

Regooglisons Internet : le Libre citoyen

De grandes nouvelles et de beaux partenariats à vous annoncer en ce jour si particulier. Nous espérons vous renseigner en bonne intelligence, afin que vos données soient plus que respectées : afin qu'elles deviennent respectables.

Framasoft signe un pacte d'intelligence citoyenne

Suite à la dernière audition à l'Assemblée Nationale dans le cadre du projet de Loi Renseignement en vue de son examen prochain, Framasoft a eu le plaisir d'être convié à une soirée organisée conjointement par Red String et Clear Shield-France dans les bureaux de Google Paris dans le 9e. Nous avons eu la satisfaction de constater que le travail de toute la communauté en faveur des solutions libres a été reconnu, car nous avons été labellisés officiellement « Laboratoire des réseaux ouverts et libres » (Labrol).

Nous tenons à remercier tous ceux qui à nos cotés luttent et soutiennent notre action pour cette distinction qui nous honore. Voici ce que signifie concrètement cette labellisation : nous allons très bientôt signer un pacte d'intelligence pour opérer dans un esprit d'ouverture totale et néanmoins citoyenne, en permettant l'accès à nos serveurs à ces sociétés agréés par le Ministère de l'Intérieur. Une prochaine convention, se référant au texte de loi mentionné ci-dessus, sera paraphée sous l'autorité d'une commission administrative, afin de servir de banc d'essai (Luc préfère dire *proof of concept*) d'un Internet plus ouvert et plus sûr pour tous.



Keys on keyboard CC-BY-SA Intel Freepress

Libérer la Justice en la servant nous-mêmes.

Si nous avons fait le choix de capitaliser sur l'image de confiance renvoyée par Framasoft, c'est parce que nous croyons fermement que l'esprit d'ouverture et de bonne entente qui règne au sein des services offerts par Framasoft, permettra de concilier les impératifs de protection des citoyens et la possibilité de services respectueux des droits sans pour autant porter devant des services judiciaires déjà débordés les petites affaires de harcèlement et de suspicion pédopornographique que nous traitons régulièrement.

De même la surveillance des groupes Djihadistes, déjà identifiés pour la plupart sur Diaspora, sera d'autant plus facile que nous n'aurons plus à prendre la responsabilité de

communiquer les données dans la mesure où un système de réquisition automatique et de transfert sera assuré par Red String, la startup leader sur le traitement automatique des métadonnées.

Nous nous réjouissons de ce partenariat gagnant-gagnant qui nous permettra, grâce au prêt de deux serveurs de Red String, d'ouvrir plus tôt que prévu notre prochain Framadrop, un *DropBox-like* très attendu, basé sur une solution libre, et qui ravira les utilisateurs. Pour assurer à Framadrop le plus haut niveau de sécurisation contre les virus et autres fichiers malveillants, les fichiers seront automatiquement scannés par l'anti-virus que met gracieusement à notre disposition Clear Shield.

Qu'ils en soient ici remerciés.

Regoogliser selon l'exception culturelle Française.

Il ne s'agit donc pas d'un changement de cap, mais bel et bien d'une révolution. Nos services LEDS (Libres, Éthiques, Décentralisés et Solidaires) se verront désormais tous adjoindre un add-on Privatif, Unipersonnel, Technocratique et Espionnant ; afin de distinguer vos données de celles des personnes en voie d'auto-radicalisation ou pouvant porter trouble à l'ordre public. Avec cet algorithme à la pointe des technologies hexagonales, nous aurons donc le meilleur des philosophies étatsuniennes (analyse de masse des méta-données en collaboration avec l'exécutif et les services de renseignement) ; mais dans une exception toute française, qui permettra de réserver l'usage du Logiciel Libre uniquement aux citoyen-ne-s qui n'ont rien à se reprocher.

Le but à moyen terme est de vous fournir un espace de liberté totale et clairement délimitée dans un ensemble de services unifiés qui, avec votre soutien indéfectible, nous permettront

enfin de créer ce qu'un grand nombre d'entre vous nous demande à cor et à cri : le Frama-Phone. Nous avons d'ores et déjà acquis le soutien de l'association libriste La Triangulation du Net qui mettra à notre disposition sa base de données libre de géolocalisation, nécessaire à la transmission et au suivi des communications électroniques.

Bon, nous sommes réalistes, ce projet ne se fera pas en un jour, il vous faudra attendre plusieurs mois avant que sa commercialisation soit effective. Mais un premier prototype récemment expérimenté aux Journées du Logiciel Libre à Lyon a déjà atteint 720 points sur le *benchmark* du FAI Orange, le fameux « Système Automatisé de Légalisation des Opérations de Protection Électronique ». La route est longue mais nous la ferons avec vous en toute transparence.

Promis : On Ne Lache Rien.

Un FramaPhone...?



On n'arrive déjà pas
à faire des stickers...

Framasoft

La route est très très longue... ..mais la voie est très très Libre.

Maya, un frama-cat qui nous a vu boire du rhum.

Conférence : Entreprises du Libre... et communauté.

En novembre 2014, plusieurs membres de Framasoft sont venus en force au Capitole du Libre de Toulouse pour y répandre bonne humeur, chatons, et la défense d'un Libre accessible à tous (oui : même aux Dupuis Morizeau !)

La fine équipe de Toulibre, le GUL toulousain organisant avec brio le #CDL2014, a eu la bonne idée de capter ces conférences pour en faire profiter les internautes. Voici donc le deuxième d'une série de trois articles « Capitole du Libre ! »

Entreprises et communautés du Libre :

Quelles relations construire ?

Par Benjamin Jean.

Face à la question de la diffusion ou contribution à un logiciel libre, les licences cristallisent généralement toutes les attentions des entreprises, laissant croire que leur seul respect assure le succès de la démarche. Néanmoins bien d'autres engagements, pas uniquement juridiques, conditionnent le bénéfice de la confiance que doit apporter un acteur jusqu'alors inconnu : la communauté.

Que ce soit dans le cas de la diffusion d'un logiciel ou pour rejoindre un projet existant, les licences Libre et Open Source cristallisent généralement toutes les attentions des entreprises, laissant croire que leur seul respect assure le succès de la démarche. Nous verrons que cela n'est pas si évident et que bien d'autres engagements, pas uniquement juridiques, conditionnent le bénéfice de la confiance que doit apporter un acteur jusqu'alors inconnu : la communauté.

En effet et pour être réel, le bénéfice du logiciel Libre et de l'*Open Source* passe nécessairement par la constitution ou la rencontre avec une communauté. C'est cette communauté, protéiforme, qui concrétisera les promesses du Libre : communication, base d'utilisateurs, remontés de bugs à grande échelle, standardisation de la technologie, mutualisation de la maintenance – voire du développement de certaines contributions –, etc. Indomptable, elle se dilue beaucoup plus rapidement qu'elle ne se construit et demande un effort constant de la part des organisations qui souhaitent avoir son support. Sans cette communauté, il ne restera que les inconvénients et risques sans les avantages et opportunités.

