

# Éclairage scientifique

# Éclairage scientifique

Voilà un siècle, les ordinateurs n'existaient pas. Ils sont aujourd'hui plusieurs milliards. Ces ordinateurs et autres machines numériques que sont notamment les réseaux, les téléphones, les télévisions, les baladeurs, les appareils photo et les robots ont changé la manière dont nous :

- concevons et fabriquons des objets ;
- échangeons des informations entre personnes ;
- gardons trace de notre passé ;
- accédons à la connaissance ;
- faisons de la science ;
- créons et diffusons des œuvres d'art ;
- organisons les entreprises ;
- administrons les États ;
- etc.

Si les ordinateurs ont tout transformé, c'est parce qu'ils sont polyvalents. C'est en effet le même objet « ordinateur » qui permet d'utiliser des logiciels de conception assistée par ordinateur, des logiciels de modélisation et de simulation, des logiciels de courrier électronique et de messagerie instantanée, des logiciels d'échange de fichiers, des logiciels de lecture de vidéos et musique, des tables de mixage numériques, des machines à commande numérique, des encyclopédies, des cours en ligne, des bases de données, des blogs, des forums, ou encore des archives numériques. Cette polyvalence s'illustre par le nombre d'outils que les ordinateurs ont remplacé : métiers à tisser, machines à calculer, machines à écrire, téléphones, télévisions, appareils photo, lecteurs multimédias...

Dans tous ces usages, les ordinateurs traitent de l'information, c'est-à-dire qu'ils appliquent, d'une manière systématique, des opérations à des objets symboliques. La recherche d'un mot dans un dictionnaire, le chiffrement et le déchiffrement d'un message secret, l'addition et la multiplication de deux nombres, la fabrication des emplois du temps des élèves d'un lycée ou la construction du planning des pilotes d'une compagnie aérienne, le calcul de l'aire d'une parcelle agricole ou encore le compte des points des levées d'un joueur au tarot sont des exemples de traitements d'informations.

Un procédé systématique permettant de traiter des informations s'appelle un algorithme. Les ordinateurs sont en fait des machines à exécuter des algorithmes et l'informatique est le domaine d'activité scientifique, technique et industriel en rapport avec le traitement automatique de l'information par des machines.

Cet éclairage scientifique sur l'informatique propose, après avoir raconté l'histoire de cette discipline, d'en explorer les 4 concepts fondamentaux : algorithme, langage, machine et information.

## Petite histoire de l'informatique

L'histoire de l'informatique est celle d'une confluence. Deux courants de pensée, centrés l'un autour de la notion d'algorithme et l'autre autour de celle de machine, ont conflué dans les années 1940, pour aboutir à la conception des premiers ordinateurs, machines à exécuter des algorithmes. Deux

autres courants de pensée, centrés autour des notions de langage et d'information, ont, par la suite, rejoint cette confluence.

L'histoire de la notion d'algorithme remonte aux débuts de l'écriture, puisque les premières traces écrites que nous avons retrouvées sont des calculs comptables : les scribes de Mésopotamie connaissaient donc des algorithmes pour effectuer des additions et des multiplications, mais aussi des opérations plus complexes comme des calculs d'intérêts composés, d'héritages... Il est même possible de faire remonter l'histoire de la notion d'algorithme plus tôt, en étendant la notion d'algorithme à des méthodes systématiques qui opèrent sur d'autres objets que de l'information, ce qui permet de concevoir les méthodes de fabrication du bronze ou de la céramique, les recettes de cuisine ou encore les procédés de tissage des étoffes comme des algorithmes. L'histoire de la notion de machine est ancienne, elle aussi, même si on la limite aux machines opérant sur de l'information, puisque les machines d'Héron d'Alexandrie, les carillons des cathédrales, les machines à calculer de Schickard, de Pascal, de Leibniz, le métier à tisser de Jacquard... contiennent déjà des ébauches de traitement de l'information.

L'histoire de l'informatique est foisonnante et s'étale sur des siècles. Voici une sélection de personnages associés à cette histoire qui marquent des jalons importants. Cette sélection permet d'expliquer des concepts de l'informatique en profitant de l'attrait de l'approche historique pour véhiculer des concepts. Elle doit cependant être accompagnée d'une modération : il est délicat d'attribuer clairement telle ou telle contribution à une personne en particulier. Les découvertes se font en effet de manière incrémentale, chaque « inventeur » ayant construit sa propre contribution grâce à celles de ses prédécesseurs et, souvent, de ses contemporains. Cette modération permet surtout de dire que ces personnes, aussi géniales ont-elles pu être, ne sont pas l'unique vecteur de ces contributions, mais qu'on peut tout à fait reconnaître l'importance majeure de leur travail comme une pierre angulaire.



## Muhammad ibn Mûsâ al-Khwarizmi

Al-Khwarizmi est un savant perse, vraisemblablement né dans les années 780. Il évolue dans un empire qui, sous le règne du calife Al-Mamun, encourage les sciences et les arts. Avec d'autres penseurs de son époque, il travaille dans « la maison de la sagesse », une école fondée par le calife à Bagdad pour étudier la géométrie, l'algèbre et l'astronomie. Son travail consiste d'abord à traduire des manuscrits scientifiques ou philosophiques grecs et indiens. Avec ces travaux il transmet à ses contemporains les connaissances de ces civilisations, comme le système décimal de numération. Puis, Al-Khwarizmi écrit ses propres textes scientifiques et en particulier l'ouvrage *Kitâb al-jabr wa al-muqâbala* dans lequel on peut entendre le son *al-jabr* qui est à l'origine du mot « algèbre ». Il cherche à fournir des outils permettant de régler des problèmes de la vie quotidienne. Le mieux est encore de laisser Muhammad al-Khwarizmi expliquer lui-même que ses travaux sont destinés à faciliter les calculs « dont ont besoin les gens dans leurs héritages, dans leurs donations, dans leurs partages, dans leurs jugements, dans leurs commerces et dans toutes les transactions qu'il y a entre eux à propos de l'arpentage des terres, du creusement des canaux, de la géométrie et d'autres choses relatives à ses aspects et à ses arts (...) ». La marque d'Al-Khwarizmi dans l'histoire est fortement liée à celle des mathématiques avec ses travaux sur l'algèbre et la géométrie. Mais son nom est également indissociable de l'informatique car chaque procédure qu'il décrit pour résoudre un calcul est un enchaînement d'instructions qui sont des manipulations sur des nombres. D'ailleurs, le nom d'Al-Khwarizmi a été traduit par *Algorismi* en latin, et donnera plus tard le mot « algorithme ».



## Johannes Gutenberg

Johannes Gutenberg naît vers l'an 1400 à Mayence, en Allemagne. Il apprend l'orfèvrerie puis met son expérience dans le travail des métaux au service de son invention : l'imprimerie. L'histoire de l'imprimerie précède toutefois largement celle de Gutenberg. Cette histoire, qui reflète l'importance pour l'humanité de rendre ses œuvres durables et d'en assurer la diffusion, remonte au VII<sup>e</sup> siècle. Des gravures sur planches de bois sont alors utilisées pour reproduire des textes et des images sur des pages de papier ou de tissu. Les procédés permettant de reproduire une page à l'identique évitent de faire ce travail manuellement et connaissent beaucoup de progrès. Le plus significatif de ces progrès est probablement l'invention par Bi Sheng du caractère mobile d'imprimerie, dont on retrouve les premières traces en Chine, vers 1040. Ces caractères sont alors en terre cuite puis ils évoluent vers le métal. En Europe, Gutenberg applique un principe habituel dans l'histoire des inventions : il perfectionne des techniques existantes et les combine. Ainsi, il crée un alliage métallique qui se révèle très adapté pour l'imprimerie : le « plomb typographique ». Il propose également la « presse à imprimer », plus efficace que les techniques existantes pour appliquer la page sur la planche. Enfin, il met au point une encre plus épaisse et adaptée pour l'imprimerie. Son invention connaît un succès grandissant et se propage rapidement en Europe. L'imprimerie, sans avoir directement influencé l'informatique, entretient avec elle des liens intéressants. Tout d'abord, parmi les innovations de l'imprimerie, penchons-nous sur le caractère mobile d'imprimerie. Il permet de remplir une page avec les caractères de son choix et de façon réversible, alors qu'avant il fallait graver la page tout entière de manière irréversible. La planche devient donc reconfigurable. On peut de surcroît fournir une planche à une autre presse qui pourra à son tour (re)produire la même page. D'une certaine manière, le simple fait de rendre une machine configurable peut être vu comme un ancêtre lointain de la notion de programmation qu'on trouvera bien plus tard dans le métier à tisser de Jacquard, dans les machines de Babbage, et finalement dans les ordinateurs. Un autre lien important avec l'informatique se retrouve dans la révolution culturelle apportée par l'imprimerie. On estime qu'au XVIII<sup>e</sup> siècle, le nombre total de livres imprimés était d'environ un milliard, ce qui a totalement changé l'accès au savoir et donc l'équilibre social.

## Joseph Marie Jacquard

Joseph Marie Jacquard est un inventeur et mécanicien français né en 1752 à Lyon. Pour avoir été tisseur dans l'atelier de son père, il connaît bien le métier à tisser, qui permet de fabriquer des étoffes. Cet outil est alors très répandu, source d'emplois, mais sa manipulation est complexe. Elle mobilise plusieurs personnes pour faire passer les navettes (qui déroulent les fils) au bon moment dans la machine. Jacquard veut mécaniser le métier à tisser pour le rendre plus facile d'emploi et plus performant. Il reprend alors le système de ruban perforé que Basile Bouchon a ajouté au métier à tisser et la mécanisation proposée par Jacques Vaucanson. En améliorant et en combinant ces deux technologies, il contribue à l'expansion du métier à tisser. Le système de cartes perforées permet de définir sur un support solide les instructions de passage des navettes. Les cartes perforées sont ensuite placées dans le métier à tisser, qui les fait défiler à chaque étape de la confection du tissu. Quand une carte perforée passe dans le métier, l'emplacement des trous dans la carte fait changer la position



des fils et guide le passage des navettes. Chaque façon de trouser la carte correspond à un passage différent, et donc à un motif précis sur le tissu. En multipliant les cartes perforées, il devient alors possible d'écrire les instructions pour obtenir le motif complet à l'échelle du lé entier. Comme pour l'imprimerie de Gutenberg, le métier à tisser devient configurable. Un ensemble de cartes peut être copié à l'identique et utilisé sur un autre métier et cela donne exactement le même tissu avec les mêmes motifs. Le même métier avec un jeu de cartes perforées différent donne un autre tissu avec d'autres motifs.

L'invention de Jacquard permet de produire plus de tissus, plus rapidement : elle connaît un grand succès avec plus de 20 000 métiers construits. Parce qu'elle utilise des cartes perforées pour donner des instructions à une machine configurable, cette invention est considérée comme un des ancêtres de l'ordinateur. Le travail de Jacquard illustre bien le schéma habituel des innovations. Jacquard cherche son inspiration dans les technologies les plus avancées de son époque. Il adapte et améliore plusieurs inventions pour les combiner en un seul mécanisme plus facile à installer et à utiliser. L'avancée majeure obtenue par Jacquard sur le métier à tisser permet sa manipulation par un seul ouvrier, au lieu de six précédemment. Ces ouvriers, appelés « canuts » se sont révoltés par la suite en raison du manque de travail, souvent attribué à cette invention.



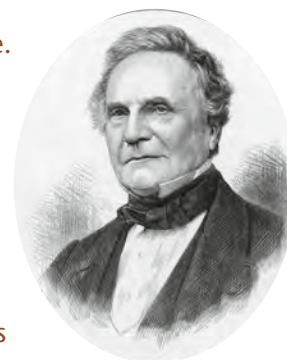
Mécanisme Jacquard (extrait d'un métier à tisser) exposé au musée des Arts et Métiers de Paris.

## Charles Babbage

Charles Babbage, né en 1791, est un mathématicien et inventeur britannique.

À cette époque, bien avant l'invention du GPS, la navigation en mer se fait par l'observation de la position des étoiles dans le ciel, et par comparaison avec leur position attendue à cette date. Cette méthode permet de localiser le bateau sur le globe terrestre. Cela suppose de calculer, à l'avance, la carte du ciel jour après jour pour les semaines à venir. Malheureusement, les calculs sont longs et difficiles et les mathématiciens qui les réalisent risquent de se tromper, ce qui peut provoquer des accidents en mer. En 1821, Charles Babbage présente un projet de machine à la Société royale d'astronomie de Londres, pour accélérer ces calculs et les rendre plus fiables : la « machine à différences ». L'objectif de cette machine est de résoudre automatiquement, par un processus mécanique, les calculs complexes permettant de prévoir la position des étoiles.

Malheureusement pour Charles Babbage, sa machine demande des pièces mécaniques très fines, très précises, qui sont difficiles à fabriquer à son époque. Il n'arrive pas à construire sa machine, mais cela ne l'empêche pas de penser à une deuxième machine, encore plus performante. Babbage veut que cette deuxième machine ne se contente pas de calculer la position des étoiles. Il veut qu'elle soit capable d'exécuter n'importe quelle tâche qui pourrait lui être « décrite ». Pour cela, inspiré par le travail de Jacquard, il prévoit que la « description de la tâche » (une succession de calculs) soit réalisée avec des cartes perforées. Babbage ne réalise peut-être pas totalement le potentiel de son invention





et c'est Ada Lovelace qui va clairement comprendre tout ce que cette « machine analytique » rendrait possible. Malheureusement pour Babbage cette deuxième machine ne sera pas fabriquée non plus, faute de financements. Pourtant, avec elle, il avait posé tous les principes d'un ordinateur, dont la programmation par cartes perforées lui donnant les instructions à exécuter, comme ce sera le cas pour les premiers ordinateurs qui apparaîtront un siècle plus tard.

