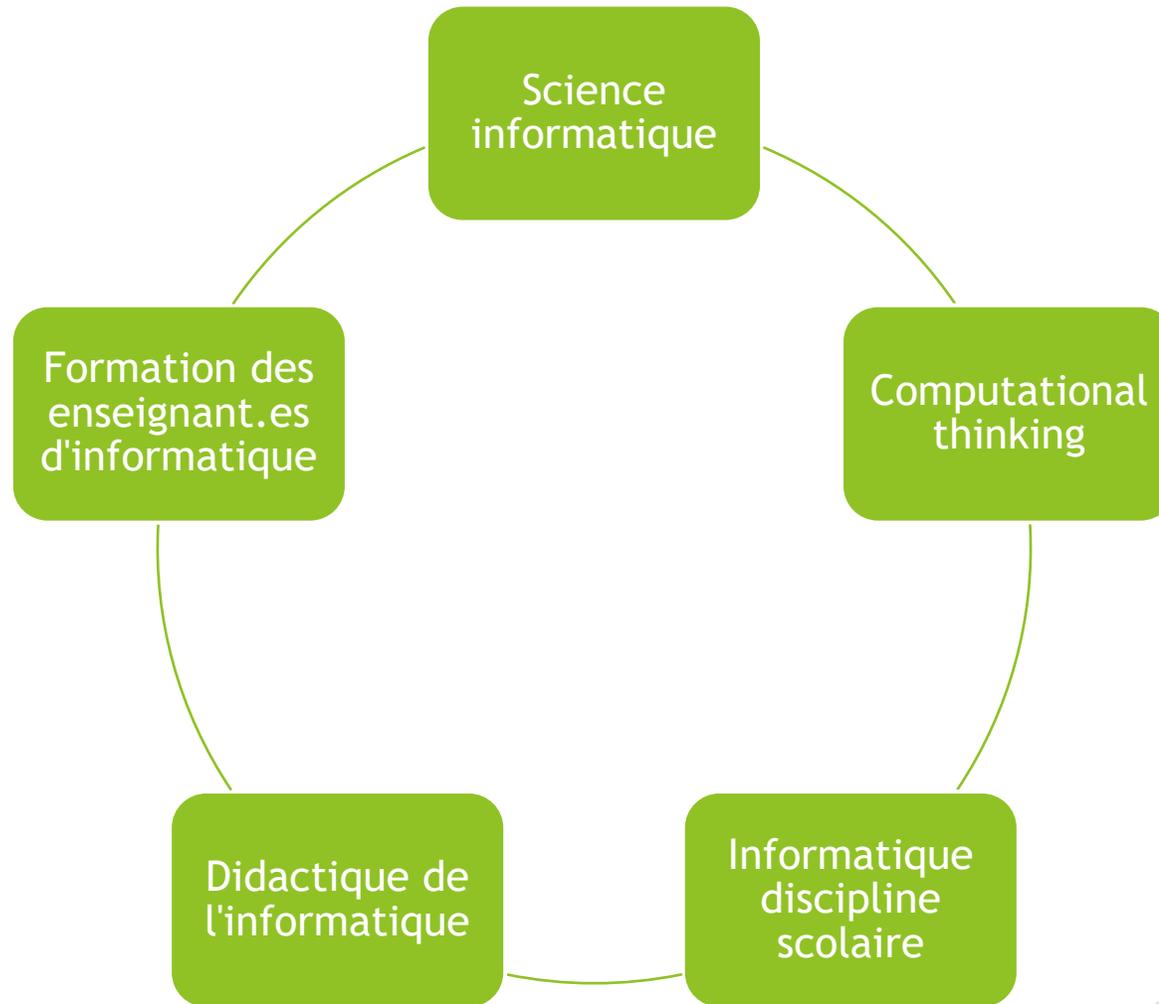
Ce que cette présentation ne va pas être :

- ❖ Un cours de didactique de l'informatique
- ❖ Un catalogue de ressources ou de références bibliographiques
- ❖ L'analyse d'un contexte institutionnel particulier
- ❖ Une réponse à « comment former les enseignant.es d'informatique ? »

Introduction : une analyse du contexte



Éléments caractéristiques du contexte



Contexte : science informatique

La **science informatique**, une description épistémologique assez largement partagée.

« À la question « Qu'est-ce que l'informatique ? », nous sommes souvent tentés de donner une réponse de la forme « L'informatique est ... ». Ainsi, l'informatique est souvent définie comme la science des algorithmes, la technique du traitement de l'information, la partie constructive des mathématiques, une branche de l'électronique, une physique du signe, ... Toutes ces réponses sont imparfaites, car partielles. Au lieu de tenter une réponse de la forme « L'informatique est ... », il semble plus sage de tenter des réponses de la forme « En informatique, il y a ... ». (...)