Premier accord d'un dialogue qu'il faudra construire et maintenir, les licences libres et *Open Source* donnent le ton et sont à cet égard incontournables pour qui choisirait soit de diffuser son projet soit de contribuer à un projet préexistant. Néanmoins, elles ne suffisent pas à elles seules à assurer la confiance et le cadre nécessaires. Qu'ils soient professionnels ou acteurs individuels, les membres de cette communauté (qui ne sont pas uniquement des développeurs) seront fidèles et fiables à la seule condition que l'équilibre défini leur assure au moins autant de garanties que ne pourrait en avoir celui qui désire gagner cette confiance. Il ne peut y avoir que des gagnants, un objectif différent entraînera nécessairement le retrait de l'une ou l'autre des parties.

Avec un regard conscient sur les aspects juridiques sans pour

autant se limiter à ces derniers, cette intervention délimitera la substance et le rôle des communautés, avant de dresser un panorama des multiples relations existantes. Elle permettra de dégager *in fine* quelques bonnes pratiques et conseils afin que les entreprises disposent de toutes les cartes pour tirer pleinement profit des avantages du Libre. Au-delà des entreprises, tout acteur (individuel, acheteur, acteurs public, autre communauté) pourra être intéressé et questionné par ces réflexions.

Dégooglisons Internet : LA conférence !

En novembre 2014, plusieurs membres de Framasoft sont venus en force au Capitole du Libre de Toulouse pour y répandre bonne humeur, chatons et défense d'un Libre accessible à tou-te-s (oui : même aux Dupuis Morizeau !)

La fine équipe de Toulibre, le GUL toulousain organisant avec brio le #CDL2014, a eu la bonne idée de capter ces conférences pour en faire profiter les internautes. Voici donc le premier d'une série de trois articles « Capitole du Libre ! »

Dégooglisons Internet : des alternatives libres sont possibles.

Et si Google était le Skynet de 2024 ?

par Pyg.

Après « Et si on quittait Google ? », l'association Framasoft s'est lancé un nouveau défi : « Et si on dégooglisait internet ? ». Évidemment, il s'agit d'un projet ambitieux (qui a dit prétentieux ?;-)). Cependant, nous pensons réellement que le logiciel libre est la meilleure alternative au monde proposé (imposé ?) par Google.

Framasoft propose déjà plusieurs services alternatifs, Framapad (Google Docs like), Framacalc (Google Spreadsheet like), Framadate (Doodle like), etc. Ces services sont utilisables par tous, librement et gratuitement (Liberté 0), sans publicité, sans exploitation ou revente des données. Mieux, ils peuvent être auto-hébergés, afin de lutter contre la centralisation des données.

Mais ne pourrait-on pas aller plus loin ? Par exemple en proposant, pour chaque service privateur, une alternative libre ? Des services comme Google Groups, Google Forms, Google Search, Youtube, etc. mais libres et respectueux de vos données et de votre vie privée. Et pourquoi pas des alternatives à d'autres géants comme Skype, Dropbox ou même Facebook ? L'idée n'est évidemment pas de concurrencer ces services, mais de montrer que des alternatives existent, et peuvent être mise en place à moindre coût.

Nous ferons donc un bilan de ce qui a été accompli ces deux dernières années, et nous dévoilerons en détail notre Plan de Libération du Monde (rien que ça ! ☐).



Bonus : Accessibilité, le cas Framadate

Par Armony Altinier et JosephK.

Framadate est un projet de sondage libre porté par Framasoft, mais pas du tout accessible... Enfin, jusqu'à il y a peu.

Tour d'horizon technique des problèmes et solutions apportées.

Programme d'informatique dès

L'école primaire ?

La France a fait le choix depuis de nombreuses années de considérer l'informatique à l'école et jusqu'au collège, uniquement à travers ses usages via le B2I. L'Éducation nationale perçoit le numérique comme un outil utile aux autres apprentissages.

Cette vision n'est pas forcément mauvaise mais elle semble trop restrictive. Le numérique ne peut se limiter à son seul usage au service des autres disciplines. Il pourrait être pertinent de mettre en place un réel enseignement de l'informatique comme il en existe dans d'autres pays ou même en France (mais seulement en option au lycée).

Les programmes de l'école primaire étant en pleine ré-écriture actuellement, Serge Abiteboul, Jean-Pierre Archambault, Gérard Berry, Colin de la Higuera, Gilles Dowek et Maurice Nivat ont envoyé au Conseil supérieur des programmes ce texte, que nous reproduisons ci-dessous, présentant les grandes orientations de ce que pourrait être un programme d'informatique à l'école primaire.



Proposition d'orientations générales pour un programme d'informatique à l'école primaire

URL d'origine du document (EPI)

Ce texte propose des orientations générales permettant de structurer un futur programme d'informatique à l'école primaire.

Comme dans les autres disciplines fondamentales, la sensibilisation précoce aux grands concepts de la science et technique informatique est essentielle. Elle donne des clés aux élèves pour comprendre le monde qui les entoure, elle évite que se forment des idées fausses et représentations inadéquates, elle fabrique un socle sur lequel les connaissances futures pourront se construire au Collège et au Lycée. À l'École, il est important de montrer les liens qui unissent les concepts de l'informatique et ceux enseignés dans les autres disciplines, ainsi que ceux qui les unissent aux objets familiers que les élèves utilisent tous les jours. Sur ces deux points, nous pouvons nous appuyer sur des expériences longues et riches d'enseignements menées en France et hors de France.

Ces orientations s'inscrivent aussi dans une vision plus globale : après cette première sensibilisation à l'École primaire, vient le temps, au Collège, de l'acquisition de l'autonomie puis, au Lycée, celui de la maîtrise des concepts. Il est important de veiller à la progressivité et à la cohérence des programmes pour l'École, le Collège et le Lycée.

La question décisive de la formation des Professeurs n'est pas traitée dans ce document. Disons simplement qu'il nous semble essentiel que les Écoles supérieures du professorat et de l'éducation intègrent une formation, mais surtout une certification, en informatique.

L'école primaire doit être le temps de la découverte des concepts fondamentaux de l'informatique, celui où l'on parle aux élèves, avec leurs mots, à partir de leur quotidien et de leurs connaissances acquises dans les autres disciplines, d'informations, de langages de programmation, d'algorithmes et de machines. L'enseignement de l'informatique à l'École nous semble être trop souvent limité à l'utilisation d'ordinateurs et de logiciels créés par d'autres. Cette vision dénature une discipline scientifique et technique qui donne un rôle essentiel à l'abstraction et à l'expérimentation personnelle. Faire de l'informatique ne consiste pas à passer des heures devant un écran, mais à acquérir des notions fondamentales et universelles. L'initiation à l'informatique doit donc n'être liée ni à un ordinateur particulier, ni à un logiciel particulier, ni à un langage particulier. Elle doit par ailleurs chercher un équilibre entre des activités fondées sur l'utilisation d'un ordinateur et des activités « débranchées », c'est-à-dire ne recourant pas à une telle utilisation.

