Faut-il enseigner le code à l'école?

En juillet 2014, Benoît Hamon, ministre de l'Education nationale de l'époque, a annoncé dans une interview au Journal du dimanche : un apprentissage du « code » - informatique - dès l'école primaire, renforcé au secondaire.

Une telle déclaration, même si elle s'adresse au péri-scolaire a bien évidemment suscité un certain nombre de réactions. Beaucoup l'attendaient, certains le craignaient et comme souvent les extrêmes se sont opposés. Si de tels débats terminent régulièrement en discours de sourds, ils ont l'avantage de nous permettre de dégager les tendances générales pour nous permettre ensuite d'essayer d'y voir plus clair en cherchant une voie plus centrale ou tout autre.

Parmi les partisans de cet enseignement, nous retrouvons « codeorg » et sa vidéo qui a très rapidement fait le buzz sur internet.

Il est vrai qu'un teaser citant Steve Jobs, présentant Bill Gates,Marck Zuckerman et autres développeurs des plus grosses entreprises du numérique a le mérite d'avoir un impact fort. Mais suffit-il d'avoir des acteurs connus pour s'assurer de remplir les salles de cinéma?

Prenons le temps de décoder cette vidéo. Coder est simple et jubilatoire… Savoir coder, c'est être en mesure de maîtriser le monde demain. Il y est même dit que les codeurs sont les rock stars s'aujourd'hui… et pour cela, nul besoin d'être « très intelligent ».On est loin de l'image du geek boutonneux délaissés et laissé de côté des cours de lycée de notre époque. Et pour preuve la présence de ce champion de NBA qui avoue ne pas avoir été compris des ses amis mais qui aujourd'hui ne regrette pour rien au monde détenir un tel savoir et de telles compétences vient démonter tous nos stéréotypes. Comparés aux magiciens de demain, les codeurs sont en plus assurés de trouver un emploi et de travailler dans des conditions de rêve : entreprises aux locaux très design avec des lieux pour jouer, se détendre, se défouler, créer… Bref, le rêve! Quand on connaît le contexte actuel de l'emploi et qu'il est annoncé qu'un million de postes sont à pourvoir, la coupe est pleine, je signe. Je veux m'inscrire aux ateliers de code.

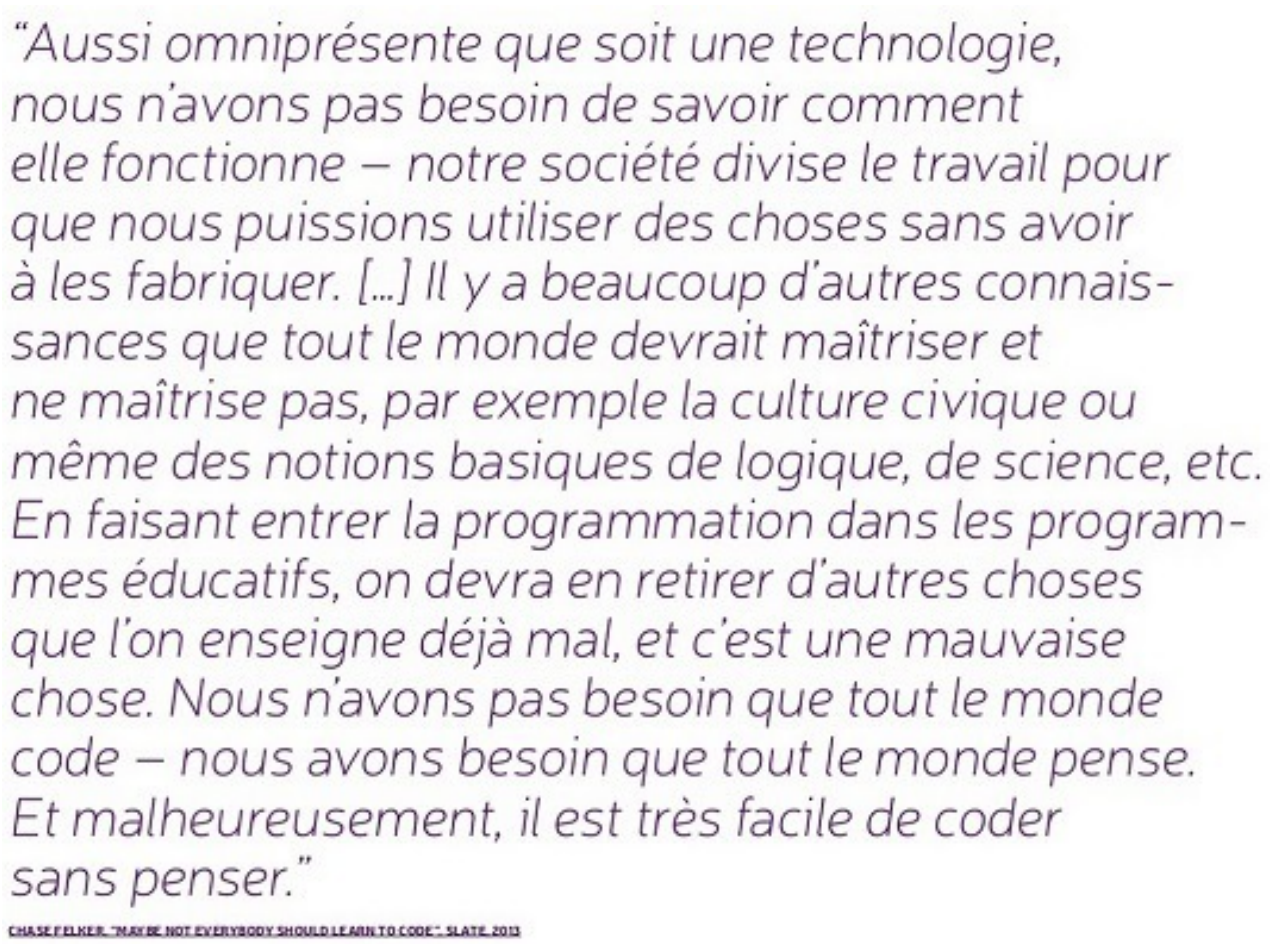
Difficile de résister à un tel discours médiatique, mais prenons tout de même le temps de le décoder. L'industrie du numérique peine à trouver des codeurs, et trop peu se forment pour combler les futurs besoins de ces grandes entreprises. 1 million de postes nous dit-on et quels chiffres annoncent-on juste en face? Seulement, 1 école sur 10 enseigne le code !

