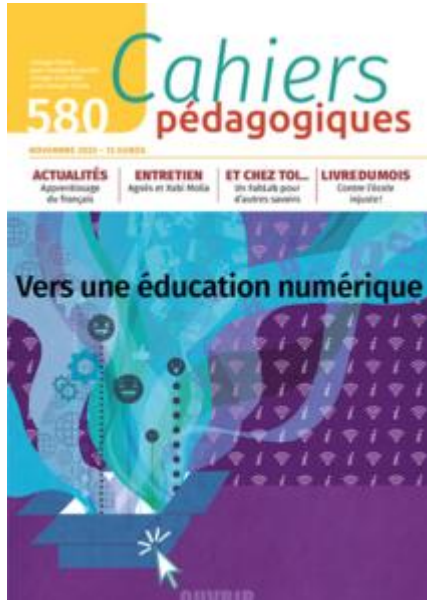


# Comment former des citoyens numériques débrouillards et critiques ?

Article publié le 28 octobre 2022 | Lecture : 8 min. | Par Alan Castellino et Carine Grancher



Quel regard portent les enseignants sur les compétences et usages numériques de leurs élèves ? Qu'est-ce que l'alphabétisation numérique pour la jeune génération ? Qui s'en charge ? Faut-il enseigner la programmation ? La discipline informatique doit-elle être autonome ? Dialogue entre Ping et Pong, deux enseignants à la recherche du curriculum idéal.

Printemps 2020, le confinement. Les vies numériques de la planète n'ont jamais été aussi riches. Les propositions de formation pour les professeurs viennent de tous les coins du monde, une caverne d'Ali Baba pour ceux qui savent les trouver. Nos deux auteurs, une femme (Pong), un homme (Ping), enseignant l'informatique, se rencontrent lors de l'une d'elles, animée par Julie Henry, dynamique chercheuse belge, spécialiste de la didactique de l'informatique. Le sujet est passionnant, des notes sont prises, échangées, et l'envie vient de continuer cette exploration ensemble.

**Pong** : Mon alphabétisation numérique ? Moi qui suis aujourd'hui ingénieure en informatique, formée dans les années 90 ? Une découverte solitaire dans une chambre d'enfant de 12 ans. Un des premiers ordinateurs personnels trône dans ma chambre. J'explore le DOS<sup>1</sup>, comme Champollion les hiéroglyphes. Mes explorations associent tout de suite usage de l'outil et kyrielle de concepts complexes à comprendre. Pas le choix.

**Ping** : Pour ma part, je me suis formé pour les métiers de l'enseignement il y a quelques années. J'ai eu un PC à la maison très tôt, avec mes parents, au début des années 2000, et une pratique en classe occasionnelle dès la maternelle, puis hebdomadaire en fin de primaire sur des projets. Pour l'appropriation, j'ai bénéficié d'apports de l'école, mais surtout de découvertes personnelles. C'était une époque où l'outil était déjà ancré dans la société, mais où le numérique n'était pas aussi ubiquitaire qu'actuellement dans une vie d'enfant ou d'adolescent. Enfin, MSN<sup>2</sup> était déjà un lieu de retrouvailles après l'école dès le CM1, et Facebook au collège...

**Pong** : J'enseigne l'informatique à des élèves du CM2 à la 2<sup>de</sup> depuis trois ans. Je vois comment s'écrivent les mots de l'informatique sur les pages presque blanches des plus jeunes.

**Ping** : J'exerce en mathématiques et informatique depuis quatre ans, actuellement dans les classes de lycée, à la fois en tronc commun et en spécialité. Mes élèves sont accros au smartphone. Plus qu'un moyen de communication, c'est un moyen d'expression identitaire. Les applis utilisées changent avec les dernières tendances, la consommation de médias se fait au gré du déroulement des algorithmes de recommandations. En revanche, dans le cadre des cours d'informatique, c'est l'ordinateur qui est utilisé. Dans ce contexte, sont-ils compétents et efficaces ? Pour la plupart, pas vraiment.

**Pong** : Au collège, ils découvrent les ordinateurs, les claviers, les souris, et pas seulement les plus jeunes d'entre eux. Je n'en crois pas mes stéréotypes sur la jeune génération ! Et encore maintenant, je m'en étonne, leur usage

des messageries, des fichiers, des routines de connexion à un compte leur pose encore beaucoup de problèmes. Pourtant, dans leur conversation enfantine ou adolescente, il n'y a presque que de l'information issue du monde numérique, des vidéos en ligne, de petites applis ou des jeux. Alors, c'est quoi l'alphabétisation numérique pour cette génération ? Qui s'en charge ? Avec quelle légitimité ? Avec quelles intentions ?