Des logiciels aux concepts

Les activités s'effectuant avec un ordinateur débutent avec l'apprentissage des logiciels les plus courants : logiciel de courrier électronique, navigateur, moteur de recherche, logiciel de traitement de texte, tableur, etc. Cet apprentissage ne doit pas rester une fin en soi, mais aussi conduire à s'interroger sur le fonctionnement de ces objets, menant ainsi à découvrir certains concepts de l'informatique.

Beaucoup d'élèves, par exemple, savent envoyer un courrier électronique, mais ils ne cherchent pas toujours à savoir comment un tel message arrive dans la boîte aux lettres de son destinataire. Pourtant l'apprentissage de l'utilisation d'un logiciel de courrier électronique est une occasion de les amener à se poser cette question et à y chercher des réponses. Cette interrogation, cette énigme, peut être l'occasion d'une

recherche collective, chaque élève proposant une hypothèse et critiquant celles des autres. Elle peut donner lieu à une contextualisation historique : l'acheminement d'un courrier électronique n'est peut-être pas si différente de l'acheminement d'un courrier postal, qui fonctionnait déjà dans l'Antiquité quand les ordres militaires irriguaient de vastes empires. Cette question permet donc d'introduire la notion de réseau – les ordinateurs sont reliés entre eux par des câbles ou par voie aérienne – et de routage – un message doit trouver son chemin dans le labyrinthe que constituent ces milliards d'ordinateurs reliés entre eux.

Le Web et ses logiciels – navigateurs, moteurs de recherche, etc. – permet de poursuivre la réflexion sur ces questions, mais aussi d'en poser de nouvelles. L'acheminement du contenu d'une page d'un serveur Web jusqu'à l'ordinateur de l'école suit les mêmes principes que l'acheminement d'un courrier électronique. À un certain niveau d'abstraction rien ne distingue un courrier du contenu d'une page web, et les méthodes permettant d'acheminer l'un permettent également d'acheminer l'autre. Mais de nouvelles questions apparaissent : quelle est l'origine des informations auxquelles on accède ? Qui écrit ? Où les informations sont-elles enregistrées ? Comment sont-elles identifiées ? Les élèves peuvent même concevoir leur propre page web et devenir ainsi des participants actifs du Web ; c'est le meilleur moyen de comprendre que n'importe qui peut dire n'importe quoi dans une page web, et de s'interroger sur la pertinence de l'information que l'on y trouve. Le Web est aussi l'occasion d'aborder la question de la recherche des pages contenant certains mots-clés, fondée sur la notion d'indexation : les moteurs de recherche recherchent d'abord les documents qui contiennent des occurrences des mots significatifs de la question qu'on leur pose. L'indexation conduit à une réflexion sur le sens d'un texte, puisqu'elle vise à identifier ce dont le texte parle, même si cette réflexion est souvent réduite à l'identification de mots clés. Or, la compréhension et

l'extraction du sens d'un texte sont parmi les buts fondamentaux de l'enseignement de la langue ; il faut désormais y adjoindre une initiation à l'indexation et une réflexion critique sur la pertinence d'un texte vis-à-vis d'une question posée.

L'initiation à l'informatique doit aussi passer par la découverte des concepts fondamentaux de langage, d'information, d'algorithme et de machine, sans toujours utiliser un ordinateur pour cela.

Des langages simples

Un langage formel se distingue d'une langue naturelle par sa spécialisation, son caractère artificiel, le caractère limité de son lexique et la simplicité des règles qui régissent sa grammaire. Un exemple simple est le langage formé de quatre mots : « nord », « sud », « est » et « ouest » et d'une construction, la séquence, qui permet de former des suites de tels mots. Ce langage permet d'indiquer un chemin à suivre sur une grille carrée, par exemple sur le carrelage du préau d'une école. L'expression « nord, nord, nord, est, est, est, sud, sud, sud, ouest, ouest, ouest » indique ainsi à un élève de se déplacer de trois carreaux vers le nord, puis de trois carreaux vers l'est, puis de trois carreaux vers le sud et enfin de trois carreaux vers l'ouest, parcourant ainsi un carré sur le sol.

Ce même mouvement peut être exprimé dans un autre langage qui ne comprend que trois mots : « avancer », « tourner à droite » et « tourner à gauche », composés par l'opération de séquence : « avancer, avancer, avancer, tourner à droite, avancer, avancer, avancer, tourner à droite, avancer, avancer, avancer, tourner à droite, avancer, avancer, avancer, tourner à droite ».

L'apprentissage de ces deux langages permet de mettre en place de nombreuses activités et de poser de nombreux problèmes. Une

première activité consiste à interpréter les instructions données par un autre élève ou à trouver la phrase qui commande d'aller d'un point du préau à un autre. C'est le jeu du « robot idiot ». Ce problème est le même que celui de la conception d'un programme dans un langage tel que Logo (il y a de nombreux enseignements à tirer de l'utilisation de Logo à l'École primaire), mais où un élève joue le rôle de l'avatar informatique afin de mieux s'approprier le lien qui s'établit entre une expression du langage et une action. On peut ensuite passer à des exercices plus élaborés, comme la traduction d'une expression d'un langage dans un autre – par exemple une expression formée dans le premier des langages présentés ci avant dans le second –, la mise en évidence de la redondance d'un langage – par exemple, un « tourner à gauche » pourrait être remplacé par une séquence de trois « tourner à droite ». Il est aussi possible d'évoquer dans une telle activité la notion de bug : une petite erreur dans une instruction exprimée dans le second langage, par exemple un « tourner à droite » de trop, change complètement la trajectoire et envoie l'élève n'importe où.

Ce type d'activité permet aussi d'aider les élèves à comprendre, dans un cadre très simplifié, quelques-uns des traits essentiels de la langue écrite : son caractère conventionnel, la nécessité de règles et la correspondance entre les mots et les actions. Elle leur permet aussi de comprendre qu'il est possible de calculer, non avec des nombres, mais avec des mots.

Des langages moins simples

Ces activités débranchées peuvent mener à des activités sur machine, par exemple à des activités de programmation dans un langage tel que Scratch, développé au Massachusetts Institute of Technology spécialement pour enseigner la programmation à l'École. Il permet d'assembler visuellement des instructions et de créer des tests et des boucles, afin d'animer de petits

personnages. Ce langage est emblématique de cette démarche ludique où les élèves créent des objets informatiques, d'abord par un simple dessin, puis les animent et augmentent leurs savoir-faire au fur et à mesure de leur découverte personnelle des possibilités offertes. De telles activités de programmation – ou de codage – sont aujourd'hui fréquemment proposées aux élèves hors de l'École. Les mettre en œuvre pour tous les élèves permettra à toutes et tous de profiter de ce type d'apprentissages essentiels dans le monde dans lequel nous vivons.