Le tour est joué, c'est la faute de l'école si l'avenir des géants du net est malmené… La solution ? Enseignons le code à l'école et ce dès le premier degré. Nul besoin de revoir leurs salaires, leur mise en valeur ou leur réelles conditions de travail. Je ne peux m'empêcher de faire un lien direct avec la définition que nous donne l'OMS de la santé mentale : *état de bien-être qui permet à chacun de réaliser son potentiel, de faire face aux difficultés normales de la*

vie, de travailler avec succès et de manière productive et d'être en mesure d'apporter une contribution à la communauté.

Pour être en bonne santé mentale, il faut travailler avec sucés et ce de manière productive. Etre humain n'est donc plus une fin en soi mais la notion de travail et de productivité semblent indissociable de la condition humaine. Quel message fait-on passer quand on sait qu'il n'y a pas assez de travail pour tout le monde?Ce n'est donc pas l'économie qui est au service de l'homme mais bel et bien l'inverse. Normal alors me direz-vous de demander à l'école de combler le vide et de s'adapter aux marchés.

Toutefois, Chase Falker nous dit :



A cel, nous pouvons ajouter, Benjamin Bayart, qui ose demander si au 19ème siècle lors de la révolution industrielle, il était demandé à l'école d'apprendre aux enfants comment fonctionnaient les machines à vapeur. A cela, nous pouvons ajouter les paroles de Verinis : « J*e peux utiliser une voiture ou prendre l'avion mais je n'ai pas besoin de savoir les construire. En quoi cela serait-il différent pour l'ordinateur?* »

A cette différence prêt, qu'un avion véhicule des gens alors que les ordinateurs véhiculent des informations et des idées et que le développement exponentiel des outils numériques donnant accès à ces informations nous induit de plus en plus dans des schémas de pensée, des habitudes et des logiques. Il n'y à qu'à faire un bref état des lieux entre notre manière de vivre aujourd'hui et 10 ans auparavant quand nos téléphones n'étaient pas tous nomades ni connectés au haut débit.

Alors faut-il suivre Douglas Rushkoff qui nous alerte avec ce message quasi publicitaire? « Programmez ou soyez programmé ». Derrière le code n'y aurait-il pas alors une question de liberté et de pouvoir? Le mouvement du libre et de l'open-source serait-il une solution à l'emprise quasi obsessionnelle de nos vies par les géants du net? A ce sujet, l'académie des sciences française milite pour l'enseignement du code à l'école afin de lutter contre les analphabètes numériques. Mais à ce titre là ne serions-nous pas tous (ou presque) des analphabètes? Aujourd'hui, qui connaît l'HTML, le javascript ou le php? Est-ce le code qu'il faut maîtriser ou les usages? Faut-il donc militer pour une éducation aux usages tel que travailler sur comment fonctionne une requête Google ou utiliser faire connaître et soutenir des outils tels que ceux développés par framasoft? ou bien remplacer google par duckduckgo par exemple?