## Ada Lovelace

Augusta Ada King (Ada Lovelace), est une mathématicienne britannique née le 10 décembre 1815 à Londres. Elle est la fille de lord Byron, poète britannique, qu'elle n'a pas connu. Sa mère, qui tient à ce qu'Ada reçoive une éducation scientifique soutenue, lui fait particulièrement étudier les mathématiques.

Cette éducation est assez atypique pour une jeune fille de cette époque.

À 17 ans, Ada rencontre Charles Babbage et se passionne pour son travail sur la machine à différences, puis sur la machine analytique. Ada Lovelace et Charles Babbage forment un véritable duo scientifique. Ada écrit la plus complète des documentations sur la machine de Babbage et le soutient dans ses difficultés financières. Lui aide Ada à approfondir ses connaissances des mathématiques en



lui recommandant d'étudier avec les plus célèbres mathématiciens du royaume.

Alors que Babbage prévoit l'utilisation de sa machine pour des calculs numériques, Ada parle déjà d'une utilisation plus universelle. À l'époque, le calcul numérique est déjà traité de manière automatique par des machines comme la pascaline: il s'agit d'un ensemble de rouages permettant de faire uniquement des additions et des soustractions. Pour Babbage, sa machine était une sorte de « calculatrice programmable »: on pouvait d'abord lui expliquer le calcul qu'elle devait faire (avec des instructions, le « programme »)

puis lui donner les valeurs sur lesquelles appliquer ces instructions.

Visionnaire, Ada a su comprendre le potentiel de cette machine peut-être encore plus que Babbage lui-même. Elle décrit dans ses manuscrits la possibilité pour cette machine de manipuler des nombres, mais aussi des lettres ou n'importe quel type de symboles. Elle explique comment cette machine pourrait être programmée pour des tâches bien différentes du calcul numérique, comme la composition de morceaux de musique. Dans ses notes, Ada décrit l'ensemble des instructions qu'il faut donner à la machine de Babbage pour qu'elle réalise un calcul particulier: celui des termes successifs de la suite de Bernoulli. En réalisant cela, elle devient la première personne au monde à écrire un programme informatique. Autrement dit, le premier programmeur est une programmeuse!

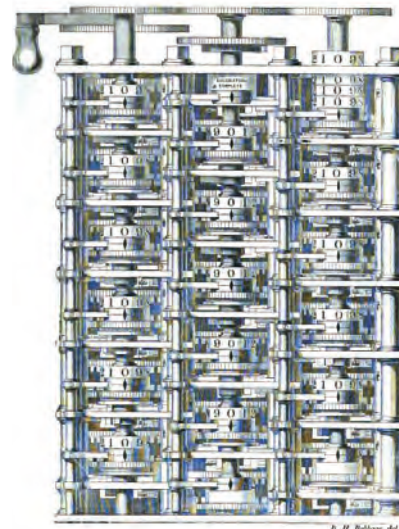


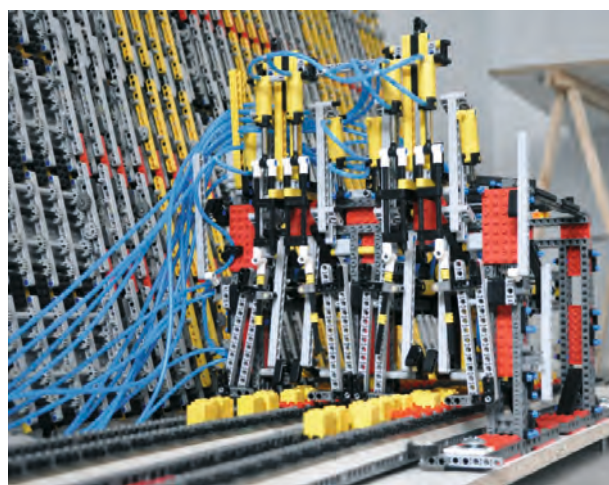
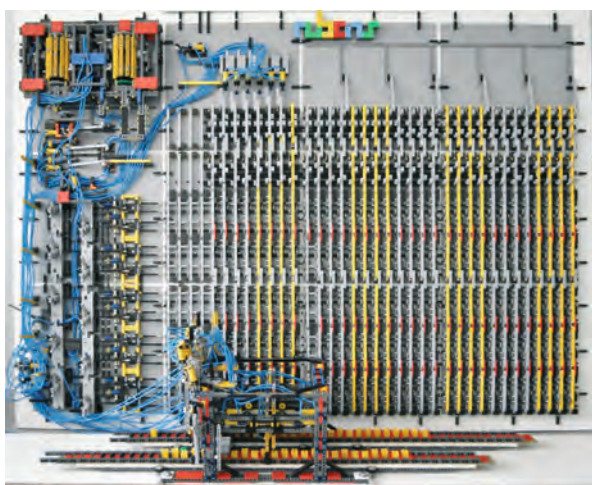
Schéma de la machine à différences de Babbage



## Alan Turing

Alan Turing est un mathématicien et informaticien britannique, né en 1912. Il est souvent présenté comme le père fondateur de l'informatique – bien qu'il soit sans doute vain de chercher un tel père fondateur à ce qui est une création collective.

Ses contributions à l'informatique naissante sont multiples. Dans les années 1930 il donne l'une des premières définitions précises de la notion d'algorithme. Son objectif est alors d'étudier ce qui est « calculable ». Une multiplication, par exemple, est calculable car on peut la réaliser avec un certain nombre d'étapes. Mais quels sont les problèmes calculables ? C'est une question difficile, particulièrement étudiée dans les années 1930, sur laquelle Turing se penche. En étudiant cette question, il propose une machine abstraite, c'est-à-dire une machine qui n'est pas destinée à être fabriquée concrètement, mais qui va lui servir à structurer son raisonnement. Imaginer son fonctionnement lui permet de mieux étudier ce qui est calculable ou pas. Cette machine se décrit très simplement. Elle utilise un ruban qui contient une suite de cases dans lesquelles sont inscrites des données (des 0 et des 1). La machine est capable de lire ce qu'il y a dans les cases, de se déplacer sur le ruban d'une case à gauche ou à droite, et d'écrire dans une case. Avec cette machine, Turing montre qu'il est possible de remplacer n'importe quelle autre machine, en écrivant sur le ruban le « programme » de la machine remplacée (par exemple la multiplication) et ensuite les « données » sur lesquelles appliquer ce programme. Cette façon de mettre ensemble le programme et les données dans une machine est fondamentale et se trouve aujourd'hui dans tous les ordinateurs. Durant la Seconde Guerre mondiale, Turing joue un rôle majeur dans la construction de machines électromécaniques permettant de déchiffrer les messages secrets utilisés par l'armée allemande, sans en connaître la clé. Après la guerre, il travaille à la conception des premiers ordinateurs britanniques au National Physical Laboratory, puis à l'université de Manchester. Il est aussi à l'origine de nombreuses autres idées sur le caractère phénoménologique de l'intelligence, sur la genèse des formes en biologie, sur la résolution approchée d'équations différentielles avec un ordinateur ou sur les preuves de programmes. Dans ce dernier cas, la problématique est la suivante : on peut faire des tests sur un programme pour vérifier sa stabilité et son exactitude. Si on trouve une erreur, on cherche alors à réparer le programme ; mais si on ne trouve pas d'erreur par cette méthode, cela



Machine de Turing réalisée en Lego® (projet Rubens, ENS Lyon) : vue globale (à gauche) et détail de la tête de lecture - écriture (à droite).

veut-il dire qu'il n'y a pas d'erreur ou qu'on n'a pas su la trouver? Existe-t-il une preuve mathématique que le programme est réellement sans erreur?

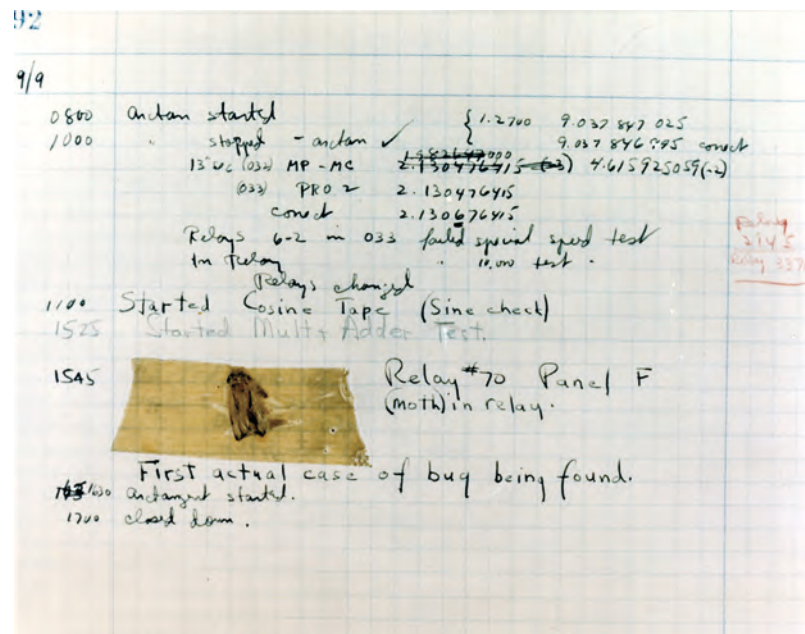
Condamné à la castration chimique en raison de son homosexualité, considérée comme un crime dans l'Angleterre de cette époque, Alan Turing s'est suicidé à 41 ans.



## Grace Hopper

Grace Hopper est une mathématicienne américaine, née en 1906. Elle grandit en montrant une véritable curiosité pour les sciences et la technologie (par exemple, en démontant les réveils de sa maison les uns après les autres, jusqu'à arriver à en remonter un entièrement... à l'âge de 7 ans). Ses parents encouragent son goût pour les sciences tout en veillant à valoriser ses choix plutôt que de suivre la tradition en matière d'éducation des jeunes filles américaines. Ainsi, Grace sera une des rares femmes de son époque à étudier à l'université Yale et à obtenir un doctorat de mathématiques.

Pendant la seconde Guerre mondiale, elle entre dans la marine, où elle est nommée lieutenant, et rejoint l'équipe qui travaille sur le Mark 1, le premier ordinateur des États-Unis. Grace Hopper fait alors partie du premier groupe capable de programmer cette machine. Cet ordinateur prend ses instructions et ses données à l'aide de cartes perforées, comme cela était prévu sur la machine de Babbage, et comme le faisait le métier à tisser de Jacquard environ deux siècles auparavant. Dans les années 1950, Grace Hopper est motivée par l'idée que l'informatique devrait être accessible à un plus grand nombre de personnes, et pas seulement aux spécialistes des ordinateurs et aux mathématiciens. Elle défend l'idée que les cartes perforées sont un frein au développement de l'informatique et travaille sur un projet de « langage de programmation » qui serait proche de l'anglais et qui faciliterait l'écriture des programmes. La difficulté est alors de concevoir le programme capable de transformer des instructions de haut niveau (le langage proche de l'anglais et facilement compréhensible par un humain) en instructions de bas



Découverte d'un bug («insecte») par Grace Hopper, en 1947



niveau (le langage de la machine). Un tel programme est aujourd'hui appelé « compilateur » et c'est en partie à Grace Hopper que nous en devons l'invention. Son langage de programmation, appelé COBOL, sera le plus utilisé entre 1960 et 1980.

Grace Hopper n'est pas célèbre uniquement pour avoir rendu les ordinateurs plus accessibles aux non-spécialistes. Un jour, alors qu'elle programme ses ordinateurs avec des cartes perforées, l'un d'eux tombe en panne. Allant voir ce qui a pu se produire, elle trouve une mite coincée dans un des trous d'une carte perforée. Elle colle alors sur son rapport le morceau de carte avec un commentaire désignant l'insecte comme coupable de la panne. « Insecte » se disant *bug* en anglais, ce terme a été utilisé auparavant par les scientifiques, comme Thomas Edison, pour désigner une erreur due à l'intervention indésirée d'une de ces petites bêtes dans leurs expériences. En l'appliquant à l'exécution échouée d'un programme informatique, Grace en a fait le terme le plus utilisé pour qualifier cette situation.