D'autres activités autour de la notion de langage sont liées à la programmation de robots physiques animés par des algorithmes. Les clubs de robotique développent ce type d'activités et leur efficacité auprès des enfants est un fait avéré.

Des activités plus difficiles peuvent être proposées à la fin de l'école primaire : des exercices visant au rangement, à la classification de données, à l'analyse de multiples situations combinatoires simples, à la recherche d'objets ayant certaines propriétés dans un ensemble fini d'objets. Les expériences menées hors de France et en France, à l'École et hors de l'École, montrent qu'il est possible d'aller assez loin dans cette direction, même avec de jeunes enfants.

La notion d'information

La notion d'information est aussi une formidable clé pour entrer dans l'informatique. La première notion à transmettre est celle de représentation : toute information peut être représentée par une suite de lettres dans un alphabet fini, par exemple par une suite de 0 et de 1. Les images, les sons, les textes, les nombres ont tous une représentation en machine, qui permet de les mémoriser, de les transmettre, de les transformer et de les reproduire à l'infini. Il est possible dès l'école primaire d'introduire l'atome

d'information, le bit, et de se demander combien de bits sont nécessaires pour représenter une information. Pour exprimer si la lumière est allumée ou éteinte, un bit suffit, alors que pour décrire la couleur des cheveux d'une personne – bruns, châains, blonds ou roux – deux bits sont nécessaires. Pour décrire une couleur parmi les 16 777 216 du système RVB, vingt-quatre bits suffisent. Ici apparaît la notion de quantité d'information contenue dans un message, qui est, en première approximation, sa taille.

Ces notions peuvent être introduites par des jeux. On peut, par exemple, proposer un langage pour coder un petit dessin en noir et blanc : il faut pour cela décomposer le dessin en pixels, puis coder chaque pixel, qui est ou bien noir ou bien blanc, par un bit. Par exemple, en supposant que 1 code pour un pixel noir et 0 pour un pixel blanc, la suite de vingt bits 11111000111100011111 représente un dessin bi-dimensionnel :

```
1 1 1 1
1 0 0 0
1 1 1 1
0 0 0 1
1 1 1 1
```

ou encore



Deux élèves ou groupes d'élèves, de part et d'autre d'un paravent, peuvent ainsi s'échanger, par oral, des dessins sur une grille de vingt-cinq ou cent pixels. Il est cependant probable que des erreurs apparaissent lors de la transmission, ce qui sera l'occasion de s'interroger sur la manière de corriger ces erreurs, par exemple en introduisant une forme de redondance dans le message. Il est aussi possible d'envoyer les bits à l'envers, du dernier au premier, et de s'interroger

sur l'effet de cette transformation sur le dessin.

La notion d'algorithme

Un algorithme est une manière de résoudre un problème en effectuant des opérations élémentaires mécaniquement et donc sans réfléchir. Tout algorithme doit s'exprimer d'une part dans une langue naturelle, ce qui est nécessaire pour sa compréhension, et d'autre part dans un langage technique précis, ce qui est indispensable pour faire en sorte que la machine puisse l'effectuer automatiquement. Les algorithmes qui transforment des symboles écrits – addition, soustraction, déclinaisons, conjugaisons, etc. – sont aussi anciens que l'écriture. Mais l'humanité a bien entendu utilisé des algorithmes avant même la naissance de l'écriture, pour tisser des étoffes, tailler des silex, etc.

Comme les notions de langage et d'information, la notion d'algorithme peut s'aborder par des activités « débranchées ». L'initiation peut commencer par l'identification d'algorithmes simples que les élèves utilisent tous les jours : pour s'habiller il faut mettre son tee-shirt avant son pull ; pour faire une tarte aux pommes, il faut mettre les pommes avant la cuisson de la pâte, mais pour une tarte aux fraises, il faut mettre les fraises après la cuisson.

Une deuxième étape est une interrogation sur les constructions qui permettent d'exprimer un algorithme :

- la séquence : faire ceci puis cela ;
- le test : si telle condition est vérifiée, alors faire ceci, sinon faire cela ;
- la boucle : faire ceci trois fois, ou alors répétitivement jusqu'à ce que telle condition soit vérifiée.

La notion d'algorithme est une formidable opportunité de relier l'informatique aux autres disciplines enseignées à

l'École telles le français, les mathématiques ou les travaux manuels, car beaucoup des connaissances enseignées dans ces disciplines se formulent sous la forme d'algorithmes. Par exemple, l'algorithme de l'addition de deux nombres de trois chiffres décimaux peut être décomposé en une boucle à l'intérieur de laquelle se trouvent des instructions élémentaires en séquence : la lecture d'un chiffre de chacun des nombres et de la retenue, la consultation d'une table qui permet d'ajouter trois chiffres, l'écriture d'un chiffre du résultat et celle de la retenue. De nombreux exercices de mathématiques proposés en cycle 2 nécessitent la mise en œuvre d'un algorithme, souvent formé d'une unique boucle, mais où les notions de donnée, d'instruction, de test, de terminaison apparaissent clairement.

Avant même l'apprentissage de l'algorithme de l'addition, l'apprentissage de l'art de compter des objets recèle une possibilité de poser plusieurs questions d'informatique. Quoique l'on compte, il faut commencer par choisir arbitrairement un élément auquel on attribue le nombre 1, puis choisir un deuxième élément, distinct du premier, lui attribuer le nombre 2, etc. Des questions essentielles apparaissent : comment se saisir d'un élément, lui attribuer un nombre et le marquer afin de ne pas le compter deux fois, tout en n'oubliant aucun élément dans l'énumération. Cette question du marquage est résolue de façons différentes selon les objets comptés : s'il s'agit des billes contenues dans un sac, on se munit d'un deuxième sac dans lequel on transfère une à une les billes déjà comptées. Si on compte des croix dessinées sur un cahier, on les entoure d'un cercle. Mais quel algorithme utilise-t-on pour compter les tuiles sur un toit ? Ou les arbres dans une forêt ?

De même l'accord d'un participe passé peut se décomposer en deux tests imbriqués, l'un sur le verbe auxiliaire utilisé, l'autre sur la présence d'un complément d'objet situé avant le verbe. Ces algorithmes s'apprennent en mathématiques, en

français, etc., par imitation d'exemples de difficulté graduelle. En informatique, il est possible de porter un nouveau regard plus systématique sur certains de ces algorithmes et de montrer comment ils se construisent à partir des constructions de séquence, test et boucle. Il est aussi possible de commencer à sensibiliser les élèves au fait que ce sont les mêmes constructions qui permettent de construire les algorithmes appris en français, en mathématiques ou en travaux manuels, anticipant ainsi la notion d'universalité, qui sera développée dans la suite du cursus des élèves.

La notion de machine

La notion de machine peut paradoxalement, elle aussi, être abordée par des activités débranchées. Il est par exemple possible de montrer aux élèves combien il est difficile pour eux de se comporter comme des robots, et de les amener à s'interroger sur l'origine de cette difficulté.