Quatre concepts - algorithme, machine, langage et information - semblent suffisants pour couvrir l'ensemble de **ce que nous appelons « informatique »**. » (Dowek, 2011)

A partir de maintenant l'essentiel de mon propos se concentrera sur la dimension programmation, pour des raisons de temps et d'autres sur lesquelles je reviendrais en conclusion, ce qui n'enlève rien à l'intérêt des autres domaines !

Contexte : computational thinking

Computational Thinking : un concept qui admet des définitions et des délimitations diverses.

Penser informatiquement, c'est reformuler un problème apparemment difficile en un problème que nous savons résoudre, par exemple par réduction, par plongement, par transformation ou par simulation.

Penser informatiquement, c'est penser récursivement (...).

Penser informatiquement, c'est utiliser l'abstraction et la décomposition quand il s'agit d'affronter une tâche large et complexe ou quand il s'agit de concevoir un système large et complexe (...).

Penser informatiquement c'est envisager le pire événement en pensant en termes de prévention, de protection et de récupération (...).

Extraits de Wing, 2006 (version française)

Contexte : computational thinking

Computational Thinking : un concept qui admet des définitions et des délimitations diverses.

La pensée informatique (PI) est un processus de résolution de problèmes qui comprend (sans s'y limiter) les caractéristiques suivantes :

- a. Formuler les problèmes d'une manière qui nous permet d'utiliser un ordinateur et d'autres outils pour les résoudre.
- b. Organiser et analyser logiquement les données
- c. Représenter les données par des abstractions telles que des modèles et des simulations.
- d. Automatiser les solutions grâce à la pensée algorithmique (une série d'étapes ordonnées).
- e. Identifier, analyser et mettre en œuvre des solutions possibles dans le but d'obtenir la combinaison d'étapes et de ressources la plus efficace et la plus efficiente possible.
- f. Généraliser et transférer ce processus de résolution de problèmes à une grande variété de problèmes.

Contexte : informatique, discipline scolaire

- ❖ La place et la délimitation de l'informatique dans l'enseignement scolaire, des approches diverses
 - Ce qui est ou non inclus dans le périmètre de la discipline
 - Une discipline qui peut être enseignée, ou au moins abordée, dans différentes disciplines scolaires
- ❖ Des processus transpositifs encore largement en cours

Contexte : informatique, discipline scolaire

- ❖ (...) Pour les discuter, je commencerai (partie 2) par revenir sur les différences de statut entre informatique scolaire et mathématiques scolaires et sur les différences qui en découlent pour leurs didactiques ; je discuterai ensuite du problème, crucial pour l'informatique, mais qui se pose pour toutes les didactiques, d'identification des contenus (partie 3) ; (...).
- ▶ (...) Mais cela ne règle pas la question de fond sur laquelle repose la remarque de Chevallard : un didacticien de l'informatique doit-il ne s'intéresser qu'à des contenus identifiés comme informatiques par des informaticiens (qui peuvent d'ailleurs ne pas être d'accord entre eux pour savoir si faire un collier de perles en suivant un algorithme, en moyenne section de maternelle, est ou non un préapprentissage de l'informatique et de l'algorithmique) ? Ou bien doivent-ils s'intéresser à ce qui est désigné par les programmes scolaires ou les manuels ? Ou encore à ce que les élèves, les enseignants ou les parents identifient comme tel ?
- ▶ Par exemple, apprendre à taper au clavier n'est certainement pas un contenu que beaucoup d'informaticiens considèrent comme relevant de leur discipline. Mais les acteurs du système scolaire peuvent l'identifier comme tel (...).

(Fluckiger, 2020)

Contexte : la didactique de l'informatique

D'une manière générale la didactique d'une discipline n'existe, ou au moins ne vie, que quand il s'agit d'une discipline scolaire, ainsi la **didactique de l'informatique**, ou **computer science education**, en tant que champ scientifique, est directement impactée par les opportunités et contraintes définies par les éléments précédents, à la fois dans son existence même mais aussi dans les sujets dont elle se préoccupe.

- Une histoire ancienne mais marquée par des discontinuités importantes
- Une communauté encore peu structurée

Guzdial, M., & Du Boulay, B. (2019). The History of Computing Education Research. In S. A. Fincher & A. V. Robins (Éds.), *The Cambridge Handbook of Computing Education Research* (1^{re} éd., p. 11-39). Cambridge University Press. <https://doi.org/10.1017/9781108654555.002>

Declercq, C. (2021). Didactique de l'informatique : Une formation nécessaire [Pdf]. *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, 28(3). <https://doi.org/10.23709/STICEF.28.3.8>

Deux spécificités de la programmation

A la différence de nombreuses autres situations, dans de nombreuses disciplines scolaires, y compris dans d'autres domaines de l'informatique :

- ▶ Ce n'est pas l'élève qui fait directement mais il doit expliquer à une machine comment faire (voir par exemple Declercq, 2021)
- ▶ Potentielle rupture d'une part du « contrat didactique dominant » dans la mesure où, potentiellement, l'élève n'est pas ou peu dépendant de l'enseignant pour déterminer si ce qu'il a fait est correct ou non (ou au moins réalise l'objectif visé) et dispose de plus, souvent, d'indications fournies par l'environnement (IDE par exemple) lui permettant d'agir sur son travail en cas d'erreur.