**Ping :** Je pense qu'il faut d'abord se demander quelles compétences sont nécessaires. Pour moi, en premier lieu, ce sont les possibilités de représentation symbolique et d'automatisation. L'enjeu n'est pas que tous les élèves soient des spécialistes de la programmation, mais qu'ils soient capables de contrôler la machine, d'utiliser et de concevoir des raisonnements algorithmiques pour résoudre certains problèmes, y compris de la vie courante, quitte à les mettre en œuvre un jour dans un langage de programmation simple d'accès. L'automatisation des tâches, la mise en algorithme peut répondre à cette problématique de démystification du numérique, en lien avec des situations concrètes pour aboutir à un usage plus cartésien, plus rationnel et une adaptabilité face à l'évolution technologique.

**Pong :** Comment faire de l'enfant, du jeune, un citoyen numérique débrouillard et critique, sachant déjouer les pièges du commerce de l'attention et de l'information falsifiée, virtuose de la communication, un expert de la recherche dans les librairies virtuelles, de la mise en forme de ses découvertes, de leur exposition au plus grand nombre ? Et ne laisser personne sur le bord de la route ? Ces « *oublié(e)s du numérique* », comme les appelle Isabelle Collet, faut-il venir les chercher tout de suite, très tôt, avec des outils pédagogiques adaptés, une didactique construite pour tous ? Et qui doit le faire ? Le système éducatif, c'est certain, mais les associations, les médias ou les entreprises à but lucratif se saisissent aussi de la question. Comment travailler de concert pour diffuser le bon message ? Et que s'agit-il d'enseigner ?

**Ping :** Il semble à priori naturel de s'interroger sur la dépendance entre les aspects objet et outil de la discipline. Si on pense qu'enseigner la science informatique peut se passer des compétences d'usage du numérique, alors le pédagogue se dirige vers des activités d'informatique débranchées (sans ordinateur) ou exploitant des ressources pensées pour cela, comme Scratch. Si on pense qu'à l'inverse l'un ne se fait pas sans l'autre, alors les activités de programmation nécessitent de savoir gérer les fichiers, d'utiliser des fonctions spécifiques des éditeurs de texte ou d'interagir avec la ligne de commande.

**Pong :** Oui, mais l'apprentissage de la programmation ne motive pas les élèves, 9 % d'entre eux choisissent la spécialité NSI (numérique et sciences informatiques) en classe de 1<sup>re</sup>, et même en Californie où cet apprentissage commence dès la primaire, seulement 3 % s'orientent au lycée dans le parcours informatique. Quand on interroge les élèves, ils invoquent la difficulté de la discipline. Du côté de la didactique de l'informatique, trop peu de recherches sont engagées, peu de résultats permettent de définir les obstacles rencontrés par les élèves et les moyens didactiques de les dépasser.

**Ping :** Tout d'abord, il est nécessaire d'opérer une distinction entre « connaissance des concepts » et « connaissance des produits »<sup>3</sup>. La première est généralisable, transférable, souvent abstraite, elle permet de mettre en relation des idées, est valable à long terme et nécessite de la compréhension et de la structuration dans le processus d'apprentissage. La connaissance liée aux produits (logiciels, langages, matériel particulier) est quant à elle nécessaire pour la mise en œuvre, mais ne jouit pas de la même généralité. Les compétences relevant de la manipulation doivent être sans cesse mises à jour, suivant l'évolution des interfaces et des fonctionnalités des programmes et n'ont donc que peu de légitimité. D'où l'idée de structurer l'enseignement autour de concepts fondamentaux ou méthodes pérennes, perceptibles, représentables, accessibles à divers niveaux et utilisées ou reconnaissables dans de nombreux domaines de l'informatique.

**Pong :** Il faudrait donc chercher les invariants entre les plateformes, les systèmes d'exploitation, les logiciels, les langages, et enseigner cela ? Mais alors, qu'y a-t-il en commun dans nos usages de l'informatique, de l'usager aux

programmeurs avertis ? Des réponses simples viennent à l'esprit : comprendre le concept de fichier et son système de sauvegarde, de nommage, la création des comptes, des droits et des mots de passe, l'écriture numérique, les recherches par mots-clés et l'exploitation de données. Il s'agirait alors pour les élèves d'apprendre à choisir l'application répondant le mieux à leurs besoins.