Le jeu du « robot idiot » peut être réutilisé ici, non pour réfléchir à la notion de langage, mais à celle d'architecture des machines. On peut fabriquer de petites cartes avec les mots « avancer », « tourner à gauche », « tourner à droite » et donner au robot humain un paquet de cartes qui est son « programme ». L'élève n'a désormais plus le droit d'écouter ce que lui disent ses camarades, mais doit uniquement lire une carte, exécuter une action et passer à la carte suivante. Pour d'autres programmes, tels le programme de l'addition, on pourra lui adjoindre des boîtes en carton – des variables – dans lesquelles il pourra stocker une valeur. Il illustrera alors le fonctionnement d'un processeur qui lit une instruction, exécute une action et passe à l'instruction suivante.

Il est aussi utile d'attirer l'attention des élèves sur quelques éléments clés de l'histoire des machines. On pourra leur raconter comment les Péruviens calculaient avec des

ficelles couvertes de nœuds – les kipu –, et comment un boulier sert de béquille à notre mémoire : on y « pose » les chiffres des nombres à additionner en déplaçant les boules, tandis que l'accumulation des boules permet d'additionner. On peut montrer la conception au XIXe siècle, de machines munies d'un moteur qui ne nécessitaient pas d'intervention humaine pour effectuer un calcul. Enfin au XXe siècle, illustrer la révolution conceptuelle des premiers ordinateurs, machines capables d'exécuter non plus un calcul particulier mais n'importe quel calcul, ce qui demande un langage pour les programmer.

Évoquer les machines du passé, du boulier au minitel, permet de s'interroger sur ce qui est invariant dans l'histoire des machines à traiter de l'information comme les concepts de représentation, d'algorithme, de langage, la généralité de l'ordinateur, etc. et, au contraire, de ce qui évolue constamment : la vitesse de calcul et la capacité des machines, leur taille, leurs domaines d'application, leur liaison avec le monde physique etc. Cela permet aussi de développer un sens de l'histoire : comment vivions-nous, comment communiquions-nous, comme cherchions-nous de l'information avant l'informatique ?

Un enseignement adapté à l'élève et à son rapport au monde

Ce qui est ébauché dans ce document répond à des objectifs spécifiques aux élèves de l'École primaire. Nous avons cherché à proposer des activités ludiques mais instructives au sens profond du terme, et adaptées aux connaissances et aux capacités des élèves. Mais l'essentiel n'est pas là. Il est d'abord dans le fait que l'informatique est une science et une technique faite d'abstractions, et que ces abstractions ne sont pas accessibles aux élèves directement. Elles doivent être appréhendées à travers des situations concrètes et de

l'expérimentation personnelle. C'est le levier des activités débranchées, qui propose aux enfants de manipuler des concepts à travers des gestes de leur propre corps et la manipulation d'objets familiers. Ensuite, à partir de sept ans, les enfants entrent dans l'âge des « pourquoi », mais à un niveau métaphorique. On doit donc leur proposer des explications, certes provisoires, des concepts informatiques, mais en mettant en place un vocabulaire précis qui leur permettra en grandissant d'affiner et d'enrichir les concepts associés aux mots.

Les activités débranchées sont complémentaires des activités avec un ordinateur, notamment l'apprentissage des logiciels les plus courants et l'utilisation des logiciels pédagogiques disponibles sur les ordinateurs et les tablettes, par exemple ceux d'apprentissage de la lecture, qui conduisent aussi à se poser des questions d'algorithmique humaine et mécanique.

On dira « langage », « information », « algorithme », « machine » et les enfants grandiront en se souvenant que l'information est aussi une quantité qui se mesure, qu'un langage peut être une forme de codage très rudimentaire, qu'une méthode devient un algorithme quand on a éliminé tous les implicites de la langue ordinaire et qu'une machine n'est qu'un outil qui permet d'exécuter des algorithmes.

Le 6 décembre 2013

Serge Abiteboul, Professeur au Collège de France (2012), membre de l'Académie des sciences, membre du Conseil National du Numérique.

Jean-Pierre Archambault, Président de l'association Enseignement Public et Informatique (EPI).

Gérard Berry, Professeur au Collège de France, membre de l'Académie des sciences et de l'Académie des technologies.

Colin de la Higuera, Président de la Société Informatique de

France (SIF).

Gilles Dowek, Directeur de recherche à l'INRIA, Grand Prix de philosophie de l'Académie Française.

Maurice Nivat, membre de l'Académie des sciences.

Ce document a été envoyé au Conseil Supérieur des Programmes (CSP), le samedi 7 décembre 2013.

Crédit photo : Lupuca (Creative Commons By-Sa)

Le logiciel libre a fait de moi l'homme que je suis

Le témoignage simple et percutant d'un développeur qui a visiblement fait le bon choix ☐



L'open source a fait de moi l'homme que je suis

Open Source made me the man I am

Matteo Spinelli – 6 octobre 2013 – Cubiq.org

(Traduction : Asta, Joseph, GregR, aKa, Cyb, Spanti Nicola, Lydie + anonymes)

De la conception de sites web pour les entreprises nationales au développement d'applications web haut de gamme pour les plus grands acteurs internationaux, tout cela grâce aux logiciels libres.

J'ai longtemps été un (triste) programmeur PHP en freelance ayant quelques compétences en front-end. Je travaillais pour de petites boîtes locales. Mon job le plus sympa en ce temps-là a été avec un distributeur de jeux vidéo, chez moi, en Italie. Le client était sympa, mais le boulot était chiant, et si frustrant parfois.

Je savais que je pouvais donner plus, mais je me sentais pris dans des sables mouvants.

La décision la plus importante que j'ai prise dans ma carrière a été de commencer à développer un logiciel libre et de bloguer sur ce sujet. J'ai commencé avec des trucs un peu stupides, comme un générateur d'URL épuré en PHP ou la suppression du délai sur l'évènement onClick et j'ai fini avec iScroll et l'ajout de widgets à l'écran d'accueil.

J'ai choisi pour eux la licence la plus libre que j'ai pu trouver (MIT) et les entreprises à travers le monde m'ont contacté pour me demander de la personnalisation et des nouvelles fonctionnalités. Mon tarif horaire était autour de 60 \$ et j'ai dû l'augmenter sur une base à la journée parce que je ne pouvais pas suivre avec l'augmentation des demandes. Maintenant je suis toujours travailleur indépendant mais je

travaille pour **Microsoft** et **Google** et mon tarif horaire est de 150 \$.

L'*open source* a augmenté ma visibilité mais ce n'est pas qu'une question d'*audience*. L'*open source* fait généralement de vous un meilleur développeur. Cela vous force à vous comparer vous-même avec d'autres développeurs et c'est le meilleur entraînement pour votre cerveau de codeur.

J'ai plus appris sur le JavaScript des gens postant des suggestions sur le rapporteur de bogues que sur n'importe quel guide, tutoriel ou livre que j'aie jamais lu.