Panorama de travaux

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the right side of the slide, creating a modern, layered effect. The text 'Panorama de travaux' is positioned on the left side of the slide in a clean, sans-serif font.

Petit état des lieux sur les travaux de recherche en lien avec la didactique de l'informatique

Pour identifier des dynamiques et questions vives en lien avec l'apprentissage de l'informatique, j'ai examiné quelques

- ▶ Thèses, récentes et en cours : un marqueur des grandes thématiques abordées dans le domaine. D'un point de vue académique elles peuvent être inscrites en informatique, en sciences de l'éducation ou dans d'autres domaines ; mais elles sont toutes en lien avec l'enseignement ou l'apprentissage de l'informatique
- ▶ Travaux universitaires, en particulier dans le cadre de la formation des enseignants qui sont des indicateurs de préoccupations du « terrain » et dans ce sens permettent d'identifier des questions qui, le plus souvent, n'ont pas de réponses satisfaisantes dans les résultats actuels de la didactique de l'informatique

Pour enrichir cet état des lieux il serait notamment intéressant d'étudier les projets de recherche en cours et les nombreuses ressources pour enseigner l'informatique (Modulo ; Vittascience ; ressources EPFL et/ou HEP-VD ; EIAH de travail de la programmation ; filius pour les réseaux ; ressources de type informatique débranchée ; etc.). Ces éléments ne sont pas dans cette présentation.

Éléments d'état des lieux de la recherche en didactique de l'informatique : thèses

Thèses soutenues (par année de soutenance) :

- Goletti, O. (2025). Improving Tutors' Practice by Integrating Explicit Instructional Programming Strategies in Introductory Programming Courses at University [PhD Thesis, UCLouvain]. <http://hdl.handle.net/2078.1/299072>
- Léonard, M. (2024). *Approche didactique et instrumentale de la pensée informatique : Focus sur le concept de motif* [PhD Thesis, Université de Lille]. <https://theses.hal.science/tel-04793189/>
- Plumettaz-Sieber, M. (2024). Institutionnalisation des savoirs informatiques lors du débriefing. Le cas de l'apprentissage avec le jeu Programming Game [PhD Thesis, Université de Genève]. <https://doi.org/10.13097/ARCHIVE-OUVERTE/UNIGE:175760>
- Branthome, M. (2023). Apprentissage de la programmation informatique : Analyses et ressources pour accompagner la transition collège-lycée [PhD Thesis, Université de Bretagne occidentale-Brest]. <https://theses.hal.science/tel-04397329/>
- Legrand, J.-M. (2023). Construction d'un concept à l'interface entre informatique et algèbre élémentaire au collège : La variable dans un rôle de paramètre [PhD Thesis, Nantes Université]. <https://theses.hal.science/tel-04578625/>
- Mari, E. (2022). Potentialités de la robotique pédagogique pour le développement de connaissances spatiales à l'école [PhD Thesis, Aix Marseille Université (AMU), Marseille, FRA.]. <https://amu.hal.science/tel-03872631/>
- Henry, J. (2022). Mesurer la compréhension en informatique à partir des représentations erronées [PhD Thesis, Université de Namur]. https://researchportal.unamur.be/files/68924221/2022_HenryJ_these.pdf

Éléments d'état des lieux de la recherche en didactique de l'informatique : thèses

Thèses en cours (par ordre alphabétique) :

Brender, J. : A Systematic Approach to Study and Teach Debugging for Computational Problem-Solving across Computational Environments (EPFL - HEPV, Lausanne)

Chane-Lune, S. : Conception et analyse d'un référentiel de compétences en programmation pour le lycée (LIM, La Réunion)

Colin, J. : Conception et évaluation d'un environnement de distributed pair programming dédié à des utilisateurs novices (IRIT, Toulouse) <http://www.atief.fr/these/these-colin>

Kirouchenassamy, B. : Système intelligent de décision de feedbacks au sein de plateformes d'entraînement à la programmation : une approche apprentissage par renforcement pour approximer une politique de décision de feedbacks adaptatifs (LIP6, Paris) <http://www.atief.fr/these/these-kirouchenassamy>

Parriaux, G. : Computer science in primary schools and teacher education (EDA, Paris) <https://theses.fr/s212397>

Orientation des travaux de thèse

Sur 12 thèses (soutenues et en cours) on peut identifier les thèmes ou objets d'études suivants :

- ❖ Programmation : 10
- ❖ Outils / environnement d'apprentissage : 6 (distributed pair-programming, AlgoPython, Pyrates, Citizen Code, programming game...)
- ❖ Place significative des enseignants : 4
- ❖ Lien avec une autre discipline : 2 (mathématique / connaissances spatiales)
- ❖ Compréhension, compétence... : 2
- ❖ Focalisation sur un objet /une notion particulière : 2 (variable ; motif)
- ❖ Robotique : 1
- ❖ En lien avec IA : 1