Car si nous revenons à notre vidéo du départ, qui nous vante l'apprentissage du code c'est finalement pour aller travailler chez Facebook. On est alors très loin du « programmez ou soyez programmé ». Mais programmez pour les programmer.

Ne sommes-nous pas en train de tourner en rond? Tentons de sortir de ce débat d'idées pour regarder ce qui se passe du coté des apprentissages. Peut-être que nous ne regardons pas le code depuis le bon angle. En effet, le code est-il une fin en soi ou bien un prétexte? Tout comme l'est l'écriture d'un recueil de contes ou la création de fusées à eau avec ses élèves. Ces dispositifs sont prétextes et supports d'apprentissage pour permettre aux élèves de développer des compétences fondamentales. Dans un article du site internetactu, Rémi Sussan nous invite à dégager le code « de son association quasi-obligatoire » avec les machines. programmer, nous dit-il est « une affaire humaine, une forme de pensée. L'un des textes fondamentaux de l'enseignement de la programmation "Structure and Interpretation of Computer Programs" (qui fut, de 1984 à 2008, le manuel de base de l'enseignement informatique au MIT) ne s'y trompe pas lorsque ses auteurs proclament dès l'introduction : "Nous voulons établir l'idée que le langage informatique n'est pas seulement un moyen de réaliser des opérations, mais plutôt un nouveau médium formel permettant d'exprimer des idées concernant la méthodologie. Les programmes doivent d'abord être écrits pour être lus par les gens et seulement de manière secondaire pour être exécutés par des machines." En 2010, l'académie des sciences américaines a publié les résultats d'une discussion sur la pensée computationelle. Une forme de pensée qui ne serait ni des mathématiques ni de l'ingénierie. mais qui aurait de fortes similitudes avec elles deux. Et la pensée computationelle n'est pas dépendante des machines. Ainsi, Rémi Sussan, nous dit :

Dans le rapport de l'Académie des Sciences Joshua Danish de l'université de l'Indiana raconte ainsi son expérience sur une simulation, la collecte du miel par les abeilles : un algorithme propre aux systèmes "multi-agents" mais qui fut, ici, réalisé "dans le monde réel" par un petit groupe d'enfants, qui devaient apprendre à élaborer un modèle de l'abeille, et

mimer la recherche du miel et la communication (par la danse) aux autres membres de la ruche. Autre expérience mentionnée dans le rapport de l'académie, celle effectuée par Tim Bell, de l'université de Canterbury (.pdf), au cours de laquelle des enfants portant des tee-shirts colorés devaient, en tenant compte d'un certain nombre de contraintes, échanger des fruits distribués aléatoirement jusqu'à ce que chacun possède un fruit de la couleur de son tee-shirt. Ils reproduisaient ainsi dans le monde réel une opération algorithmique bien connue des professionnels du réseau sous le nom de "routage".

Bien sûr, tout cela est proche des mathématiques ou de l'ingénierie. Sussman explique que "*la pensée scientifique traite de pommes et d'oranges et se demande comment ces pommes et ces oranges peuvent être différentes ou semblables. Le sujet de la pensée mathématique, ce sont les sphères, avec les aires et volumes qu'elles possèdent (…). La pensée computationnelle cherche à comprendre comment un groupe peut couper et partager une pomme afin que chaque personne soit sûre qu'elle a obtenu une juste part de la pomme*."

Très proche des mathématiques des sciences, ce mode de pensée est proche de l'opérationel.

Et ce type de pensée peut même être utilisée par les poètes. Sussman cite à ce sujet un essai d'Edgar Allan Poe qui décrit le processus de composition poétique comme un algorithme. Nous sommes ici loin du code comme fin en soi, comme activité presque mécanique ne nécessitant pas d'être intelligent mais bien face à un mode de pensée. Celui-ci doit-il être enseigné à l'école, ces activités peuvent-elles être prétexte à développer une pensée logicomathématique, et doivent-elles être forcément reliées au monde des machines? Autant de question qui restent en suspend et qui mérite d'être traitées. Mais pour lesquelles, il nous faut certainement vivre des situations d'apprentissage de code pour ensuite les décoder et tenter de comprendre ce qui se passe chez l'apprenant, ce qui se joue en terme d'apprentissage pour ensuite décider si ces activités ont leur place à l'école au service du développement de la personne.