Les logiciels libres m'ont fait également devenir un développeur plus modeste. Je sais comment patcher de petites portions de code et je suis moins sévère quand je remonte des bogues sur les dépôts des autres.

Mais c'est juste une partie de l'histoire.

Vous ne faites pas des logiciels libres juste pour la gloire (et l'argent). Peut-être qu'au début c'était mon intention, mais une fois que vous êtes impliqué vous comprenez que vous faites bien plus.

Beaucoup de gens utilisent votre code, vous aidez les startups dans leur projet en créant potentiellement de nouveaux emplois. Avec peut-être 48h de votre vie, vous pouvez possiblement aider des dizaines d'entreprises et leurs employés. Une personne a fait un plugin pour WordPress qui était essentiellement une couverture PHP pour mon *Ajouter à l'écran d'accueil* et il a levé 50k\$ de fonds (peut-être plus maintenant). Vous pouvez penser que je suis jaloux de lui, mais je suis en fait heureux pour lui (et tous ses utilisateurs).

En outre, plus je développe du logiciel libre plus j'apprécie les autres logiciels libres et j'en deviens accro. Je comprends ce que signifie coder pour la **sécurité** et, plus

particulièrement, l'importance de la **vie privée** de l'utilisateur (et la mienne).

J'étais un fervent utilisateur d'Apple parce que c'est joli et bien rangé et cela fonctionne simplement, mais peut-être qu'il y a des choses plus importantes qu'une interface de qualité et un dégradé parfait de pixels. J'utilise maintenant les produits Apple seulement pour tester et ma plateforme principale est Linux.

Je peux sans hésiter dire que l'open source a fait de moi un homme meilleur et je vous encourage à publier votre code sous une licence libre, parce que si ça a marché avec moi, ça marchera probablement avec vous aussi.

Crédit photo : Beshef (Creative Commons By)

Éducation : priorité au logiciel libre ou respect de la « neutralité technologique » ?

On commence à connaître la chanson et elle est emblématique de l'époque actuelle : le *privé* qui s'alarme et demande au *public* de le soutenir sur le dos des *biens communs*.

Ici nous sommes dans le secteur sensible de l'éducation et pour refuser la priorité aux logiciels libres on est prêt à tout, comme sortir du chapeau la notion pour le moins vague et floue de « neutralité technologique » (sans oublier le FUD sur l'innovation, la croissance, la destruction d'emplois,

toussa...)

Le Sénat a en effet examiné cette semaine le projet de loi sur la refondation de l'école de la République. Parmi les dispositions introduites par la commission de la culture, de l'éducation et de la communication, on trouve une modification apportée à l'article 101 **qui donne la priorité au logiciel libre et aux formats ouverts dans le futur service public du numérique éducatif.**

Pour ceux (comme l'April ou l'Aful) qui se battent depuis des années, voire des décennies, pour qu'il en soit ainsi, c'est déjà un grand pas en avant.

Un pas en avant clairement non partagé par le Syntec Numérique et l'AFINEF (Association Française des Industriels du Numérique dans l'Education et la Formation) qui ont publié dans la foulée un communiqué de presse que nous avons reproduit ci-dessous.

Dans la mesure où les communiqués de l'April, de l'Aful et du Syntec ne proposent pas d'espace de discussion sous article, **nous invitons toutes celles et ceux que le sujet intéresse à intervenir dans les commentaires.**

PS : Rappelons à l'occasion l'article de Richard Stallman : Pourquoi les écoles devraient utiliser le logiciel libre et l'enseigner.



Numérique éducatif et pédagogique : les professionnels du numérique interpellent le Gouvernement sur le respect de la neutralité technologique

CP du Syntec Numérique et l'AFINEF (22 mai 2013)

URL d'origine du document

Syntec Numérique et l'AFINEF interpellent le Gouvernement sur la mention favorisant de manière prioritaire l'utilisation de logiciels libres pour le service public du numérique éducatif, dans le projet de loi en débat ce mercredi 22 mai au Sénat sur la Refondation de l'Ecole de la République. Syntec Numérique, le syndicat professionnel des industries et métiers du numérique, et l'Association Française des Industriels de l'Education et de la Formation (AFINEF) interpellent le Gouvernement sur les dispositions modifiées du Projet de Loi de Refondation de l'Ecole de la République issues du travail

en commission au Sénat, **donnant la priorité à l'utilisation de logiciels libres pour le service public du numérique éducatif.**

Malgré un avis défavorable du Gouvernement en commission, la rédaction retenue à l'alinéa 7 de l'article 10, « *Ce service public utilise en priorité des logiciels libres et des formats ouverts de documents* », **porte atteinte au principe de neutralité technologique** qui est la règle notamment pour la commande et l'achat publics.

Par ailleurs, le rapport annexé à la Loi à l'article 1er, définissant les moyens et les orientations de la refondation de l'école de la République mentionne que « *L'incitation au développement de ressources numériques se fera notamment en faveur de contenus et de services numériques dits « libres »* ».

Les professionnels du numérique regrettent que **ces deux dispositions contredisent la circulaire du premier ministre du 19 septembre 2012** sur les modalités de l'utilisation des logiciels libres dans l'administration tout **en portant atteinte au pluralisme** des ressources informatiques.

En effet, au moment où le Gouvernement engage une politique d'aide à la création d'une filière d'acteurs français du numérique éducatif et pédagogique, ces dispositions, si elles sont définitivement adoptées, handicaperont gravement les efforts de développement de la plupart des entreprises déjà présentes sur cette filière qui n'a pas encore trouvé sa consolidation et dissuaderont de nouveaux acteurs innovants de s'y engager.

Pour les acteurs du numérique, **le pluralisme et la neutralité technologique, adossés à l'interopérabilité, sont les conditions sine qua none** afin de s'engager dans le développement des produits et supports technologiques pour répondre à l'enjeu essentiel de la modernisation du service public éducatif et à l'accompagnement des enseignants dans

leur investissement pédagogique.

Ils souhaitent par ailleurs alerter les pouvoirs publics sur **les conséquences que ces dispositions auraient sur la lisibilité de l'offre e-éducative française à l'international**, notamment en direction de continents où le pluralisme et la neutralité technologique conditionnent la pénétration des marchés.

Ils soulignent enfin que ces dispositions évidemment inconstitutionnelles, augmenteront **le risque de recours contentieux** entre les opérateurs privés du secteur et les administrations. En effet, ces dispositions rentrent en contradiction avec les principes d'égalité de traitement et de liberté d'accès à la commande publique, rappelés à l'article 1er du Code des marchés publics et qui ont acquis valeur constitutionnelle (Cons. Const. 26 juin 2003) : *« Les marchés publics et les accords-cadres soumis au présent code respectent les principes de liberté d'accès à la commande publique, d'égalité de traitement des candidats et de transparence des procédures. Ces principes permettent d'assurer l'efficacité de la commande publique et la bonne utilisation des deniers publics. Ces obligations sont mises en œuvre conformément aux règles fixées par le présent code »*.