Éléments d'état des lieux de la recherche en didactique de l'informatique : préoccupation d'enseignants

Quelques exemples, par thématiques :

- **Analyser et évaluer des ressources d'apprentissage** (numériques, tangibles) : serious-game d'apprentissage de la programmation ; micro:bit ; analyse de l'intégration d'EIAH pour l'enseignement de la boucle
- **Approches pour enseigner la programmation** : pair-programming conditions d'utilisation ; constitution des paires) ; intégration d'IA génératives ; organisation et enjeux en lien avec le *mode projet*
- **Moyens pour aider les enseignants** : IAG pour aider à la conception de séquence ; rendre visible les parcours d'apprentissage des élèves
- **Comprendre des caractéristiques de l'enseignement de l'informatique** : un "meilleur" ordre pour aborder les notions de programmation ? ; focalisation sur la notion de variable (sa place dans différents contextes)

Focus sur un travail en cours et ses
exploitations : modéliser l'activité de
programmation

Question initiale : ((Apprendre à Apprendre à) programmer, c'est quoi ?

Programming subtasks	Basic processes	Knowledge domains	Mental representations	External representations
Understanding the problem	COMPOSITION ↓ ↑ COMPREHENSION	Domain knowledge (e.g., statistics, banking)	Situation model	Requirements document Specifications document
Design		Design strategies Programming Algorithms and methods Design language	Solution model Plan representation	Design document
Coding		Programming language Programming conventions	Program representation	Code
Maintenance		All knowledge domains Debugging, testing strategies Frequent kinds of error	All representations	All documents

Figure 1: The tasks of programming.

- Pennington, N., & Grabowski, B. (1990). The tasks of programming. In Psychology of programming (p. 45-62). Elsevier.

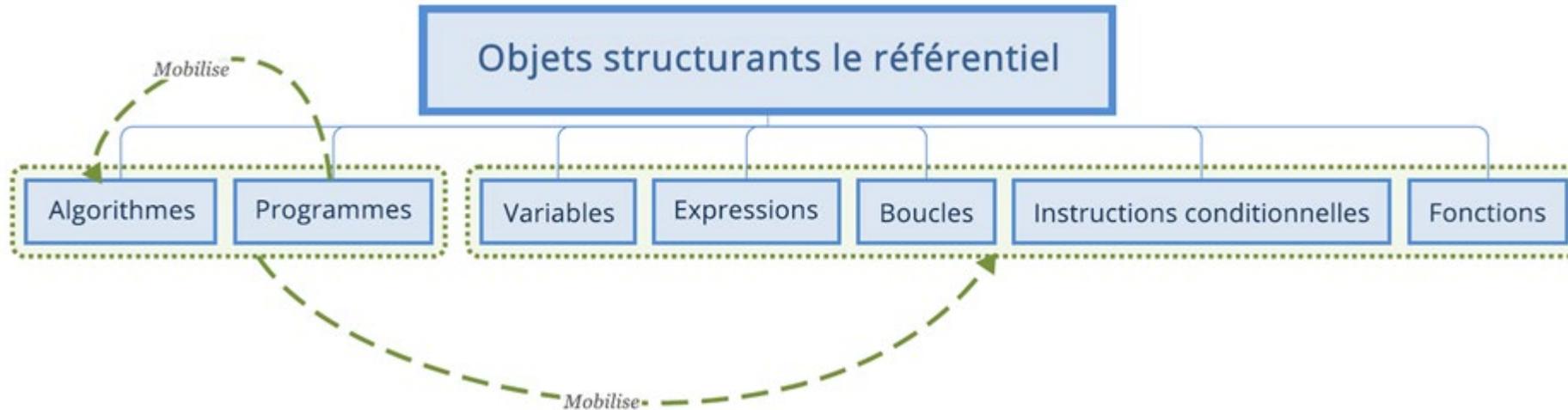
<https://www.sciencedirect.com/science/article/pii/B9780123507723500082>

Le travail de base : modélisation de l'activité de programmation

- ❖ Une approche basée sur l'approche praxéologique de la théorie anthropologique du didactique
- ❖ Principe : décrire l'activité **humaine** dans le cadre d'une activité de programmation
- ❖ Méthode : étude d'éléments issus du savoir de référence, de curriculums, de moyens d'enseignements...
- ❖ Choix d'une structuration et identification d'un certain nombre de types de tâches
- ❖ Mise en relation de différents types de tâches

Travail réalisé en collaboration avec Patrick Wang (HEPL) ; Eva Chouaki (IUFE - UNIGE) ; Anne-Claire Gobard (Lycée Askler, France)