Derrière chacun de ces savoir-faire à acquérir se cachent les principes génériques de la « pensée informatique ». Comment l'enseigner ? Existe-t-il du matériel pédagogique pertinent ?

**Ping :** Plusieurs voies ont été explorées. Ce qui est intéressant, c'est que cette pensée constitue une compétence transversale en résolution de problème et ne s'applique pas uniquement à des situations informatiques. Une piste est de penser que faire de la programmation permet de construire des concepts, quand une autre pose l'hypothèse qu'avoir un projet et problématiser dans le cadre de ce projet rend nécessaire de programmer et donc de développer une pensée algorithmique. On peut penser que c'est une question de pédagogie qui interroge la place de l'enseignement explicite et celle du projet. Ces deux pistes s'opposent-elles, se complètent-elles ?

**Pong :** Aujourd'hui, dans mes classes d'informatique de collège et de lycée, j'ai fait le choix de m'appuyer sur ce que certains curriculums mondiaux proposent : faire tôt de la programmation dite créative avec Scratch ou de la robotique éducative ; choisir des thèmes de projet porteurs pour les populations moins enclines à choisir les options informatiques ; présenter les nombreux métiers autour de la conception d'outils numériques et donner des mentors aux élèves pour qu'ils se projettent dans des métiers numériques divers ; laisser des cours de programmation traditionnels pour les 10 % de futurs informaticiens programmeurs, mais envisager l'apprentissage d'un langage informatique comme celui d'une langue étrangère (ne pas commencer par la rédaction mais par la lecture, la compréhension, la reproduction de modèles puis l'extension<sup>4</sup>) ; travailler sur l'interdisciplinarité pour les autres élèves en s'associant à des projets issus d'autres disciplines.

**Ping :** Ici se pose le problème de l'intégrité disciplinaire : la discipline informatique doit-elle nécessairement être autonome ? S'il semble assez évident que l'algorithmique peut être mobilisée dans des disciplines scientifiques connexes, telles que les mathématiques, la physique, la biologie, du fait des modélisations possibles et donc des expériences rendues possibles dans un monde virtuel, le lien avec les disciplines littéraires et artistiques est moins évident. Pour les premières, le recours à la programmation permet de systématiser les traitements (gagner du temps, minimiser le risque d'erreurs dues à la perception humaine). Dans les secondes, l'emploi de méthodes numériques est une voie explorée depuis le XX<sup>e</sup> siècle. Entrer dans la programmation par la création artistique ne date pas d'hier. C'était déjà une idée défendue par Seymour Papert et le Logo<sup>5</sup>. Elle était également le principe fondateur de l'enseignement ICN (informatique et création numérique) proposé en 2<sup>de</sup> et 1<sup>re</sup> jusqu'en 2019. Si on assiste, ces dernières années, à une fusion entre les thématiques de formation au numérique et de formation par le numérique, avec le développement de nouvelles approches, la didactique de l'informatique est encore loin de fournir les réponses à toutes les énigmes que pose cet enseignement indispensable et pourtant si rejeté.

**Alan Castellino et Carine Grancher**

*Alan Castellino est enseignant au Cours Maintenon à Hyères (Var).*

*Carine Grancher est professeure de Sciences numériques et technologie en collège et lycée à Paris.*

## Notes

1. Système d'exploitation où les instructions sont données via des commandes (et non une interface graphique comme dans les systèmes actuels).
2. MSN Messenger, service de chat en ligne lancé en 1999, aujourd'hui disparu.
3. Concepts issus d'*Enseigner l'informatique*, de Werner Hartmann, Michael Näf et Raimond Reichert, Springer, 2011.
4. Nous faisons référence à l'outil PRIMM (Predict-Run, Investigate, Modify, Make) de Sue Sentance qui propose de nouvelles routines de didactique informatique.
5. Seymour Papert est un des pionniers de l'intelligence artificielle, ainsi que l'un des créateurs du langage de programmation Logo. Voir : [http://www-irem.univ-paris13.fr/site\\_spip/spip.php?article32](http://www-irem.univ-paris13.fr/site_spip/spip.php?article32).