### Une brève histoire du réseau Internet

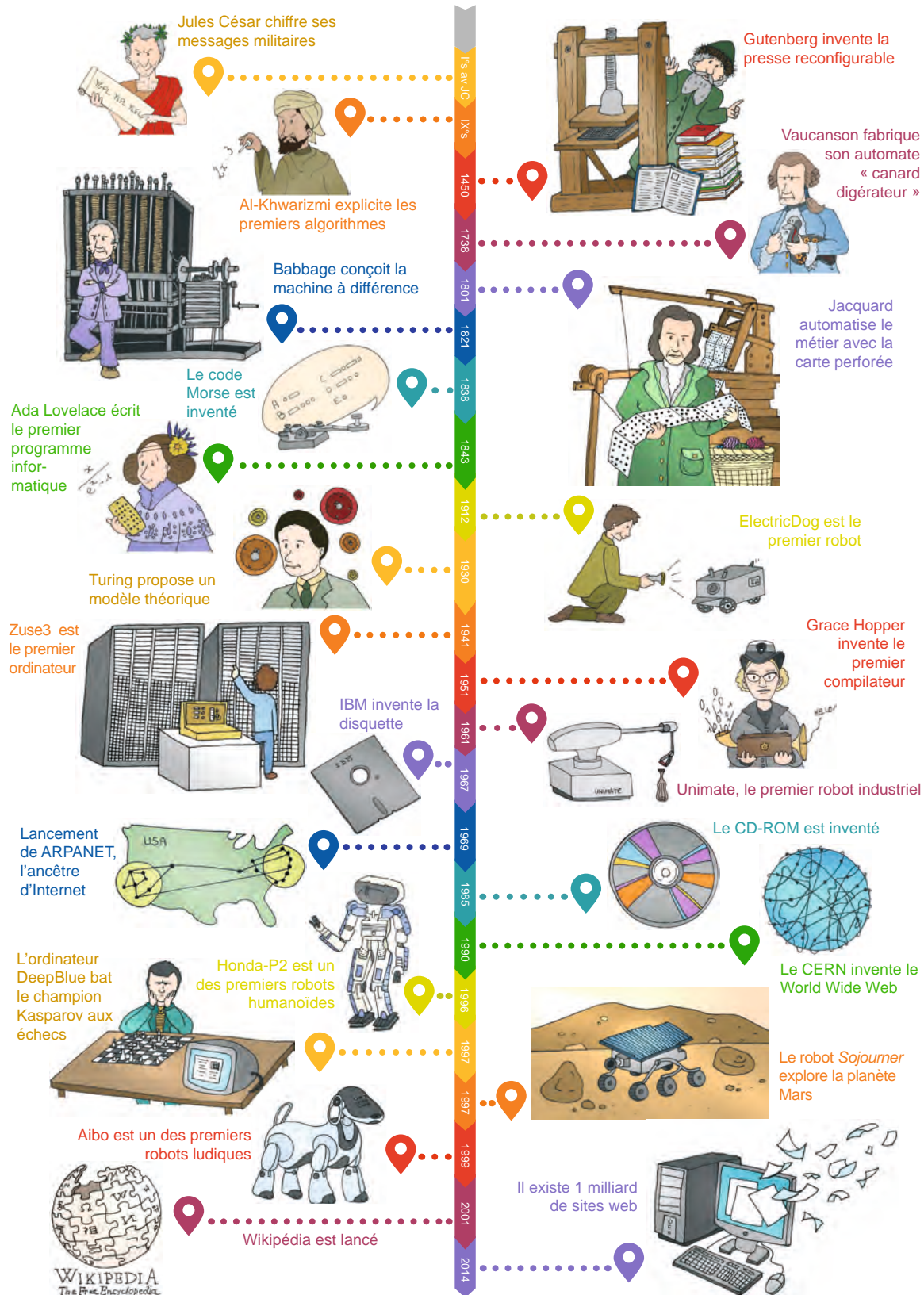
L'idée d'une connexion de nombreux ordinateurs en réseau naît dans les années 1960 aux États-Unis : un chercheur, Leonard Kleinrock, formalise en 1961 un moyen de transmettre des données entre ordinateurs en découpant ces données en paquets, en transmettant les paquets par des voies qui peuvent être différentes et en les rassemblant dans l'ordre voulu au niveau du récepteur. C'est le principe de la méthode utilisée encore aujourd'hui pour transmettre des données – comme un message électronique – sur Internet (voir illustration page 34). Dès l'année suivante, Joseph Licklider promeut l'ordinateur comme un outil de communication et de partage des ressources, et propose la construction d'un réseau d'ordinateurs connectés entre eux. Licklider intègre l'ARPA (Agence des projets de recherche avancée), organisme créé en 1958 pour assurer la supériorité militaire et technologique des États-Unis, à la suite du lancement par l'URSS du premier *Sputnik*. Il devient le directeur du Bureau des techniques de traitement de l'information de l'ARPA, et c'est à ce poste qu'il lance la construction d'un réseau expérimental reposant sur le principe de la transmission des données par paquets. Ce réseau, nommé Arpanet, voit le jour en 1969. Il comporte seulement 4 ordinateurs, qui s'échangent des données à un rythme de 50 000 bits par seconde (6 Ko/s).

Les premiers utilisateurs d'Arpanet développent des programmes permettant d'envoyer des messages d'un ordinateur à un autre, via le réseau. Le premier e-mail, envoyé en 1972, utilise déjà le caractère @ pour séparer le nom du destinataire et son adresse. Les principales fonctions des messageries électroniques – tri, réponse, transfert de messages – sont inventées dès cette époque.

La première démonstration publique d'Arpanet a lieu en 1972, lors d'une conférence internationale. C'est un grand succès qui contribue à l'expansion du projet, à la naissance d'autres réseaux et fait naître l'idée d'une interaction entre réseaux. Une telle interaction nécessite des modifications dans le principe de fonctionnement d'Arpanet : la solution développée est le protocole TCP/IP, encore utilisé aujourd'hui sur le « réseau de réseaux » qu'est Internet.

Tim Berners-Lee, utilisateur du réseau du CERN (appelé Cernet), imagine en 1989 de faciliter la navigation au sein du réseau par l'usage d'un navigateur et de liens hypertextes. Il propose la création d'une toile permettant d'aller d'un contenu à un autre par des voies multiples. Ce projet, approuvé par le CERN, est nommé le « World Wide Web », ce qui signifie littéralement « toile d'araignée mondiale ». C'est l'un des réseaux d'Internet.

# Frise chronologique



La séance « Bilan: qu'est-ce que l'informatique? » pour le cycle 3 (page 335) propose aux élèves de reconstituer une frise chronologique similaire à partir d'études documentaires portant sur l'histoire de l'informatique

### Une brève histoire du réseau Internet (suite)

Les projets foisonnent à partir des années 1990, entourés de spéculation boursière, jusqu'à éclatement de la « bulle » Internet en 2000 : le premier navigateur Web possédant une interface graphique voit le jour en 1993. Il sera rapidement supplanté par d'autres, eux-mêmes supplantés par d'autres... L'annuaire Internet Yahoo! est créé en 1994 puis il devient un portail international. La start-up Google est créée en 1998, puis ce sont eBay ou encore Amazon, entreprises géantes de l'Internet.

# Algorithmes, langages et programmes

## L'ambiguïté du verbe « coder »

Le verbe « coder » est très à la mode (on le trouve par exemple dans la formule « lire, compter, écrire et coder » au sujet de l'introduction de l'informatique à l'école). Ce mot pouvant être porteur de plusieurs ambiguïtés, discutons-en ici.

La première de ces ambiguïtés est certainement la plus facile à clarifier. En effet, il arrive parfois que le mot « coder » soit utilisé pour faire référence au fait de modifier le contenu d'un message pour le rendre illisible à quiconque ne connaîtrait pas le « code secret ». Dans la suite de ce manuel, cette action sera désignée comme du « chiffrement », soit l'action de « chiffrer » le message pour le rendre illisible à quiconque ne connaît pas la « clef de chiffrement » permettant non seulement de « chiffrer » le message, mais aussi de le « déchiffrer » pour le rendre à nouveau lisible. Plus généralement, le mot « coder » fait référence à l'action de donner des instructions à une machine, comme un ordinateur. Cette activité peut également être désignée par le terme de « programmation ». Les instructions seront alors interprétées par cette machine, qui va les exécuter pour produire un résultat.

Enfin, « coder » peut être utilisé pour représenter une information à l'aide de symboles (par exemple, écrire un texte en binaire, à l'aide de 0 et de 1). Selon les cas, on utilisera les termes « encoder » ou « decoder ». Dans ce guide pédagogique, c'est ce sens qui a été retenu pour le verbe « coder », bien que le grand public utilise plutôt « coder » pour « programmer ».

Le mot « coder » véhicule toutefois une seconde ambiguïté, qui ne concerne pas sa définition mais plutôt l'idée que l'on peut se faire de l'informatique à travers l'activité de programmation. Car l'informatique, ce n'est pas que de la programmation. Il s'agit certainement de l'activité la plus visible. Celle qui, dans la culture populaire, représente concrètement le pouvoir qu'ont les programmeurs, et qui illustre la séparation entre ceux qui subissent les machines en tant qu'utilisateurs (même tacites) et ceux qui les font fonctionner. C'est, par exemple, l'image souvent caricaturale du programmeur qui enchaîne des lignes de code très rapidement dans les films. C'est, encore, l'image d'un écran qui fait défiler

des pages entières de code en arrière-plan d'un informaticien dans un reportage ou un documentaire (probablement parce que cela donne un côté impressionnant, ou très spécialisé). Cette image pourrait participer à la diffusion de l'idée que l'informatique est difficile d'accès et réservée à une élite ou à des esprits qui lui sont prédestinés. Heureusement, ce n'est pas le cas. De toute évidence, programmer fait entièrement partie de l'informatique et, pourtant, cette activité n'en est pas le seul engrenage. Programmer permet effectivement de voir se concrétiser ses idées. Mais ces idées peuvent naître et mûrir en dehors de toute forme de programmation, en utilisant le raisonnement comme moteur principal. Ce raisonnement, associé à la nature mécanique des ordinateurs et de leur fonctionnement, constitue la « pensée informatique » et permet de mettre au point les algorithmes qui font aujourd'hui tourner une grande partie de notre monde.

## Qu'est-ce qu'un algorithme ?

L'omniprésence des algorithmes n'est plus à démontrer. Quand on fait une requête sur un moteur de recherche, ce sont plusieurs algorithmes qui sont mis en jeu pour trouver les pages les plus pertinentes possible. Quand les marchés financiers voient des opérations se faire par milliers en une seconde, de la part d'un même trader, ce n'est pas un trader humain qui donne les ordres, mais un algorithme de trading à haute fréquence. Quand un gouvernement veut analyser de très grandes masses de données concernant les activités humaines, afin d'essayer d'y détecter des comportements suspects, alors il se tourne vers les algorithmes<sup>4</sup>. Les algorithmes jouent un rôle important dans nos vies, qu'on en fasse le choix ou pas.

Pour illustrer la notion d'algorithme, l'exemple de la recette de cuisine est assez bien adapté. Après tout, il s'agit d'une série d'instructions à appliquer pour obtenir un résultat. Et les analogies entre une recette de cuisine et un algorithme peuvent aller plus loin. Une recette peut être diffusée dans le monde ou bien elle peut être gardée secrète. Elle peut être recopiée à la main ou transmise oralement. Mais tant qu'elle n'est pas incarnée par des actions en cuisine, elle reste un concept. En informatique, ce concept est un algorithme. Il peut rester écrit à la main, sous une forme plus ou moins formelle mêlant des explications et des symboles mathématiques. Tant qu'il reste sous cette forme, sans être appliqué, il est un savoir qui ne produit aucun résultat. La puissance d'un algorithme se manifeste quand il est interprété et concrètement déroulé, que ce soit par un humain ou par une machine.

Pour la vitesse d'exécution, c'est autre chose... Un algorithme simple peut être interprété par un être humain, en suivant les étapes et en notant les résultats intermédiaires, s'il y en a. C'est le cas, par exemple, pour l'algorithme de la division que les élèves peuvent appliquer systématiquement une fois qu'ils l'ont appris. Ils peuvent alors dérouler n'importe quelle division à la main en appliquant rigoureusement les étapes de l'algorithme. Ils notent les valeurs intermédiaires et recommencent les opérations jusqu'à ce que les conditions d'arrêt du calcul soient atteintes. Pour un algorithme plus compliqué cela peut devenir délicat et pour la plupart des algorithmes qui se déroulent en ce moment autour de nous, cela devient inenvisageable. Les machines, elles, peuvent faire dérouler ces algorithmes à des vitesses chaque année plus grandes. Pour que cela soit possible il faut réunir au moins deux conditions :

1. Concevoir et écrire un algorithme qui permet d'atteindre le résultat visé.
  2. Traduire cet algorithme dans un programme que la machine peut interpréter.
- Ces deux activités sont certainement le cœur de la production d'objets numériques. La programmation

---

4. La question de la pertinence des algorithmes pour cet objectif n'est pas le sujet ici, mais ce sont bien des algorithmes qui sont envisagés pour traiter ces données colossales qui ne peuvent pas être analysées « à la main ».



peut connaître différents niveaux de qualité, en fonction de l'expérience du programmeur et de sa connaissance du langage utilisé. Un programme peut être écrit de plusieurs manières tout en suivant le même algorithme. Certains programmeurs connaissent tellement bien leurs langages préférés qu'ils peuvent obtenir des gains de performance importants, en écrivant des programmes plusieurs fois plus rapides que les programmes écrits par des programmeurs novices. Cela n'est pourtant pas toujours suffisant pour obtenir des programmes efficaces.

Car les algorithmes, eux aussi, peuvent être écrits de plusieurs manières. Simplement parce qu'ils peuvent être conçus de plusieurs manières. Un algorithme peut être particulièrement bien pensé, mais il peut aussi être basique et utiliser des choix malvenus. Ce que nous allons voir par la suite, c'est qu'un algorithme mal conçu ne peut pas être compensé par une programmation efficace, ou un ordinateur ultra performant. Un algorithme bien conçu, en revanche, même programmé par un novice sur un ordinateur vieillissant, peut offrir des performances radicalement supérieures. Cette conception de la méthode, ce raisonnement permettant de produire l'algorithme le plus efficace possible, est une véritable clef en informatique. La discipline qui lui correspond est l'algorithmique. Explorons ensemble quelques fondements de cette discipline, à la lumière de la petite histoire de l'informatique que nous venons de voir.