Structuration de base du référentiel



https://link.infini.fr/ref_prog



Une structuration systématique

Types de tâches relatifs aux instructions conditionnelles

1 Produire ou modifier une instruction conditionnelle

1.1 Produire une instruction conditionnelle

1.2 Modifier une instruction conditionnelle

2 Analyser une instruction conditionnelle

Types de tâches relatifs aux algorithmes

1 Produire ou modifier un algorithme...

1.1 Produire un algorithme

1.2 Modifier un algorithme

2 Analyser un algorithme ..

2.1 Analyser un algorithme sans se référer à une tâche particulière

2.2 Analyser un algorithme en référence à une tâche t qu'il doit réaliser

Extrait sur les boucles

1.1 Produire ou modifier une boucle bornée

1.1.1 Produire une boucle bornée

1.1.1.1 Concevoir une boucle bornée

Déterminer la séquence d'instructions
1.1.1.1.1 à exécuter à chaque itération
d'une boucle bornée

Déterminer si une (ou plusieurs)
1.1.1.1.1 instructions du corps d'une boucle bornée
utilise(nt) l'itérateur

Déterminer le nombre
1.1.1.1.2 d'itérations à réaliser
pour une boucle bornée

1.1.1.1.3 Déterminer l'itérable
pour une boucle bornée

1.1.2 Modifier une boucle bornée

Ajouter, supprimer, modifier
1.1.2.1 une ou des instructions
à une boucle bornée existante

1.1.2.2 Modifier le nombre
d'itérations d'une boucle bornée

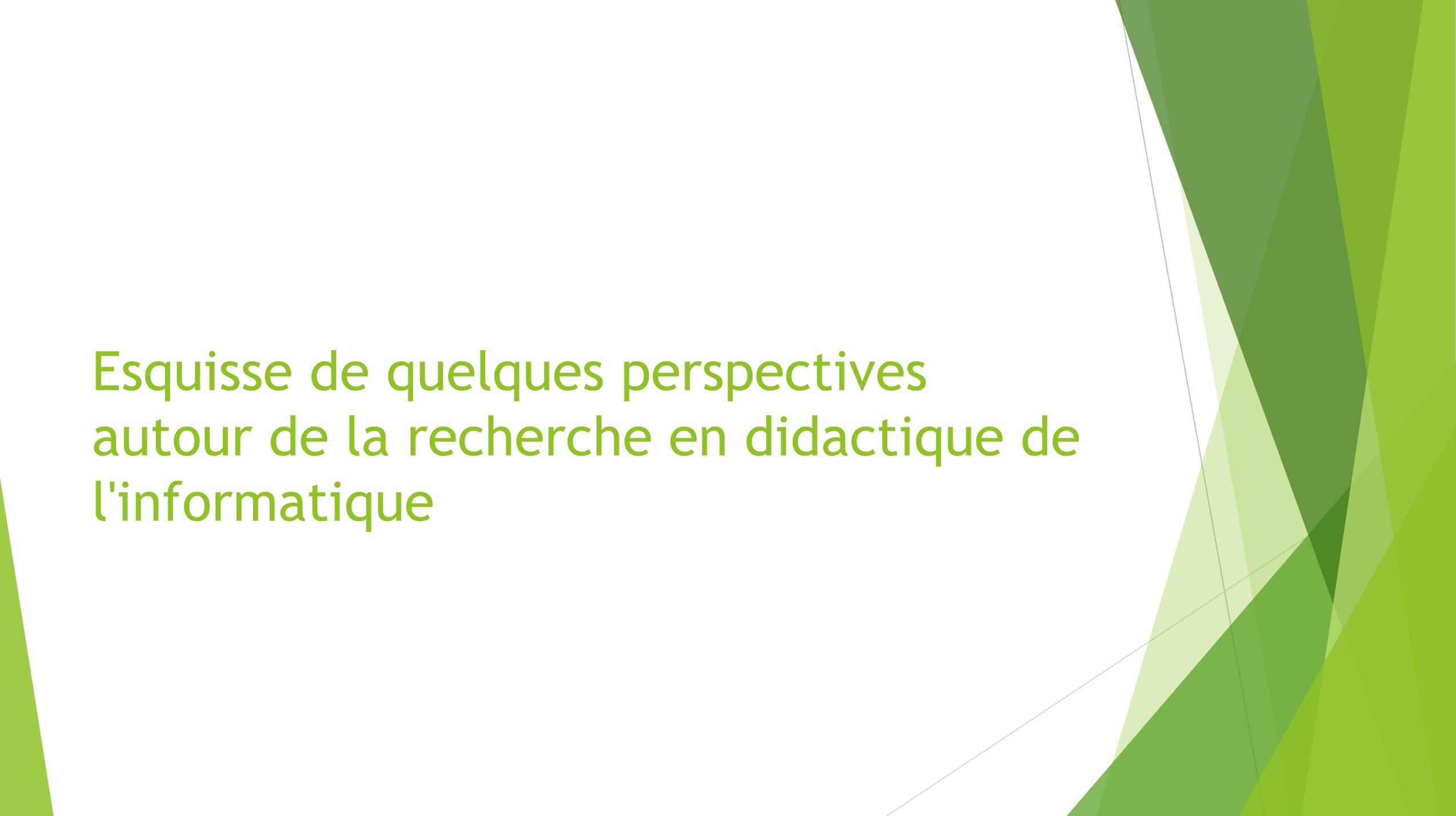
Quelques exploitations, en cours ou potentielles