Force est de constater que ces dispositions vont en effet à l'encontre du principe de neutralité du droit des marchés publics, qui ne permet aucunement de favoriser des opérateurs économiques au détriment d'autres. Au contraire, les règles de la commande publique ont pour objet de permettre au pouvoir adjudicateur de sélectionner l'offre économiquement la plus avantageuse, correspondant à ses besoins, après une mise en concurrence. Ainsi, écarter la fourniture de certaines solutions, en imposant un type de logiciels, violerait les principes fondamentaux de la commande publique consacrés tant au niveau français qu'europpéen et jamais démentis.

Syntec Numérique et l'AFINEV en appellent ainsi à la sagesse

de la Haute Assemblée et au respect de la feuille de route numérique du Gouvernement, afin de soutenir une filière qui a besoin d'être stimulée par un environnement législatif et juridique stable, et non par des mesures discriminatoires infondées, pour donner la mesure de tout son potentiel, en termes d'innovation, de croissance et de création d'emplois.

Crédit photo : Sashomasho (Creative Commons By-Sa)

Former la prochaine génération de bidouilleurs libres

Comment des hackers adultes peuvent-ils s'assurer de faire émerger une nouvelle génération de hackers libres ?

La réponse d'un père de famille dynamique et avisé ☐



Former la prochaine génération de bidouilleurs open source

Growing the next generation of open source hackers

Dave Neary (Red Hat) – 26 février 2013 – OpenSource.com

(Traduction Framalang : Antoine, cherry, psychoslave, Jeff_, Eijebong, biglittledragoon, goofy, Vero, mathilde, tcit, Quentin, Metal-Mighty, jtanguy, Penguin, Pat, Asta, arnaudbey + anonymes)

En tant que père de trois enfants de 5, 7 et 10 ans, j'ai hâte de partager avec eux les valeurs qui m'ont attiré vers l'open source : partager et créer ensemble des choses géniales, prendre le contrôle de son environnement numérique, adopter la technologie comme moyen de communication plutôt qu'un média de consommation de masse. En d'autres termes :

Comment des bidouilleurs adultes peuvent-ils s'assurer de faire émerger une nouvelle génération de bidouilleurs open source ?

Une des choses que j'ai apprises est qu'il ne faut pas aller trop vite. J'ai mis mes enfants devant Scratch et Sugar à l'âge de 5 et 8 ans, et, bien qu'ils se soient amusés à changer les nombres sur un petit programme que je leur ai montré et aient aimé dessiner leurs propres voitures pour ensuite les diriger sur l'écran, ils étaient trop petits pour comprendre le concept de lier des fonctions entre elles pour arriver à obtenir des comportements plus sophistiqués.

Voici quelques-unes des leçons que j'ai apprises en tant que parent qui, je crois, peuvent être adaptées selon l'âge et les centres d'intérêt de vos enfants.

Un espace à vivre bidouillable

Nous avons encouragé nos garçons à décorer leur chambre, à organiser leurs meubles comme ils voulaient et à avoir leurs propres petits fiefs. Parfois cela nous rend complètement dingues en tant que parents, et, régulièrement, nous devons les aider à ranger, mais leur espace leur appartient.

De même, chaque enfant de plus de 7 ans peut avoir un vrai couteau qu'il peut utiliser pour tailler du bois et couper de la ficelle.

Ingénierie préscolaire

J'adore les jouets qui permettent aux enfants de donner libre cours à leur imagination. En plus c'est génial, parce qu'en tant qu'adulte, je prends autant de plaisir qu'eux à jouer ensemble ! Mes jeux de construction préférés (achetés à peu près au moment où les enfants ont l'habileté nécessaire pour les manipuler) sont les Kapla, les trains en bois, les lots de Duplo, Playmobil, Lego et les voitures Meccano.

Lego et Meccano notamment font un super boulot pour faire des

kits adaptés aux enfants de tout âge. Une autre petite astuce est d'encourager le mélange et d'assembler différentes marques de jouets. Nous avons des ponts Kapla passant par-dessus des trains Ikea et des camions Lego qui transportent des personnages Playmobil.

Les Kapla aussi sont très intéressants. Ce sont des planchettes en bois découpées selon des proportions très précises ; elles sont trois fois plus larges qu'épaisses, et cinq fois plus longues que larges. Avec ces simples proportions et la précision des découpes, il est possible de construire des objets très complexes, comme la Tour Eiffel ou une maison Kapla.

Se lancer dans l'électronique

Nous avons un kit Arduino, et mon aîné commence à avoir le niveau pour comprendre comment câbler un circuit, mais il n'a pas encore découvert comment programmer dans le dialecte C propre à Arduino.

Mais même avant quelque chose de ce genre, les arts et les activités artisanales sont un excellent entraînement pour le DIY (NdT : Do It Yourself, c'est-à-dire « Faites-le vous-même »), et nous avons toujours quelques bâtonnets de glaces ou des pinces à linge et un pistolet à colle pour des cadeaux « faits main ».

Puis vous pouvez laisser traîner des tournevis, pinces, multimètres et autres fers à souder, pour que les enfants puissent désosser leurs vieux jouets, ou des appareils électroniques cassés, réparer les choses par eux-mêmes avec de simples circuits électriques, lorsque que quelque chose ne marche pas, et récupérer des pièces détachées pour les intégrer dans leurs futurs projets. Une supervision parentale est recommandée avec le fer à souder jusqu'à ce qu'ils maîtrisent son utilisation.

Apprendre aux enfants à bidouiller

J'adorerais entendre parler de ressources pour que les enfants apprennent à maîtriser la programmation ! Je connais l'existence de la Code Academy et la Khan Academy qui apprennent aux enfants à coder ; et Scratch and Sugar, que j'ai déjà mentionné.

Merci de partager vos propres conseils sur la manière d'endoctriner la prochaine génération de bidouilleurs open source !

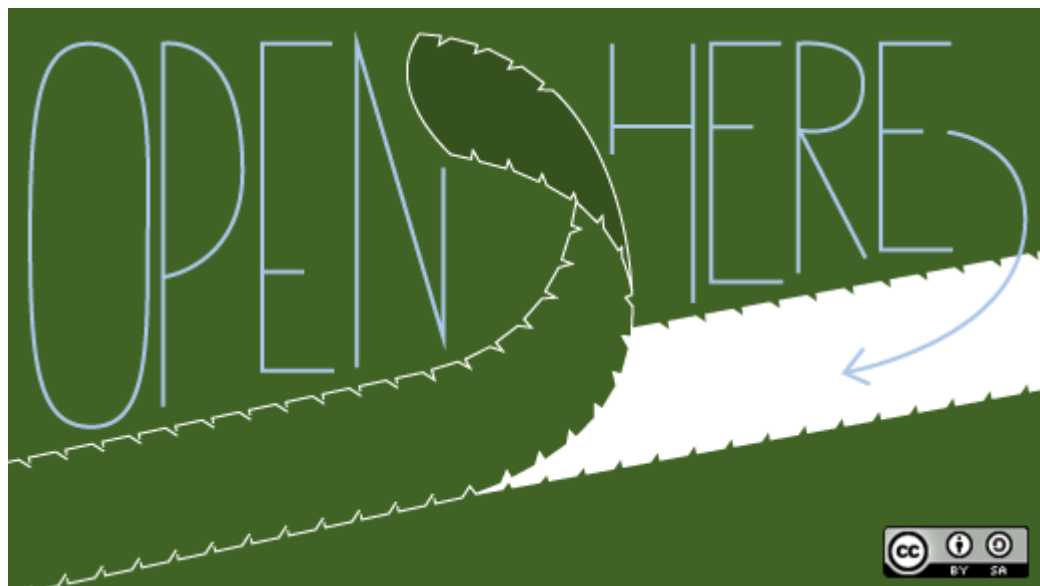
Crédit photo : See-Ming Lee (Creative Commons By-Sa)