## Langages

Les premiers ordinateurs étaient programmés avec des cartes perforées. C'est d'ailleurs ce que Babbage avait prévu pour sa machine analytique. Comme nous l'avons déjà vu, en voulant rendre la programmation plus accessible, Grace Hopper a compris les limites des cartes perforées comme interface entre la machine et ceux qui la programment. Pour dépasser ces limites, elle a contribué à la création des compilateurs. Il s'agissait de pouvoir exprimer un programme dans un langage proche du langage humain. Toute la difficulté étant de traduire ce programme dans le langage de la machine, pour qu'elle exécute les actions correspondantes. Cette traduction est connue sous le nom de « compilation ». Le programme qui se charge de la compilation est un compilateur. Il faut, pour compiler un programme, relever un grand nombre de défis, parmi lesquels :

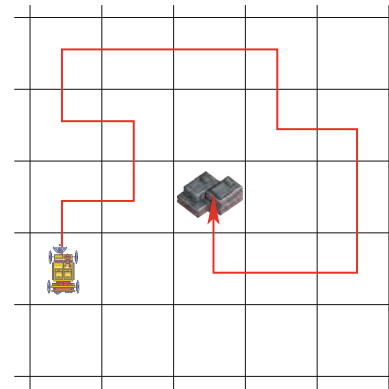
- Reconnaître et séparer les mots et les éléments du langage pour en faire des entités avec lesquelles travailler par la suite. Par exemple la ligne de code `resultat=a+3` doit permettre de reconnaître la variable `resultat`, l'opérateur `=`, la variable `a`, l'opérateur `+` et le nombre `3`.
- Détecter les incohérences dans le langage et s'assurer que le programme est correct d'un point de vue syntaxique ou grammatical (donc vérifier que le programme respecte les règles du langage dans lequel il est écrit).
- Détecter les incohérences sémantiques. Par exemple, si deux variables portent le même nom alors il est impossible de traduire le programme pour la machine (il y a une ambiguïté qui rend impossible la compilation).
- Générer effectivement le code, dans le langage de la machine, à partir des éléments construits dans les étapes précédentes.

L'apport des compilateurs est majeur dans l'histoire de l'informatique. Il est évidemment beaucoup plus facile d'utiliser des mots, des nombres et des opérateurs pour programmer que d'inscrire les états de la machine, un par un, sur des cartes perforées. Cet apport se traduit aujourd'hui par une multitude de langages de programmation, parmi lesquels figure *Scratch*, le langage que nous avons choisi pour initier les élèves à la programmation. Les différences entre les langages peuvent être très importantes. Dans le projet pédagogique proposé pour le cycle 3, une séance (page 215), illustre la diversité des

langages utilisables pour guider un robot : les langages autocentrés et allocentrés.

Les langages de programmation possèdent un vocabulaire extrêmement limité et très précis, sans la moindre ambiguïté. Dans notre exemple, le robot peut être guidé par 4 instructions allocentrées (absolues) : Nord (ce qui signifie « avance d'un pas vers le nord »), Sud, Est, Ouest. Dans un langage autocentré (relatif), le robot peut répondre à trois instructions : avance (avance d'un pas droit devant toi), droite (c'est-à-dire « tourne sur toi-même de 90° vers la droite »), gauche. On peut même simplifier encore l'instruction gauche en droite-droite-droite, ce qui réduirait encore le vocabulaire à deux éléments seulement. Il est possible de passer d'un langage à un autre, mais l'exercice n'est pas simple, et le faire en classe illustrera bien le mérite des compilateurs !

Une fois mise en place cette contribution majeure que sont les compilateurs et donc des langages de programmation, il nous reste à... programmer ! Et pour cela, le plus sage est encore de commencer par bien concevoir son algorithme avant de le traduire en programme (et le compilateur va ensuite traduire ce programme pour la machine). Voyons maintenant comment ces fameux algorithmes peuvent naître, de quoi ils sont faits et surtout... comment tout le monde peut en produire !



Quel langage utiliser pour piloter ce robot ? (extrait de la progression pédagogique cycle 3, page 215)

## Algorithmes, raisonnement et pensée informatique

Nous savons aujourd'hui que les algorithmes peuvent tous s'écrire avec quelques instructions élémentaires qui sont assemblées à l'aide de structures de contrôle.

L'instruction élémentaire la plus courante est l'affectation d'une variable, par exemple  $a = 3$  dont l'exécution consiste à affecter à la variable  $a$  la valeur 3. Cette notion d'affectation demande donc de comprendre le concept de variable : « boîte qui contient une valeur que l'on peut utiliser et modifier au cours de l'exécution du programme ».

Le nombre de fois où une vidéo a été visionnée sur une plateforme de streaming est un exemple de variable. Chaque fois que la vidéo est visionnée, la variable « nombre de vues » est modifiée (par exemple, l'instruction nombre de vues = nombre de vues + 1 permet d'affecter à la variable nombre de vues sa nouvelle valeur, soit un de plus que précédemment).

D'autres instructions élémentaires sont l'affichage d'un message à l'écran, le déplacement d'un lutin, etc.

Ces instructions élémentaires sont ensuite assemblées avec des structures de contrôle, qui permettent de fabriquer des instructions complexes en assemblant des instructions plus simples. Les principales sont :

- la séquence, qui permet d'exécuter une instruction puis une autre ;
- le test, qui permet d'effectuer une instruction ou une autre, en fonction d'une condition. Par exemple, dans un lave-linge, si le poids du linge est inférieur à 2 kg alors lancer le programme économique, sinon lancer le programme classique.
- la boucle, qui permet de répéter une instruction plusieurs fois.

Mais d'autres structures de contrôle existent, par exemple en programmation événementielle (en particulier dans *Scratch*), le déclenchement d'une instruction à la suite de la survenue d'un événement. Les premiers ordinateurs sont capables d'exécuter des algorithmes composés de ces ingrédients...

ce sont donc des machines capables d'exécuter tous les algorithmes du monde : des **machines universelles**. Gageons que pour les pionniers de l'informatique, voir s'incarner les algorithmes dans des machines universelles devait être exaltant.

Comme ces pionniers, l'apprenti programmeur découvre les capacités de son ordinateur de façon progressive. Il commence avec des algorithmes simples bien connus comme des opérations mathématiques (l'addition, la multiplication, etc.). Ensuite, les algorithmes se complexifient, reproduisant des suites mathématiques comme celle de Fibonacci ou de Bernoulli. Justement, c'est en écrivant un programme pour calculer la suite de Bernoulli sur la machine analytique de Babbage qu'Ada Lovelace est devenue la première personne au monde à écrire un programme informatique. Il existe bien d'autres programmes encore. La machine universelle nous permet d'appliquer des algorithmes connus, mais également de rechercher des algorithmes pour résoudre des problèmes qu'on ne sait pas encore résoudre. Cependant, le fait que l'on trouve un algorithme signifie-t-il forcément que cet algorithme nous donnera toujours une solution ? Pour y voir plus clair, explorons ce que l'on appelle la « pensée informatique ».

La **pensée informatique** est un ensemble de connaissances et d'attitudes mises en œuvre pour comprendre le monde qui nous entoure et nous aider à diriger nos actions. Elle couvre une vaste étendue de concepts, et nous allons tenter d'en éclairer ici quelques-uns<sup>5</sup>.

Tout un chacun applique, tous les jours, des algorithmes... parfois sans le réaliser vraiment. Nous avons établi des stratégies, des habitudes, des raisonnements que nous appliquons à nos activités du quotidien. Dans un magasin, un samedi de courses, on peut choisir de parcourir les allées du début à la fin du magasin en vérifiant, pour chaque allée, si elle contient un produit qui est sur sa liste de courses. Mais pourquoi ne pas se faciliter la vie en rédigeant cette liste au préalable, en fonction de sa connaissance du magasin, pour que la liste présente les produits dans le bon ordre ? Pour aller travailler, on peut prévoir son trajet en fonction du jour de la semaine parce qu'il est connu que le mardi et le jeudi il faut absolument éviter la rue Kolmogorov qui est saturée. Enfin, quand on cherche un mot dans un dictionnaire, on l'ouvre à l'endroit le plus probable en fonction de la position de la première lettre de ce mot dans l'alphabet (vers le début pour la lettre « c » ou vers le milieu pour la lettre « p », etc.)... puis, par tâtonnements, on recule ou on avance de quelques pages, en fonction de la lettre qui débute les mots de la page courante. Et lorsque la première lettre du mot est trouvée, on recommence avec la deuxième lettre, et ainsi de suite.

Les ordinateurs peuvent nous aider à traiter ces tâches quotidiennes ! Il s'agit alors de poser le problème de manière formelle, puis d'écrire un algorithme basé sur les quatre ingrédients vus plus haut (instructions, boucles, tests et variables) et permettant de le résoudre. Pour automatiser l'exécution de ces algorithmes, il faut les traduire en programmes et laisser une machine dérouler leurs instructions. Une partie de la pensée informatique consiste à rationaliser de tels procédés pour qu'une machine soit capable de les mettre en œuvre.

Voyons maintenant quelles sortes de défis nous pouvons résoudre grâce à la pensée informatique et aux algorithmes. Il y a les défis que l'on peut résoudre avec des algorithmes qui s'inspirent du raisonnement humain. Ce sont des algorithmes comme, par exemple, celui qui établit la liste des courses dans l'ordre des allées du magasin. Beaucoup d'algorithmes sont conçus de cette manière. La personne qui met au point l'algorithme essaye de formaliser son propre raisonnement. C'est un peu comme expliquer la méthode à quelqu'un d'autre, en voulant s'assurer qu'il n'y aura aucune erreur d'interprétation.

---

5. L'article « La pensée informatique » [https://interstices.info/jcms/c\\_43267/la-pensee-informatique](https://interstices.info/jcms/c_43267/la-pensee-informatique) donne une vue d'ensemble de ce que recouvre cette expression.

Ensuite, il y a les défis que l'on va résoudre avec un algorithme inspiré du raisonnement ou des pratiques humaines, mais en y ajoutant une étude qui permet de rendre l'approche plus rigoureuse et plus générale. Reprenons l'exemple de la recherche d'un mot dans le dictionnaire. Il existe un exercice très similaire qui consiste à chercher son nom, ou son prénom, dans une liste (par exemple, pour savoir si on est reçu à un concours). Essayons de formaliser la manière dont on réalise cette tâche.

Imaginons une liste, non triée, de quelques centaines de prénoms. Il faudra à tout un chacun quelques minutes pour y retrouver son nom (ou être sûr qu'il n'y est pas). Dans un second temps, si l'on fournit la même liste triée par ordre alphabétique, l'exercice est beaucoup plus rapide !

Cette différence d'efficacité vient du fait que, quand la liste est triée, il existe une méthode qui est bien plus efficace que si la liste n'est pas triée. Quand elle n'est pas triée, on peut chercher un peu partout au hasard, ou bien on parcourt systématiquement la liste du début à la fin en vérifiant les prénoms un par un. En revanche, quand la liste est triée, on repère l'endroit où se trouvent les prénoms qui commencent par la première lettre de son prénom. Ensuite la deuxième lettre, etc.

Cette approche fonctionne, mais peut encore être perfectionnée. Car si le nombre d'objets est très grand (comme dans un dictionnaire, voire un répertoire téléphonique national avec des millions d'entrées) alors on peut faire de grands bonds en avant ou en arrière pour gagner du temps. Il existe un algorithme de recherche dans une liste triée très efficace (le plus efficace connu pour répondre à ce défi) : la « dichotomie », qui tire son nom du grec pour l'expression « couper en deux ». Pour trouver un item dans une liste triée, on coupe la liste en son milieu ; si l'item était au milieu, on l'a trouvé, sinon on sait dans quelle moitié de la liste triée il se trouve ; on recommence donc la dichotomie sur la demi-liste restante. Et ainsi de suite. À chaque étape, la liste dans laquelle on cherche l'information est deux fois plus petite.

Une liste de 20 prénoms est fouillée en 4 coups dans le pire des cas : le nombre d'opérations dans le pire des cas est donné par le nombre de fois qu'on peut diviser 20 par 2 avant d'arriver à 1 (on arrondit la première division si le nombre de cases est impair). Sans dichotomie, il aurait fallu (dans le pire des cas) 20 itérations pour lire successivement chaque prénom, du premier au dernier, pour se rendre compte que le sien n'y était pas. La puissance de la dichotomie se voit vite dans le cas de listes bien plus longues : une liste de 300 000 items est fouillée par dichotomie en 19 opérations seulement. Imaginons maintenant que chaque opération de lecture dans le tableau demande une milliseconde à l'ordinateur. Le tableau suivant estime le temps d'application des deux algorithmes dans le pire des cas :

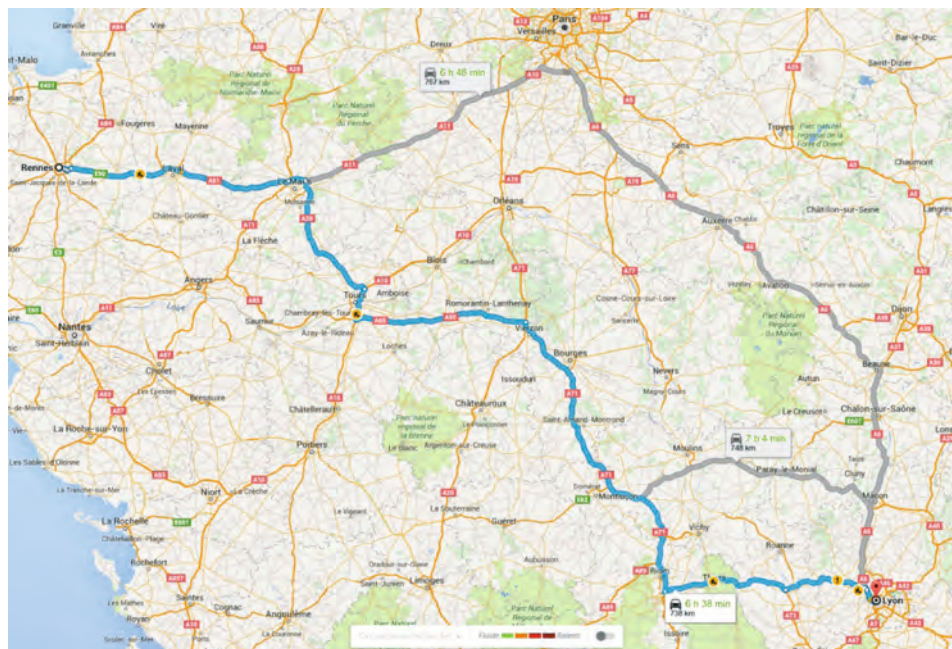
Nombre de prénoms dans la liste	300	3 000	30 000	300 000	3 000 000
<b>Algorithme 1 : parcours de toute la liste, du premier au dernier</b>	0,3 s	3 s	30 s	300 s	3 000 s
<b>Algorithme 2 : dichotomie</b>	0,008 s	0,012 s	0,015 s	0,018 s	0,022 s

Comparaison du temps mis, dans le pire des cas, pour rechercher un prénom dans une liste triée, à l'aide de 2 algorithmes différents. L'algorithme le plus simple (on parcourt la liste du début à la fin) met 10 000 fois plus de temps pour s'exécuter si la taille de la liste est multipliée par 10 000 (on passe de 3 dixièmes de seconde à 1 heure), tandis que la recherche dichotomique n'est rallongée que d'un centième de seconde.