Du point de vue de la recherche, le référentiel de types de tâches présentés précédemment est, ou pourrait être, exploité pour :

- ❖ Analyser et décrire des ressources d'enseignement (exercice, EIAH,...)
- ❖ Modéliser et concevoir des rétroactions
- ❖ Produire une typologie d'erreurs en programmation
- ❖ Définir des parcours d'apprentissage

Du point de vue de la formation des enseignants :

- ❖ Support possible pour le travail sur les plans d'étude
- ❖ Mise en évidence de la complexité de certaines tâches perçues comme "anodines" (ou d'un bloc) par certain.es enseignant.es
- ❖ Aide à la conception de plan de cheminement / séquence
- ❖ Etc.

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the right side of the slide, creating a modern, layered effect. The text is positioned on the left side of the slide, set against a plain white background.

Esquisse de quelques perspectives autour de la recherche en didactique de l'informatique

Une carte largement inexplorée

- ❖ Plan des villes disponibles du point de vue des savoirs « savants »... beaucoup moins du point de vue de son enseignement et de la didactique
- ❖ Des routes à construire pour penser un enseignement « global » de l'informatique
- ❖ Des visites / voyages thématiques à construire pour intégrer des questions sociétales, éthiques, etc.



Exemple de zones d'ombre sur le cas de la programmation

Mêmes si les villes *langage* et *algorithme* sont sans doute les deux plus explorées du point de vue enseignement / apprentissage / didactique elles restent largement inexplorées de ces points de vue. Voici par exemple les aspects de la programmation identifiés comme absents à l'issue de la présentation de plusieurs travaux de recherche en cours lors d'un atelier proposé aux RJC-EIAH 2024.

Quels aspects de la programmation sont absents ?

- Programmation bas niveau (langage machine, assembleur)
- Programmation déclarative (langage fonctionnel ou logique)
- Programmation impérative orientée objet
- Programmation Web

- Technique de production de programmes (procédés, IDE, framework, ...)
- Modélisation (objet/UML, BD, IHM, 3D, ...)

- Globalement, les activités sont davantage orientées exercices/puzzle que développement d'une application

A quoi peut bien servir la didactique (de l'informatique) ? Un point de vue...

Ne répond pas à la question « comment enseigner l'informatique ? » mais **peut y contribuer** en :

- Transformant des *questions d'enseignant.es* en *questions de recherche...* et en essayant d'apporter des éléments de réponse
- Proposant des moyens d'observation et d'analyse des pratiques et de leurs effets
- Contribuer à la définition des curriculums et / ou à leur étude
- Penser et accompagner les transitions entre niveaux d'enseignement
- Proposer des moyens d'analyse de ressources
- Contribuer à la production de ressources et d'outils en s'assurant qu'ils soient, notamment, fondés sur des connaissances didactiques
- Fournir des connaissances expertes pour différents besoins (système mobilisant des IA par exemple)

Pourquoi faire de la recherche en didactique de l'informatique ?

- ❖ Continuer à produire des connaissances scientifiques notamment pour (ré)interroger les *croyances, certitudes, convictions empiriques, validations du type « sourire des élèves», etc.*, relatives à l'informatique, son apprentissage et son enseignement et apporter des éléments de réponse sur la base d'approches scientifiques
- ❖ Enjeux pour la formation des enseignants d'informatique
- ❖ Enjeux sociétaux, environnementaux et éthiques (questions liées au genre, "sobriété" numérique, avoir les moyens de comprendre le monde dans lequel on vit) qui ne sont pas de la seule responsabilité de l'informatique mais pour lesquels l'informatique a sa part

Pourquoi faire de la recherche en didactique de l'informatique ?

Un certain consensus se dégage concernant les questions de recherche [à explorer], mais l'enseignement de l'informatique est un domaine nouveau, et une grande partie de la recherche sous-jacente n'en est qu'à ses débuts ; la fiabilité des preuves actuelles peut peut-être être restreinte en raison des limites des études à partir desquelles elles ont été obtenues. Ces études ont souvent été menées sur de courtes périodes et avec un petit nombre d'apprenants, dans un cadre informel plutôt que dans une salle de classe, et sans moyens solides pour évaluer les interventions avant et après leur mise en œuvre. Néanmoins, les données dont nous disposons nous donnent un point de départ. (Traduction de Waite, J., & Sentance, S., 2021, p.5)

Exemples de questions d'enseignant.es

- ❖ Dans quel ordre faut-il aborder les différentes notions de programmation ?
- ❖ Quel est, ou quels sont, les meilleurs langages pour apprendre à programmer ?
- ❖ Est-ce utile ou négatif de commencer à programmer dans le cadre de la programmation par blocs ?
- ❖ Apprend-on quelque chose, ou qu'apprend-on, quand on utilise des environnements du type code.org ; Blockly ; etc. ?