10 propositions pour débiter dans le Libre (sans avoir rien à coder)

Il fut un temps où débiter dans « le Libre » se résumait avant tout à coder ou plus modestement installer une distribution GNU/Linux. Aujourd'hui les choses ont bien changé et il existe de multiples autres façons d'y entrer. Framasoft est d'ailleurs là pour en témoigner ☐

Une invitation à venir nous rejoindre en somme...

Remarque : Il s'agit d'une traduction et donc les liens renvoient vers des ressources anglophones. Si vous avez des liens plus locaux à proposer, surtout ne pas hésiter.



10 façons de commencer dans l'open source

10 ways to get started with open source

Jason Hibbets – 29 janvier 2013 – OpenSource.com

(Traduction : goofy, Tibo_R, Xe02, Steph, Alpha, Sylvie, jtanguy, aKa, Liaz, Norore + anonymes)

Par expérience, je sais qu'un grand nombre de personnes veulent découvrir et participer à l'open source, mais ne savent pas par où commencer ; et l'idée que l'on est obligé d'écrire du code pour contribuer à un projet open source constitue une véritable barrière. J'ai donc esquissé 10 façons de commencer avec l'open source et ce sans jamais écrire une seule ligne de code.

Je suis ouvert à toutes idées et ajouts ; il y a sans doute beaucoup plus que 10 façons de contribuer.

10 façons de commencer à utiliser l'open source

1. Utiliser de l'open source dans votre travail quotidien. Téléchargez et installez un navigateur web, un client de messagerie, ou une suite bureautique libres – peu importe le système que vous utilisez. C'est l'une des façons les plus simples de commencer à utiliser des logiciels libres. Je

conseillerai Firefox pour la navigation internet et Thunderbird pour les emails. Utilisez LibreOffice pour votre traitement de texte, vos tableurs et vos diaporamas, vous aurez un équivalent de Microsoft Office gratuit ! J'appelle ces logiciels des applications porte d'entrée, parce qu'une fois que vous commencez à les utiliser, vous allez découvrir d'autres outils *open source* (et vous n'aurez pas envie de revenir en arrière !)

2. Rejoindre un projet *open source*. Je sais que rejoindre un projet *open source* peut faire peur, mais les contributeurs de tous niveaux sont les bienvenus. Les communautés *open source* utilisent des chefs de projets, des graphistes, des communicants, des commerciaux et beaucoup d'autres compétences dans leurs travaux. Si vous souhaitez présenter l'*open source* aux étudiants, voilà une très bonne façon de commencer. On ne sait jamais, s'impliquer et participer activement à un projet *open source* peut améliorer un CV et mener à un emploi.

3. Lire un livre à propos de l'*open source*. Voici un choix de quelques titres auxquels vous pouvez jeter un coup d'oeil : *Open Advice* (*NdT : que nous sommes en train de traduire*), *Coding Freedom*, *The Power of Open*, ou l'un de nos livres numériques. (*NdT : En français il y a évidemment tous les titres de la collection Framabook*)

4. Apprendre à créer et nourrir des communautés de contributeurs. Parcourez le livre en ligne *The Open Source Way*, et partagez vos nouvelles connaissances en créant une communauté ou en rejoignant une existante.

5. Commencer à utiliser les licences Creative Commons. Avant de créer votre nouvelle œuvre d'art, photographie, écrit ou musique, utilisez un copyleft au lieu d'un copyright. En utilisant des licences Creative Commons, vous pouvez partager votre travail avec le monde entier. Vous devrez d'abord choisir celle qui vous correspond, vous pourrez ensuite trouver intéressant de découvrir comment les Creative Commons

sont utilisées dans des environnements aussi variés que les gouvernements, les entreprises ou le journalisme. (NdT : Voir aussi *L'éducation utilise une licence Creative Commons défectueuse*, par R. Stallman sur le Framablog)

6. Commencer l'exploration. Regardez le projet OpenROV et explorez l'océan ou un lac local. Si vous ne voulez pas être mouillé, enfiler une combinaison spatiale et regardez ce que ça fait d'explorer Mars.

7. Bricoler par soi-même et créer quelque chose. Les petites cartes Linux, comme la Raspberry Pi, font des choses incroyables. Découvrez les autres cartes électroniques de création comme les « Makey Makey » (cf cette vidéo) ou une variété de produits électroniques de « SparkFUN ». Si vous êtes dans l'impression 3D, assurez-vous de savoir comment vous pourriez utiliser Inkscape.

8. Devenir créatif. Remplacez Photoshop par GNU Image Manipulation Program (GIMP), InDesign par Scribus, ou utilisez d'autres outils comme MyPaint, Inskape, Audacity et Blender. Si cela vous intéresse, regardez notre présentation en 7 minutes des outils créatifs *open source*. Puis découvrez l'étendue des outils de design en 2012. Assurez-vous d'avoir pris connaissance de nos autres outils tels que Dream Studio, TuxPaint et KDElive pour vos besoins créatifs.

9. Apprendre la programmation. Remarquez que je n'ai pas dit « Apprendre à coder ». Différents outils sont pré-installés sur certains Raspberry Pi et sont utilisés pour apprendre aux enfants à programmer. J'aurais aimé avoir ce genre de choses quand j'ai appris la programmation au lycée.

10. Suivre un cours en ligne. Le mouvement OpenCourseWare, mené par MITOCW, est en train de changer notre mode d'apprentissage. Commencez par regarder ce Webcast sur le MIT OpenCourseWare. Il y a tellement d'événements *open source* dans le champ éducatif: « Moodle » et « School management software

for teachers and students » sont deux de ces nombreuses ressources fantastiques. (NdT : *Exemple en France la présentation du MOOC ITyPA*)

Le fait est qu'il y a énormément de manières de commencer dans l'*open source*. Vous souvenez-vous de la façon dont vous avez débuté ? Partagez l'histoire de votre première expérience avec l'*open source* ou comment vous l'avez présentée à quelqu'un d'autre.