Les approches pour mettre au point des algorithmes sont nombreuses. Nous ne pouvons pas les passer toutes en revue ici. En revanche, il est important de se pencher sur des défis d'un type particulier, pour lesquels on sait écrire un algorithme correct, mais qui ne connaîtront pourtant jamais de solution. Ou plutôt devrions-nous dire qu'on n'obtiendra pas la solution en un temps raisonnable. Il s'agit là



d'un aspect fondamental de l'algorithmique. Considérons par exemple le « problème du voyageur de commerce » que nous décrivons lors d'une séance débranchée pour le cycle 3 (page 273). Il s'agit de trouver, pour un ensemble de villes dans lesquelles doit se rendre un voyageur de commerce, l'ordre dans lequel visiter ces villes de façon à passer le moins de temps possible sur la route. S'il n'y a que 2 villes à visiter, la réponse est simple, avec un seul trajet possible pour les relier. Avec 3 villes, il y a 6 trajets possibles à comparer (dans ce décompte, on distingue des trajets « identiques » mais parcourus dans 2 sens différents, par exemple Lyon-Bordeaux-Marseille et Marseille-Bordeaux-Lyon). Avec 4 villes, il y a 24 trajets possibles. Avec 20 villes, il y a déjà plus de deux milliards de milliards de possibilités. Un ordinateur mettrait plusieurs milliards d'années à tester toutes les possibilités pour y trouver à coup sûr la meilleure ! Ce problème se retrouve dans de nombreuses applications comme la tournée du facteur (ou des éboueurs, des déneigeuses...) : dans quel ordre parcourir les rues dans lesquelles il faut passer ? Il s'agit du même problème que dans les usines où des bras articulés (des robots) doivent visser des boulons sur une plaque à différents endroits : dans quel ordre visser les boulons pour minimiser les déplacements et donc économiser de l'énergie et d'allonger la durée de vie du robot ? Quand le nombre « d'objets » (villes à visiter, boulons à visser...) augmente alors le nombre de possibilités explose. On parle d'« explosion combinatoire » ou de « croissance exponentielle ». Au-delà d'un nombre d'objets relativement petit, la solution à tous ces problèmes est... tout simplement inconnue ! Bien entendu, pour réduire le temps de calcul, on peut envisager de prendre un ordinateur plus performant ou un autre langage plus efficace. Le nouveau programme, écrit par un programmeur expérimenté dans un langage plus performant, sera peut-être dix fois ou même cent fois plus rapide. Mais, quand le temps de calcul est estimé à 10 milliards d'années, on passera peut-être à « seulement » cent millions d'années. Imaginons même, pour la beauté du raisonnement, qu'on trouve une architecture informatique, un langage de programmation, un système qui permette de calculer la solution exacte au trajet entre 20 villes en une minute. Que se passera-t-il si le problème initial change



Ce chemin proposé pour relier Nantes et Lyon n'est pas le meilleur chemin possible: le trouver nécessiterait des milliards d'années de calcul. C'est une heuristique, un compromis entre la qualité du chemin et le temps de calcul alloué à sa recherche. Le meilleur chemin n'est probablement pas beaucoup plus court que celui-ci. © Google Maps

un peu ? Ajoutons une 21<sup>e</sup> ville : il y a 21 fois plus de trajets à tester qu'avec 20 villes, ce qui nécessite 21 fois plus de temps, soit 21 minutes pour trouver la solution. Ajoutons une 22<sup>e</sup> ville : le super-ordinateur tourne pendant 7 heures. Une 23<sup>e</sup> ville ? Le calcul dépasse 7 jours. Cette machine imaginaire, et déjà totalement hors de portée, se trouve ainsi mise à mal avec seulement quelques items supplémentaires. Toutefois, la situation n'est pas désespérée car il existe plusieurs méthodes pour répondre à ces défis, mais elles ne calculent pas la solution exacte. On propose une solution qui paraît convenable, on l'améliore plusieurs fois, et quand on pense que c'est trop difficile de faire mieux alors on en reste là en estimant que ce n'est déjà pas si mal. En informatique une telle approche, donnant un résultat satisfaisant mais non optimal, est appelée une « heuristique ». Cela revient un peu à appliquer l'adage qui veut que « le mieux est l'ennemi du bien ». Savoir reconnaître ces problèmes pour lesquels on ne peut pas calculer de solution en un temps raisonnable est un aspect majeur de l'algorithmique. Si un programme est très long à s'exécuter, ce n'est peut-être pas à cause d'un bug ou d'une mauvaise connaissance du langage. C'est peut-être simplement parce que l'algorithme est correct, tout comme le programme, mais... qu'il met beaucoup de temps à s'exécuter.

### Qu'est-ce que l'intelligence artificielle ?

Le terme « intelligence artificielle » (IA) est utilisé pour caractériser des algorithmes et des machines dotées de formes d' « intelligence ». Il désigne aussi le domaine de recherche qui étudie ces algorithmes, et qui a été créé dans les années 1950. Les machines étudiées en IA sont très diverses, tant sur le plan de leurs mécanismes, de leurs fonctionnalités et du but de leurs concepteurs.

Des exemples de tels algorithmes sont ceux qui permettent à des ordinateurs de battre les meilleurs joueurs du monde aux échecs ou d'aider des mathématiciens à trouver automatiquement la preuve de certains théorèmes. Ces algorithmes reposent sur la manipulation astucieuse de symboles, et le but de leurs concepteurs est de construire des machines efficaces capables de nous aider dans des tâches quotidiennes comme la recherche d'information sur Internet ou la résolution de problèmes de logistique. Bien qu'ils soient très performants sur des tâches particulières, ces algorithmes fonctionnent de manière très différente du cerveau humain, et ne sont pas capables de s'adapter de manière autonome à des tâches nouvelles (c'est-à-dire sans l'aide d'un ingénieur qui les reprogrammerait).

D'autres travaux visent au contraire à construire des algorithmes qui modélisent les processus cognitifs et neuronaux humains, afin de contribuer à mieux les comprendre. Certains laboratoires modélisent par exemple les processus d'apprentissage et de développement des savoir-faire sensoriels et moteurs de l'enfant, ou l'acquisition du langage, souvent en les expérimentant sur des robots. Cependant, les mécanismes qui sont mis en œuvre ont encore des capacités cognitives et d'adaptation qui sont très éloignées de celles de jeunes enfants, et des défis fondamentaux de compréhension de la cognition restent à résoudre.

Enfin, on peut mentionner depuis quelques années l'importance croissante du big data (masses énormes de données) dans le développement de nouvelles formes d'intelligence artificielle. Des chercheurs ont, par exemple, fait visualiser des milliers d'images de chats à un programme qui s'est alors construit, seul, le « concept » de chat. La traduction automatique s'inscrit également dans ce champ : l'algorithme de traduction de Google, par exemple, apprend au fur et à mesure qu'il réalise des traductions.

Bien entendu, cela ne signifie pas que l'on peut se passer de programmeurs expérimentés qui connaissent bien le fonctionnement de leurs langages préférés, ni que les progrès technologiques sont sans importance. Beaucoup seront satisfaits de voir les performances de leur logiciel préféré améliorées par une programmation judicieuse ou un processeur plus rapide. Mais les progrès de l'algorithmique sont indispensables à une meilleure connaissance de la nature des problèmes que l'on aborde et de la manière dont on peut les résoudre... ou pas!

## Les objets informatiques : ordinateurs, robots, réseaux...

Comme nous l'avons évoqué précédemment, l'informatique est née lorsqu'on a pu concevoir des machines capables d'exécuter des algorithmes. Ces machines « universelles » sont appelées aujourd'hui des ordinateurs.

### Qu'est-ce qu'un ordinateur ?

Un ordinateur est une machine capable de manipuler automatiquement des informations et de les mémoriser, et que l'on peut programmer. Les manipulations de l'information sont notamment des suites d'opérations arithmétiques et logiques. Bien que le mot « ordinateur » soit souvent associé à l'image de nos ordinateurs personnels, avec écran et clavier, les ordinateurs au sens large sont aujourd'hui présents dans de très nombreux objets quotidiens sous la forme de processeurs « embarqués » dans ces objets<sup>6</sup> : téléphones, tablettes, montres électroniques, voitures, avions, feux de circulation, caméras, appareils photo, télévisions, appareils ménagers, jouets interactifs, appareils auditifs, thermostats... et tous les robots dont nous parlerons plus loin.

### Des ordinateurs mécaniques aux microprocesseurs

Aujourd'hui, la plupart des ordinateurs sont des machines électroniques; cependant, les premiers ordinateurs ont d'abord été mécaniques. Précurseur, Charles Babbage a conceptualisé le premier ordinateur au début du XIX<sup>e</sup> siècle, sous la forme d'une machine composée d'un jeu complexe de rouages mécaniques (voir ci-avant, page 14). On fournissait à cette machine des instructions de calcul sous la forme de cartes perforées, selon un principe similaire à la programmation des machines à tisser. La machine de Babbage était en particulier capable de réaliser des boucles et des branchements conditionnels, c'est-à-dire de décider de la suite d'instructions à réaliser en fonction du résultat d'un calcul.

6. Il y a environ 10 fois plus d'ordinateurs « embarqués » que d'ordinateurs « classiques » (avec écran, clavier...) dans le monde.

Les premiers ordinateurs électromécaniques sont apparus pendant la Seconde Guerre mondiale. Ils étaient utilisés pour les calculs balistiques des missiles. Ces ordinateurs utilisaient l'électricité pour activer et désactiver des relais mécaniques afin de réaliser des calculs. Les ordinateurs entièrement électroniques sont apparus peu après, basés sur l'utilisation de lampes à vide: citons le Colossus en Grande-Bretagne qui servit à décoder les messages secrets échangés par l'armée allemande (voir ci-avant, page 15, à propos d'Alan Turing). Les lampes à vide furent ensuite remplacées par les transistors, dont la petite taille, la faible consommation d'énergie et la robustesse permirent leur accumulation sur des « circuits intégrés ». La miniaturisation constante des circuits intégrés a donné naissance aux microprocesseurs dès la fin des années 1950.



Lampe à vide de 1947



Transistor PNP de 1953



Microprocesseur Intel© 4040 de 1974. Contient 3 000 transistors



Microprocesseur Intel© Core i7 de 2014. Contient 2,6 milliards de transistors

Aujourd'hui, des laboratoires de recherche imaginent des alternatives aux circuits électroniques pour les ordinateurs de demain. En particulier, ils étudient comment utiliser des propriétés de la matière afin d'accélérer certains types de calculs (comme les états quantiques dans le cas des « ordinateurs quantiques »), ou comment utiliser certaines molécules complexes (comme l'ADN dans le cas du « DNA computing »).

## Les composants des ordinateurs

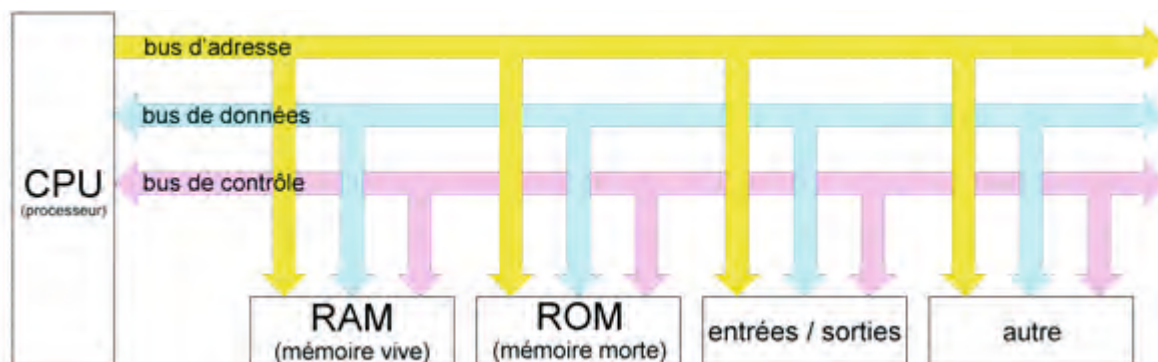
La plupart des ordinateurs actuels reposent sur quatre entités: l'unité de contrôle, l'unité arithmétique et logique (ALU), la mémoire, et les dispositifs d'entrée-sortie. Ces entités, souvent constituées elles-mêmes de plusieurs composants, sont interconnectées via des conducteurs électriques (pistes en cuivre, câbles) que l'on appelle de façon générique des « bus ». Ces bus transmettent des informations suivant un codage précis, le « protocole de communication ».