Qui peuvent permettre de définir autant, et même plus, de questions de recherche.

Quelques implications de ces questions pour la recherche

- ❖ Que mesure-t-on et comment mesure-t-on ? Exemple de mesure des compétences en programmation :

https://web.eecs.umich.edu/~weimerw/data/FSE2021_To_Read_or_to_Rotate/Participant%20Forms/SCS1_Info_Sheet.pdf

<https://dl.acm.org/doi/pdf/10.1145/2960310.2960316>

- ❖ Veut-on mesurer des effets ou comparer des effets ?
- ❖ Existe-t-il des dispositifs d'enseignement (informatique débranchée, pair-programming...) plus ou moins adaptés à des contenus d'enseignement ?
- ❖ Existe-t-il en programmation des ruptures épistémologiques comme le passage de l'arithmétique à l'algèbre ou de la géométrie instrumentée à la géométrie déductive ?

Du constat aux perspectives : pour étendre la connaissance de la carte

Nécessiter de développer des interactions entre spécialistes du savoir de référence dans des domaines tels que les réseaux, les BDD, etc. ; des enseignant.es de ces domaines et des didacticien.nes.

Du constat aux perspectives : cadres théoriques et méthodologiques

- ❖ Quels cadres théoriques et méthodes exploiter, adapter ou construire pour mener des recherches en didactique de l'informatique ?
 - Cadres issus de didactiques d'autres disciplines : théorie des champs conceptuels ; théorie des situations didactiques ; théorie anthropologique du didactique. Cadres et registres ; dialectique outil - objet (Duval ; Douady)
 - Travaux sur l'approche instrumentale (Rabardel ; Trouche) ; sur l'orchestration (Drijvers ; Trouche) ; sur l'approche documentaire (Trouche & Geudet) ?
- ❖ Lesquels exploiter en formation des enseignant.es d'informatique ?

Du constat aux perspectives : des recherches à mener dans des contextes interdisciplinaires ?

- ❖ EIAH et didactique de l'informatique
- ❖ Quels effets des liens entre mathématiques en informatique ?
- ❖ Apports et articulations avec les travaux sur les artefacts tangibles
- ❖ Les approches STEM (science, technology, engineering, and mathematics) - STEAM (science, technology, engineering, art and mathematics)
- ❖ Informatique et créativité
- ❖ Regards croisés dans le contexte de la formation

Pour conclure, quelques rendez-vous
à venir et ressources commentées

Quelques rendez-vous à venir

- Atelier [APIMU](#), juin 2025, Lille
- Atelier [EIAH et didactique\(s\)](#), juin 2025, Lille
- [Colloque etic](#) (Axe 1 : L'entrée dans les cultures numériques dans et hors l'École) , octobre 2025, Le Mans
- [Didapro](#), février 2026, Grenoble
- RJC-EIAH, juin 2026
- Appel à articles, [Numéro spécial STICEF](#) - Cadre d'usage et de fonctionnement des IA génératives (IAG) en éducation

Bibliographie

- ❖ Declercq, C. (2021). Didactique de l'informatique : Une formation nécessaire [Pdf]. *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, 28(3).
<https://doi.org/10.23709/STICEF.28.3.8>
- ❖ Dowek, G. (2011). Les quatre concepts de l'informatique. Sciences et technologies de l'information et de la communication en milieu éducatif: Analyse de pratiques et enjeux didactiques., 21-29.
<https://edutice.hal.science/edutice-00676169>
- ❖ Fluckiger, C. (2021). La didactique de l'informatique est-elle la science des conditions de diffusion des connaissances informatiques? In A. Chesnais & H. Sabra (Éds.), Actes du séminaire national de didactique des mathématiques 2020 (p. 29-46). IREM de Paris. <https://hal.archives-ouvertes.fr/hal-03545511>
- ❖ ISTE. (2011). Operational Definition for Computational Thinking.
https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Bibliographie

- ❖ Sentance, S., Waite, J., Yeomans, L., & MacLeod, E. (2017). Teaching with physical computing devices : The BBC micro:bit initiative. Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 87-96.
<https://doi.org/10.1145/3137065.3137083>
- ❖ Tchounikine, P. (2017). Initier les élèves à la pensée informatique et à la programmation avec Scratch. Consulté 1 septembre 2024, à l'adresse <https://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.html>
- ❖ Tchounikine, P. (2024). *Quels sont les enjeux de l'enseignement de l'informatique à l'école?* [Note d'expert scientifique, conférence de consensus du Cnesco « Nouveaux savoirs et nouvelles compétences des jeunes »]. https://www.cnesco.fr/wp-content/uploads/2025/03/Cnesco_CC-savoirs-competences_Notes-experts_P-Tchounikine.pdf
- ❖ Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. (version originale : <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf> ; traduction en français : <http://www.cs.cmu.edu/afs/cs/usr/wing/www/ct-french.pdf>)
- ❖ Projet Astrapi : [Astrapi](#) (Apprentissage et transfert analogique de la pensée informatique)

Bibliographie (autopromotion : Jolivet & al.)