L'unité de contrôle est le chef d'orchestre qui dirige l'ensemble des autres entités: elle lit et décode les instructions et les données de la mémoire, les transforme en signaux qui vont activer d'autres entités (par exemple envoyer à l'ALU un calcul à effectuer, récupérer le résultat et le ranger à un emplacement particulier dans la mémoire).

On appelle souvent « CPU » (*central processing unit* en anglais) l'ensemble composé de l'unité de contrôle, de l'ALU et de cellules mémoire spéciales, les « registres », utilisées pour mémoriser l'emplacement en mémoire de la prochaine instruction.

La mémoire d'un ordinateur peut être vue comme un ensemble de cases dans lesquelles on peut stocker et écrire des nombres (encodés selon le codage binaire: un circuit élémentaire allumé code la valeur « 1 », et un circuit élémentaire éteint code la valeur « 0 »). Chaque case a une adresse qui permet aux autres composants de la localiser.





Les ordinateurs sont souvent associés à des dispositifs d'entrée-sortie, les « périphériques », qui leur permettent d'échanger des informations avec l'extérieur. Au-delà des claviers, souris, disques durs, écrans, haut-parleurs ou imprimantes, les périphériques de la plupart des ordinateurs embarqués dans notre quotidien sont des capteurs (par exemple de lumière, de mouvement, de chaleur, de signal GPS...), des actionneurs (par exemple des moteurs, des valves, des radiateurs,...), et surtout d'autres ordinateurs, situés à l'intérieur du même objet, ou à l'extérieur. Par exemple, un ordinateur personnel ou un téléphone contiennent plusieurs ordinateurs dédiés au traitement du son ou de l'image et qui communiquent les uns avec les autres, ainsi qu'avec les autres ordinateurs, externes, connectés au réseau Internet.

## Les robots

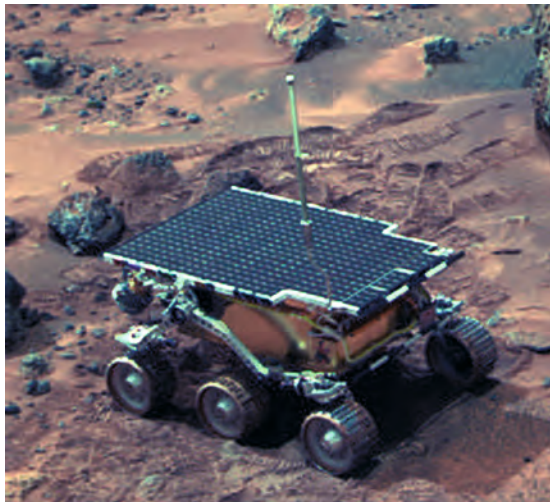
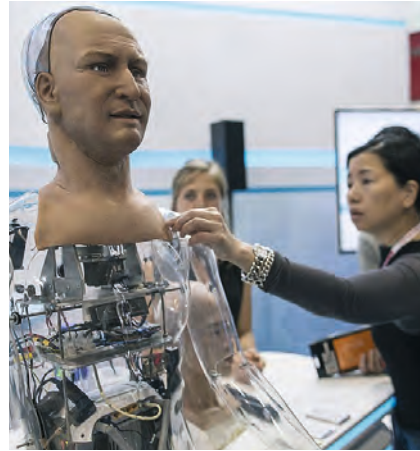
Parmi les objets qui embarquent des ordinateurs, les « robots » ont une importance scientifique, sociétale et économique grandissante. Les robots sont partout : dans les usines et dans les champs, au fond des mers et dans l'espace, dans les jardins et les salons. En outre, ils ont pénétré notre culture et certains d'entre eux participent au renouvellement de la vision que nous avons de nous-mêmes.

D'un point de vue technique, un robot est une machine dotée de capteurs (de contact, de distance, de couleur, de force...) qui lui permettent de percevoir son environnement, de moteurs l'autorisant à bouger et à agir sur cet environnement, et d'un système qui contrôle ce qu'effectue le robot en fonction de ce qu'il perçoit. Une caractéristique fondamentale des robots, qui les distingue des automates, est cette rétroaction entre perception et action. Les automates (tels que ceux de Jacques de Vaucanson ou de Pierre et Henri-Louis Jaquet-Droz au XVIII<sup>e</sup> siècle) ne sont pas des robots car leurs mouvements ne dépendent pas de ce qui se passe autour d'eux : ils n'ont pas de capteurs et leurs enchaînements sont entièrement prédéterminés par le programme.

En pratique, cette définition d'un robot recouvre une très vaste diversité de machines : les bras articulés programmables des usines automobiles, les voitures (en mode d'assistance à la conduite) et les avions qui sont aujourd'hui largement automatisés, les aspirateurs qui font le ménage tout seuls, certains objets électroniques des magasins de jouets ou les robots biomimétiques en forme d'animaux (singes, poissons...) que l'on rencontre parfois dans les laboratoires de recherche.

Cette diversité n'est pas seulement une diversité de formes, c'est aussi une diversité d'usages, qui sous-tendent une diversité de logiques de fonctionnement. Ainsi, les robots se spécialisent selon deux axes majeurs : l'autonomie et les capacités d'adaptation et/ou d'apprentissage.

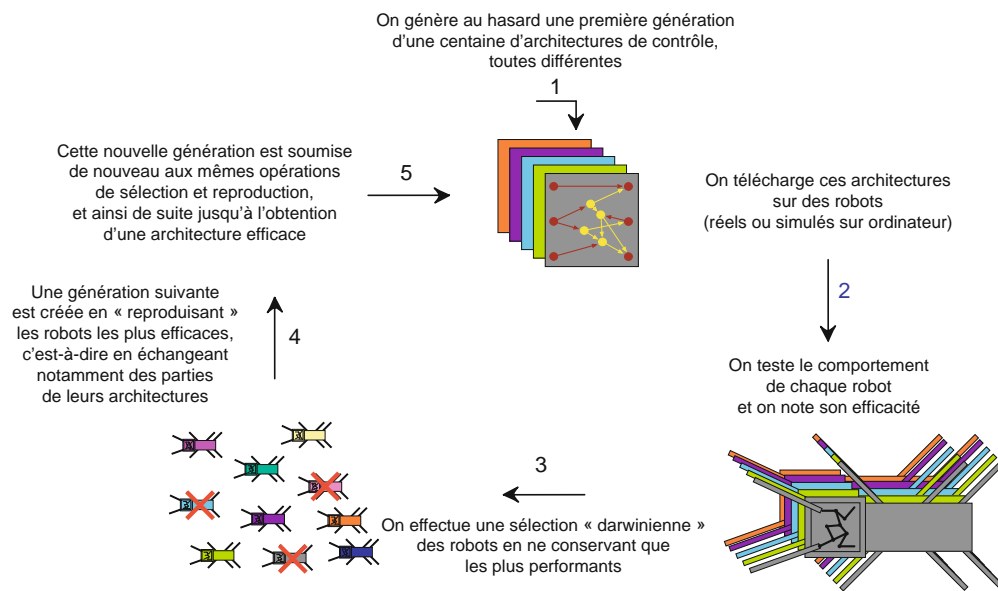
**L'autonomie** : il existe des robots qui agissent sans qu'un humain les guide, et d'autres dont le comportement est soit influencé par un humain, soit presque totalement contrôlé par un humain. Par



Quelques exemples de robots. De gauche à droite et de haut en bas: bras mécaniques utilisés dans l'industrie automobile; robot humanoïde dédié à la reconnaissance et à la reproduction des émotions; robot explorateur sur Mars et robot militaire destiné à transporter des charges lourdes sur des terrains accidentés

exemple, dans une usine, les robots qui travaillent à la chaîne et répètent toujours le même geste le font souvent de manière autonome. En revanche, les robots utilisés dans les centrales nucléaires (pour opérer dans les zones à forte radiation) sont typiquement téléguidés par un humain qui leur dit où aller et quoi faire après chaque action.

**L'adaptation et l'apprentissage:** le comportement de certains robots est figé au départ et une fois pour toutes par le programmeur, alors que d'autres robots sont capables d'acquérir de nouveaux comportements et de nouvelles connaissances par leurs expériences: leur comportement évolue en fonction de l'histoire de leurs interactions avec l'environnement. Ainsi, certains robots sont capables d'apprendre à reconnaître des objets dans des images ou encore d'apprendre à marcher en expérimentant et en évaluant eux-mêmes différentes stratégies. Ces mécanismes d'adaptation sont permis par des «algorithmes d'apprentissage», qui reposent sur la détection automatique de régularités dans des flux de données captées par le robot, et sur des méthodes d'«optimisation» permettant de raffiner de manière progressive et itérative les paramètres des solutions à un problème.



Exemple de processus itératif permettant à des «générations» de robots d'apprendre à réaliser une tâche par eux-mêmes (par exemple, se déplacer)

Grâce à ces algorithmes d'apprentissage, certains robots sont capables d'inventer des solutions et des comportements non prédits par leur concepteur, et même de sélectionner pour eux-mêmes des objectifs qui ne sont pas préprogrammés. Il est par exemple possible de programmer un robot en lui donnant des instructions lui demandant de rechercher des situations nouvelles afin d'augmenter ses connaissances sur le monde qui l'entoure. Ainsi, certains algorithmes permettent de doter ces machines de formes d'apprentissage et de créativité. Néanmoins, les capacités et les performances de ces algorithmes sont aujourd'hui, et encore pour très longtemps probablement, très faibles en comparaison des capacités d'adaptation et de raisonnement de nombreux animaux, et en particulier de l'humain.

Comme nous l'avons vu, ces différentes logiques de fonctionnement répondent à différents besoins : il existe en effet une grande variété de raisons, et donc de fonctions, pour lesquelles les robots sont construits et utilisés. On peut considérer trois familles de fonctions : *travailler et explorer, assister l'humain, modéliser les mécanismes cognitifs et comportementaux du vivant.*

**Travailler et explorer.** La plupart des robots en service dans le monde sont industriels (on en compte aujourd'hui environ 9 millions). Très tôt, les entreprises se sont intéressées à ces machines pour deux raisons :

- D'abord, les robots peuvent être utilisés pour remplacer les travailleurs humains dans les tâches répétitives, pénibles et nécessitant de faibles compétences, comme les montages, la peinture ou les soudures de pièces ;
- En outre, ces machines sont capables de réaliser des travaux à la chaîne bien plus rapidement et efficacement que les humains.

Le premier robot industriel, Unimate, est apparu en 1961 : installé dans une usine automobile de General Motors, ce bras articulé manipulait de lourdes pièces de fonderie. Dans les années 1970, l'usage des robots dans l'industrie a pris son envol. Aujourd'hui, les robots ont pénétré toutes les branches de l'industrie et ne sont plus restreints au seul secteur de l'automobile. Par exemple, dans l'agriculture et l'agro-alimentaire, les robots vont dans les champs cueillir fruits et légumes, certains coupent, pressent et mettent en bouteille ; d'autres trient et mettent en cartons ; d'autres encore groupent en palettes. Dans certains aéroports, des flottes de robots s'occupent de transporter les bagages et de les charger dans les soutes.

Les robots ne sont pas seulement utiles dans l'industrie pour les tâches simples et répétitives, ils sont aussi utilisés pour travailler dans des environnements dangereux pour l'homme. L'industrie nucléaire est un exemple typique. Qu'ils soient autonomes ou partiellement téléguidés, les robots des centrales nucléaires peuvent déambuler dans les enceintes confinées et radioactives, ils peuvent manipuler les substances dangereuses et s'occuper de la maintenance des autres machines. Un autre exemple est l'industrie pétrolière : les robots sous-marins sont par exemple utilisés pour contrôler l'état de la coque des navires afin de prévenir les accidents et d'identifier les « navires poubelles ».

Les robots sont enfin cruciaux pour l'exploration des endroits où l'homme ne peut pas aller, au premier rang desquels l'espace et les corps du Système solaire. C'est en 1966 que le premier robot mobile arrive sur la Lune, embarqué dans la sonde *Surveyor*. Suivront le soviétique *Lunokhod*, puis toute la série des américains *Mariner*. En 1997, un robot atterrit sur la planète Mars : *Sojourner*, qui est propulsé par l'énergie qu'il capte grâce à ses panneaux solaires ; il envoie à la Terre des milliers de clichés et provoque un engouement du grand public. *Sojourner* navigue en partie de manière autonome car, étant donné la distance avec la Terre, il est très difficile de le téléguider en temps réel. En 2004, une nouvelle mission robotisée concentre l'attention du monde entier : *Spirit* et *Opportunity*, équipés de spectromètres et d'un bras qui leur permet de creuser la surface, apportent la preuve que de l'eau a coulé sur Mars.

**Assister l'humain dans son quotidien.** Si le  $xx^e$  siècle a vu l'avènement des robots travailleurs et explorateurs, à l'aube du  $xxi^e$  siècle, une autre grande famille de robots prend son essor : les robots d'assistance à la personne. Dans nos maisons, les robots d'assistance aux travaux ménagers se démocratisent. On retrouve aussi des assistants robotisés dans les magasins ou sur nos lieux de travail. L'activité médicale est particulièrement transformée par la robotique : si les robots d'assistance chirurgicale sont quotidiennement utilisés depuis une quinzaine d'années, on voit émerger aujourd'hui de nouveaux usages. Ainsi, des robots accompagnent les personnes qui ont des difficultés physiques ou cognitives, par exemple pour les aider à se lever et à s'asseoir, pour les stimuler cognitivement quand elles ont des problèmes de mémoire, ou pour jouer le rôle de facilitateur des contacts avec la famille ou l'entourage médical. Demain, des chercheurs introduiront d'autres robots, beaucoup plus petits, dans les blocs opératoires : ce sont des capsules endoscopiques miniaturisées capables d'explorer conduits intestinaux, artères ou veines pour aider le chirurgien dans son diagnostic. Apparaissent aussi depuis quelques années des prothèses robotisées de mains ou de bras entières destinées aux amputés.



Prothèse robotisée ©BeBionic

**Modéliser le vivant et la cognition.** Les robots sont devenus essentiels comme outils pour penser et modéliser les systèmes complexes dans le domaine du vivant. En particulier, les robots sont aujourd'hui utilisés dans des laboratoires de recherche pour comprendre comment les êtres vivants s'adaptent à leur environnement naturel, en termes de comportement et de déploiement de capacités cognitives. En effet, ce comportement est le résultat de la dynamique des interactions entre un cerveau, un corps physique et un environnement, dynamique qui se redéfinit en permanence car le cerveau est modifié à chaque nouvelle interaction. Les robots embarquent à la fois un « cerveau » (des programmes qui permettent de traiter l'information acquise par le robot, selon des règles précises), des « organes sensoriels » (les senseurs) et des « systèmes moteurs » (les actuateurs). De cette manière, les robots peuvent, comme les êtres vivants, modifier l'environnement physique dans lequel ils agissent et être modifiés par lui. Leur





Le robot humanoïde Poppy est *open source*. Il est développé par l'Inria à la fois pour des recherches sur la cognition et pour des projets éducatifs. Plus de détails sur : <https://www.poppy-project.org>

« cerveau » aussi en est modifié, car le robot acquiert de nouvelles connaissances qu'il peut réutiliser dans ses interactions ultérieures.

Les chercheurs peuvent étudier la complexité de l'interaction cerveau-corps-environnement, grâce à des expérimentations possibles sur des robots et impossibles sur des êtres vivants – comme par exemple « éteindre » une partie de leur cerveau artificiel pour voir comment le comportement est modifié ou altérer des parties du corps. Certains laboratoires étudient ainsi le contrôle moteur, la perception visuelle, le repérage dans l'espace, ou encore les mécanismes de l'apprentissage et de l'évolution de la parole et du langage chez l'humain. Dans le cadre de ces projets, les interactions avec les neurosciences, la biologie, la psychologie ou encore l'éthologie jouent un rôle central.

## Les réseaux

De nos jours, la plupart des ordinateurs sont interconnectés entre eux, et rares sont ceux coupés du réseau. On les connecte pour leur permettre d'échanger des informations, qu'il s'agisse de transmettre un courriel ou de télécharger des données stockées sur les ordinateurs de Wikipédia pour les lire.

On interconnecte aussi les ordinateurs pour leur permettre de collaborer à une tâche trop volumineuse pour être réalisée sur une seule machine. De nos jours, des « fermes » d'ordinateurs regroupent des milliers voire des millions de machines – qui, prises individuellement, ont des performances assez classiques – dans le même hangar climatisé, pour faire toujours plus de calculs à la seconde.

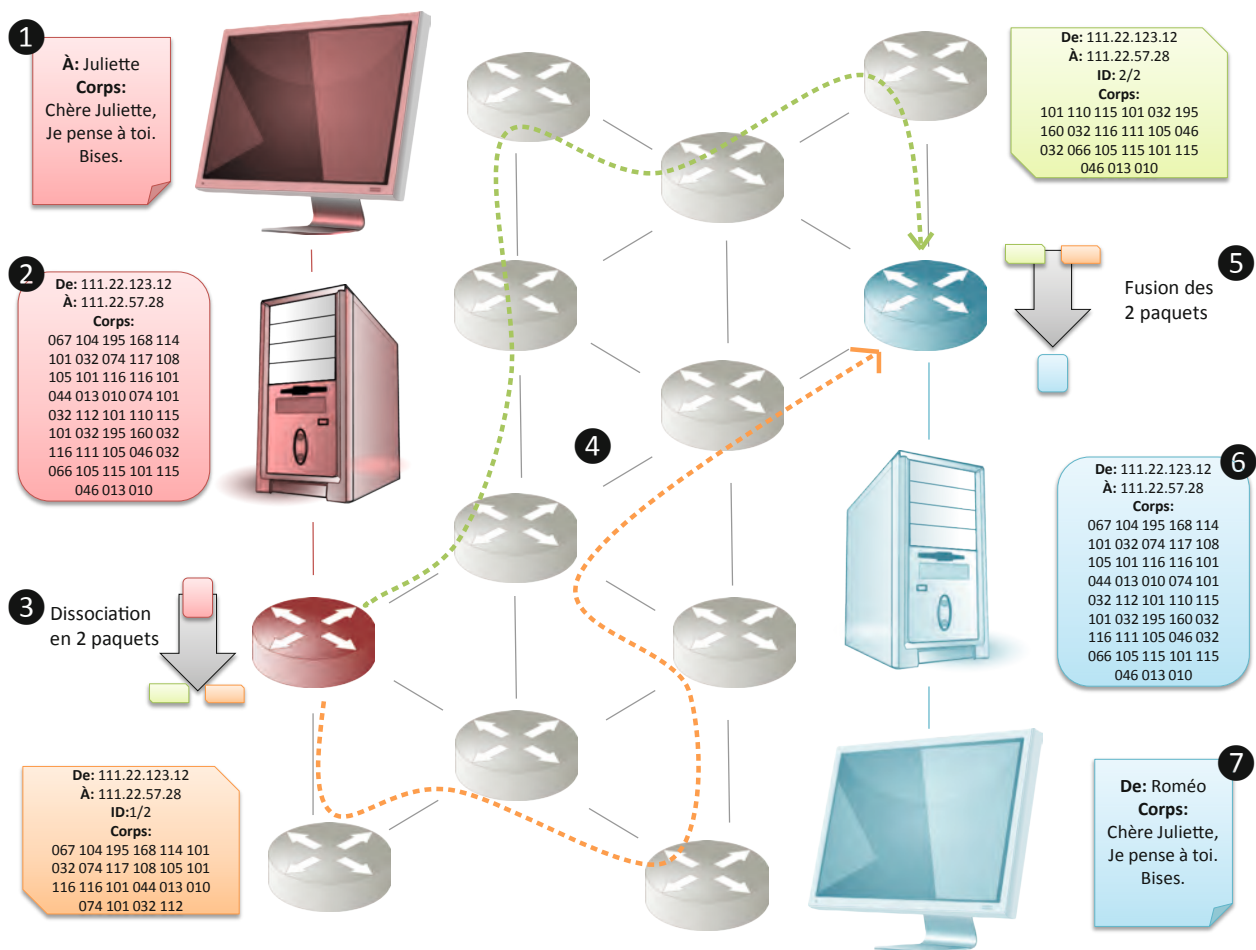
Les premiers réseaux à grande distance sont apparus à la fin des années 1960. On ne comptait qu'une quinzaine de machines interconnectées en 1971, un millier en 1984, un million en 1992, et elles sont plusieurs milliards aujourd'hui. Le premier e-mail date de 1972, le premier site Web de 1991, le premier tweet de 2006.

Dans les anciens réseaux téléphoniques, il fallait connecter physiquement deux téléphones de bout en bout pour qu'ils puissent communiquer. Des opérateurs dans des centraux téléphoniques réalisaient ces branchements, qui demandaient d'avoir autant de lignes que de discussions en cours.

La plupart des réseaux informatiques actuels suivent une autre approche plus efficace, nommée « commutation de paquets ». Les informations sont coupées en paquets, qui sont envoyés les uns derrière les autres. À chaque nœud du réseau (qui est maillé), des « routeurs » reçoivent ces paquets et les font suivre au suivant, jusqu'au destinataire.

Plusieurs technologies ont été proposées pour échanger ces paquets entre ordinateurs, au risque de faire ressembler les réseaux informatiques à la tour de Babel. Heureusement, Internet permet à tous les ordinateurs qu'il connecte de communiquer malgré leurs différences. Pour parvenir à ce tour de force, les scientifiques et ingénieurs ont appliqué une approche classique en informatique : le problème est découpé en sous-problèmes, qui sont résolus séparément, puis les réponses partielles sont combinées. Ainsi, le problème consistant à échanger des informations entre des ordinateurs est découpé en couches.

C'est ce découpage qui rend possible un réseau à l'échelle mondiale : les technologies sont interchangeables au sein de chaque couche, et l'ensemble collabore grâce à des interfaces standardisées entre les couches. Internet Protocol (IP) est le nom d'un protocole permettant d'interconnecter n'importe quels réseaux locaux entre eux. C'est pour cela qu'on dit qu'Internet est le « réseau des réseaux ».



Roméo utilise son logiciel de messagerie pour envoyer un e-mail à Juliette (1). Ce message est d'abord encodé en ASCII (2) puis en binaire. Un routeur le découpe en plusieurs paquets (3: dans la réalité, il y a bien plus de paquets!). Chaque paquet transite sur le réseau via plusieurs autres routeurs (4). Les paquets sont finalement fusionnés (5), puis le message est transmis à l'ordinateur de Juliette (6), avant d'être décodé et affiché à l'écran (7)

## Internet ou Web ?

Dans le langage courant, et très souvent dans les médias, il n'existe pas de différence entre ces deux termes. Erreur !

Internet est le « réseau des réseaux », et le Web n'est que l'un des réseaux qu'Internet relie (voir page 19 pour l'historique).

Le World Wide Web, inventé en 1990 au CERN, à Genève, est un réseau de pages liées entre elles exclusivement par liens hypertextes (on reconnaît facilement leur nature car leur URL – *Uniform Resource Locator* –, leur adresse, commence souvent par « www. ». Mais il existe d'autres réseaux sur Internet, comme par exemple :

- les services de courrier électronique (les e-mails), qui utilisent les protocoles SMTP (*Simple Mail Transfer Protocol*), POP3 (*Post Office Protocol version 3*), IMAP (*Internet Message Access Protocol*)... ;
- les services de messagerie instantanée (les chats), sur protocole IRC (*Internet Relay Chat*), XMPP (*Extensible Messaging and Presence Protocol*)... ;
- les serveurs de forum Usenet (newsgroups), sur protocole NNTP (*Network News Transfer Protocol*)...

# Représentation de l'information<sup>7</sup>

L'idée de représenter un objet, concret ou abstrait, par un autre est très ancienne, puisque toutes les langues reposent sur une telle représentation des objets par des sons.

Comme les objets sont trop nombreux pour être chacun représenté par un son différent, ils sont représentés, non par un son unique, mais par une suite de sons, choisis dans un ensemble de petite taille. Par exemple, le français classique utilise trente-six sons, ou phonèmes, pour former tous les mots. Dans une langue, un message est donc une suite finie d'unités choisies dans un ensemble fini. Cette idée n'est cependant pas complètement aboutie dans les langues orales où le volume, le débit, l'intonation du message contribuent à sa signification, tout autant que la suite des unités dont il est constitué.

## Combien d'information contient un texte ?

L'écriture, et surtout son utilisation pour exprimer les langues, élimine ces éléments accessoires, mais elle constitue, au moins dans un premier temps, une régression, puisque les premières écritures des langues associaient un symbole différent à chaque objet. Ce n'est qu'avec l'alphabet que se retrouve l'idée d'associer à chaque objet non une unité, mais une suite d'unités prises dans un ensemble de petite taille : l'alphabet. L'alphabet du français, par exemple, contient 110 caractères environ : 42 minuscules

7. Le projet « 1, 2, 3... codez ! » propose plusieurs activités aux cycles 2 et 3 pour travailler la représentation de l'information (texte et image), notamment les séances situées aux pages 121, 129 (cycle 2) et 221, 227, 233, 302, 310 et 315 (cycle 3).

(les 26 lettres fondamentales, 13 voyelles accentuées, le c cédille et les 2 ligatures æ et œ), autant de majuscules, 10 chiffres, et une vingtaine de signes de ponctuation.

Le nombre de suites de  $n$  caractères que l'on peut former en les choisissant dans un alphabet de  $k$  éléments est  $k^n$ : en effet, il y a  $k$  possibilités pour le premier caractère,  $k$  possibilités pour le deuxième, ... et  $k$  possibilité pour le  $n$ -ième, donc le nombre total de possibilités est  $k * k * \dots * k$  ( $n$  fois) c'est-à-dire  $k^n$ . Plus ce nombre est grand, plus chacune de ces suites contient d'information. Par exemple, un texte de 4000 caractères choisis parmi les 110 caractères de l'alphabet français contient plus d'information qu'un texte de deux caractères choisis parmi trois.

L'unité d'information est appelée le « bit ».

Pour des raisons historiques on utilise souvent une autre unité d'information, l'octet qui vaut huit bits, ainsi que ses multiples: le kilooctet (ko) qui vaut mille octets, soit huit mille bits, le mégaoctet (Mo) qui vaut un million d'octets, le gigaoctet (Go) qui vaut un milliard d'octets, le téraoctet (To) qui vaut mille milliards d'octets, et le pétaoctet (Po) qui vaut un million de milliards d'octets.