- ❖ Jolivet, S., Kirouchenassamy, B., Yessad, A., & Luengo, V. (2025, accepté). Modélisation et décision de rétroactions épistémiques dans l'environnement AlgoPython. *Acte de la 10e conférence sur les Environnements Informatiques pour l'Apprentissage Humain, EIAH25*. 10e conférence sur les Environnements Informatiques pour l'Apprentissage Humain, EIAH25, Lille.
- ❖ Dechaux, E., Kirouchenassamy, B., Jolivet, S., Yessad, A., & Luengo, V. (2025, soumis). Typologie d'erreurs en programmation et détection automatique dans un code Python. *Actes de l'atelier APIMU 1000*. Atelier APIMU 1000, EIAH25.
- ❖ Muratet, M., & Jolivet, S. (2024). GTC un formalisme permettant l'analyse automatique des compétences de la pensée informatique dans un jeu sérieux : Application au jeu SPY. *STICEF*, 31(1), 1-27. <https://doi.org/10.23709/sticef.31.1.1>
- ❖ Jolivet, S., Wang, P., & Dechaux, E. (2024). Retour d'expérience de formation des enseignants d'informatique : Constats et enjeux didactiques. *Radix*, 1. https://imag.umontpellier.fr/~modeste/numero1/PREPRINT_RADIX_1_JOLIVET_AL.pdf
- ❖ Jolivet, S., Dechaux, E., Gobard, A.-C., & Wang, P. (2023). Construction et exploitation d'un référentiel de types de tâches d'apprentissage de la programmation. In S. Iksal, C. Pélissier, J.-M. Gilliot, & P. Marzin-Janvier (Éds.), *Actes du colloque EIAH2023 : 11ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain* (p. 166-177). https://eiah2023.sciencesconf.org/data/pages/actesEIAH2023v_3.pdf
- ❖ Jolivet, S., Dechaux, E., Gobard, A.-C., & Wang, P. (2023). Description et analyse de trois EIAH d'apprentissage de Python. *Actes de l'atelier APIMU : 11ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain*. Atelier APIMU : 11ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain, Brest.

Annexe pour faire suite aux échanges : pourquoi enseigner l'informatique ?

- ▶ Lors des échanges qui ont fait suite à la présentation a été abordée la question de la place et de la légitimité de l'enseignement de l'informatique comme discipline scolaire, en particulier dans le contexte de la réforme de la maturité à Genève.
- ▶ Mon point de vue :
 - ▶ Cette discussion dépasse le cadre de la didactique de l'informatique et la réflexion sur la place respective et la légitimité des différentes disciplines scolaires prend en compte des enjeux sociétaux, économiques, politiques... le résultat obtenu est souvent le fruit d'un compromis et de rapports de force
 - ▶ Il est raisonnable de veiller à ne pas ce que ces débats ne se transforment pas en affrontements entre enseignants de différentes disciplines (et il n'est pas forcément satisfaisant que ce qu'on peut estimer être des avancées se fassent au détriment des conditions de travail de collègues)
- ▶ Sur la diapo suivante le point de vue de Pierre Tchounikine (cf bibliographie)

Annexe pour faire suite aux échanges : pourquoi enseigner l'informatique ?

« L'enseignement de l'informatique, de façon générale et, notamment, à l'école élémentaire, est actuellement un objet de réflexion.

Il y a sur le sujet des discours très divers, plus ou moins fondés, et également plus ou moins liés aux enjeux sous-jacents (disciplinaires, médiatiques, politiques). Il y a beaucoup d'appels en faveur de cet enseignement, avec des arguments différents, étayés ou pas ; en substance : « cela permet de développer des compétences importantes », « cela développe la créativité des élèves », « c'est moderne », « c'est un domaine scientifique important », « c'est un secteur qui embauche », « mon petit frère adore », ou encore « c'est mon domaine d'activité, donc il est important, donc il faut l'enseigner ». Il faut donc prendre de la distance par rapport à un effet de mode et des discours parfois trop généraux et/ou trop prosélytes (ou caricaturalement négatifs), dont certains visent à faire agir le politique sur des questions de long terme plus qu'à aider les enseignants ici et maintenant. Cependant, prendre de la distance ne veut pas dire jeter le bébé avec l'eau du bain, i.e., refuser de considérer l'enseignement de l'informatique car certains discours sont trop généraux et/ou infondés (pour certains, complètement idiots), ou encore porteurs de confusions (par exemple qu'enseigner l'informatique a pour but de faire des élèves de futurs programmeurs : enseigner la lecture et l'écriture ce n'est pas former des écrivains, même si certains élèves deviendront peut-être écrivains). Il y a enfin des ressources pédagogiques mais qui, si l'on n'a pas les idées claires sur ce que l'on veut enseigner, pourquoi et comment, peuvent être difficiles à bien exploiter. » (Extrait de Tchounikine 2017)