La quantité d'information contenue dans un caractère de l'alphabet est de l'ordre de 1 octet. Celle contenue dans une page de 2000 caractères est d'environ 2 kilooctets, celle contenue dans un livre de six cents pages est de l'ordre de 1 mégaoctet, celle contenue dans une petite bibliothèque de mille volumes est de l'ordre de 1 gigaoctet. Celle contenue dans la collection de livres imprimés de la Bibliothèque nationale de France, qui compte 14 millions de volumes (en laissant de côté les images, les films...) est de l'ordre du téraoctet (précisément 14 téraoctets). Un téraoctet est aussi la capacité d'un disque vendu quelques dizaines d'euros. Un pétaoctet correspond à cent fois la taille de la BNF. Chaque année, le CERN (Centre européen pour la recherche nucléaire) produit de l'ordre de 15 pétaoctets de données.

## Le binaire : un alphabet à 2 lettres

### Comment représenter un nombre ?

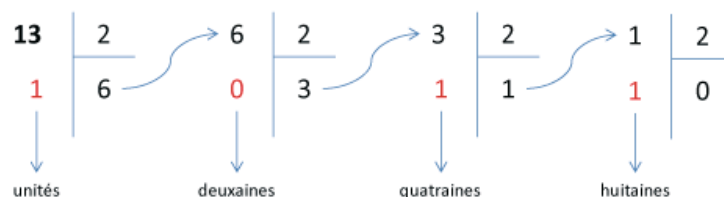
Une branche entière de l'informatique s'intéresse à la représentation des données sous la forme d'une suite de symboles pris dans un alphabet fini, souvent dans un alphabet qui ne contient que deux caractères: **0** et **1**. Ainsi, pour représenter les nombres entiers dans un alphabet qui ne contient que ces deux caractères, on peut associer une suite de quatre caractères «0 ou 1» à chaque chiffre, par exemple:

- 0: 0000
- 1: 0001
- 2: 0010
- 3: 0011
- 4: 0100
- 5: 0101
- 6: 0110
- 7: 0111
- 8: 1000
- 9: 1001

Puis, pour représenter un nombre, la première idée, intuitive, qu'on peut avoir consiste à juxtaposer la représentation de chacun de ces chiffres. Le nombre 13 par exemple se représente alors en juxtaposant la représentation 0001 du chiffre 1 avec la représentation 0011 du chiffre 3: 00010011. Cette notation, peu compacte, n'est pas utilisée en pratique pour les nombres.



Une autre représentation, plus concise, consiste à représenter le nombre en base deux. Le nombre 13 par exemple, qui se décompose en une unité, zéro paquet de deux unités, un paquet de deux paquets de deux unités et un paquet de deux paquets de deux paquets de deux unités ( $1 + 4 + 8$ ) s'écrit de droite à gauche: 1-0-1-1, soit 1101 de gauche à droite.



Pour écrire le nombre 13 en base 2, on doit le décomposer comme une somme de puissances de 2. Pour cela, il faut effectuer une série de divisions par 2, jusqu'à obtenir un quotient égal à 0. Les restes des divisions successives (qui sont des 0 et des 1) correspondent aux chiffres des unités, deuxaines, etc. Pour écrire 13 en binaire, il faut remettre les chiffres dans l'ordre (avec les unités à droite).

Une succession de 64 caractères «0 ou 1» permet alors de représenter les nombres de 0 à  $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$ . La question de la représentation des nombres ne s'arrête cependant pas là, car les nombres négatifs et les nombres décimaux ont leur propre mode de représentation.

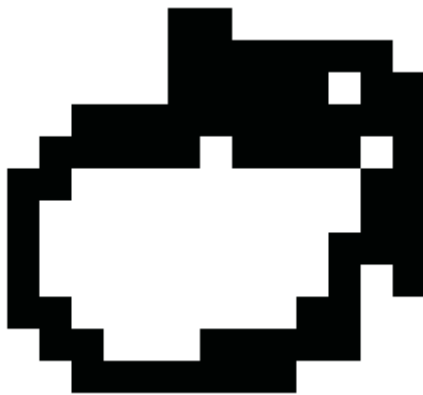
## Comment représenter un texte ?

La représentation des caractères textuels – et des textes qui sont des successions de caractères textuels – est aussi une question très riche. Une première idée pour représenter les caractères textuels était d'associer un groupe de 7 caractères «0 ou 1» à chaque caractère d'un alphabet qui en contenait 95 (26 minuscules, 26 majuscules, 10 chiffres et 33 autres symboles, incluant les signes de ponctuation). Le caractère textuel «a» était représenté par 1100001, «b» par le 1100010, etc. Et un mot était simplement représenté par la juxtaposition de la représentation de ses caractères – ses lettres. Le mot «Le» par exemple était représenté par la juxtaposition de la représentation 1001100 du caractère «L» et la représentation 1100101 du caractère «e»: 10011001100101.

Mais cette représentation, appelée American Standard Code for Information Interchange (ASCII), était adaptée à l'anglais et non aux langues qui, comme le français, utilisent des signes diacritiques: accents, cédilles,... La représentation ASCII a donc été étendue plusieurs fois pour arriver à la représentation Unicode, qui permet de représenter les quelque 110 000 caractères utilisés par les différentes écritures du monde, y compris les écritures non alphabétiques, comme le chinois.

## Comment représenter une image ?

La représentation des images est plus difficile et elle repose sur un découpage de l'image en pixels, puis par un codage de la teinte de chaque pixel. Dans une image en noir en blanc, chaque pixel est représenté par un seul «0 ou 1»: 1 s'il est noir, 0 s'il est blanc.



```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0
0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0
0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

À gauche, une image en noir et blanc de 16 x 16 pixels. À droite, le codage binaire de cette image. Pour faciliter la lecture, on a disposé les bits sur une grille 16 x 16.

Dans une image en niveaux de gris chaque pixel est classiquement représenté par un nombre compris entre 0 et 255, lui-même représenté en base deux par un groupe de huit «0 ou 1», 0 = 00000000 s'il est noir, 1 = 00000001 s'il est gris très foncé... 254 = 11111110 s'il est gris très clair, 255 = 11111111 s'il est blanc. Bien entendu, le choix d'utiliser 255 niveaux de gris est arbitraire et d'autres formats existent. Dans une image en couleurs, chaque pixel est classiquement représenté par trois nombres, chacun compris entre 0 et 255 qui indiquent la quantité de rouge, de vert et de bleu dans ce pixel. Par exemple un pixel rose bonbon est obtenu en additionnant 249 unités de rouge, 66 unités de vert et 158 unités de bleu; le nombre 249 se représente en base deux par 11111001, le nombre 66 par 01000010 et le nombre 158 par 10011110, si bien qu'un pixel rose bonbon se représente par la succession de 24 «0 ou 1»: 111110010100001010011110. Une image se représente naturellement en juxtaposant les représentations de chacun de ses pixels. Si bien qu'une image formée de deux pixels rose bonbon se représente par: 11111001010000101001111011110010100001010011110.

## Organiser l'information

Ces principes permettent de représenter de petits objets: un nombre, un texte, une image, un son, une vidéo... On peut, par exemple, les utiliser pour stocker sur le disque d'un ordinateur un seul texte, une seule image, un seul son ou une seule vidéo. Mais, bien souvent, on veut stocker sur un tel disque plusieurs objets de cette nature. Cela mène à d'autres principes qui permettent non de représenter, mais d'organiser l'information.

Un disque est un espace sur lequel il est possible de stocker un grand nombre de «0 ou 1», typiquement huit mille milliards pour un disque d'un téraoctet. Pour pouvoir y stocker plusieurs textes, images,... on commence par découper cet espace en plusieurs zones appelées «fichiers» qui permettent chacune de stocker un texte, une image, un son, une vidéo... Mais dès que ces fichiers sont nombreux il est nécessaire de les regrouper en dossiers, eux-mêmes regroupés en dossiers..., c'est-à-dire de les structurer en un arbre. Toutefois, retrouver un fichier dans un tel arbre devient vite difficile, si bien que l'on a inventé d'autres principes pour organiser l'information à grande échelle. Le premier est l'utilisation de liens hypertextes.

Un tel lien permet d'indiquer, dans un premier fichier, le nom et la position d'un second fichier situé sur le même disque, ou sur un autre.

Ainsi, on peut indiquer dans le premier fichier que le second se trouve à l'adresse «disque3: archives/2016/lettre.txt», c'est-à-dire qu'il porte le nom «lettre.txt» qu'il est situé dans le dossier «2016», lui-même situé dans le dossier «archives», sur un disque qui porte le nom «disque3». De cette manière, en utilisant un navigateur pour lire le premier fichier, il suffira de cliquer sur ce lien pour accéder au second. Cette manière d'organiser l'information à l'aide de liens hypertextes, à travers un réseau de disques et d'ordinateurs, est l'idée du Web (voir plus haut, page 35).

Cette organisation de l'information en fichiers, puis en dossiers, et son enrichissement par des liens hypertextes est elle-même insuffisante quand on veut stocker beaucoup d'information, et il existe deux autres manières d'organiser de grandes quantités d'information. La première, qui est adaptée à l'information structurée, est la constitution d'une base de données. Par exemple, un carnet d'adresses est une petite base de données, formée de quintuplets (5 données) constitués chacun d'un nom, d'un prénom, d'une adresse, d'un numéro de téléphone et d'une adresse électronique. Il est possible d'interroger cette base de données en formulant une requête, dans laquelle on demande la liste des quintuplets qui contiennent par exemple un nom, ou alors un prénom, voire un numéro de téléphone. Pour les données moins structurées, on peut laisser en vrac un ensemble de fichiers sur un ou plusieurs disques et laisser un moteur de recherche les indexer, puis accéder à ces fichiers en formulant une requête au moteur de recherche, qui indiquera alors où se trouvent tous les fichiers qui contiennent, par exemple, un certain mot. Cette manière d'organiser l'information demande moins de travail que leur structuration dans une base de données, mais donne des résultats de moins grande qualité. Chercher le numéro de téléphone de Mme Marchand en formulant une requête dans un moteur de recherche risque d'indiquer que le mot «Marchand» est présent dans de nombreux textes, dont beaucoup ne parleront que du métier de marchand.

## Transformer et manipuler l'information<sup>8</sup>

Représenter et organiser ainsi les informations permettent de les transmettre d'un lieu à un autre (c'est-à-dire de les faire voyager dans l'espace), de les archiver pour les restituer plus tard (c'est-à-dire de les faire voyager dans le temps), mais surtout de les transformer. De nombreux algorithmes permettent de transformer des données particulières; par exemple l'algorithme de la multiplication ne s'applique qu'à des nombres. Mais trois types d'algorithmes s'appliquent à des données quelconques: les algorithmes de compression, de correction et de chiffrement.

### Compression

Nous avons vu qu'un pixel rose bonbon se représente par 111110010100001010011110 et qu'une image monochrome d'un million de pixels de cette même couleur s'obtient en juxtaposant 1 million de fois ce bloc de 24 «0 ou 1». Au lieu de produire un texte très répétitif de vingt-quatre millions de signes, on peut utiliser une représentation plus concise qui consiste à indiquer que le bloc 111110010100001010011110 doit être répété un million de fois. C'est ce que l'on appelle une compression.

8. Le projet «1, 2, 3... codez!» propose deux activités pour aborder la correction et le chiffrement au cycle 3 (pages 322 et 330).

## Correction

Les algorithmes de correction d'erreurs permettent de repérer, et parfois de corriger, une erreur dans une information, par exemple une erreur introduite lors de la transmission de cette information. Une méthode simple, mais coûteuse en espace mémoire, est de répéter trois fois chaque information unitaire. Par exemple 101 est transformé en 111000111. Si une erreur se produit, par exemple dans le troisième triplet: 111000110, il est facile non seulement de repérer cette erreur, mais aussi de la corriger: comme il y a une majorité de 1 dans 110, la lettre initiale était sans doute un 1. Cet algorithme de correction est mis en échec uniquement quand deux erreurs se produisent dans le même triplet.

## Chiffrement

Enfin les algorithmes de chiffrement permettent de rendre une information inintelligible, sauf aux personnes qui possèdent la clef de chiffrement. Une méthode simple, mais peu efficace, était déjà utilisée par Jules César pour communiquer avec ses armées. Elle consiste à décaler chaque lettre d'un nombre fixe de places dans l'alphabet. Ainsi en décalant les lettres de trois places dans la phrase «VENI VIDI VICI» on obtient «YGPL YLPL YLEL». Pour déchiffrer ce message il faut connaître la clé, c'est-à-dire le nombre de places dont il est nécessaire de décaler chaque lettre pour retrouver le message initial. Comme il n'y a que vingt-trois lettres en latin, il n'y a que vingt-trois clefs possibles, si bien que ce message n'est pas très difficile à casser (ou à «cryptanalyser»), c'est-à-dire à déchiffrer sans connaître au préalable la clé.

L'analyse fréquentielle permet de trouver rapidement la clé: dans une langue donnée, toutes les lettres ont une fréquence d'apparition connue (en français, le E – dans toutes ses déclinaisons – est de loin la lettre la plus fréquente; les lettres A, I, N, O, R, S, T, L sont ensuite assez fréquentes... et à l'autre bout du spectre les lettres Z, W et K sont très peu utilisées). Si, la plupart du temps, la lettre la plus fréquente donne directement la clef d'un chiffrement de César, il peut être nécessaire d'utiliser un histogramme plus complet, qui fournit les fréquences de toutes les lettres. Il suffit, pour connaître la clé, de décaler l'histogramme du texte chiffré jusqu'à le faire correspondre au mieux avec l'histogramme de la langue considérée. Dans les 3 exemples ci-dessous, la clef de chiffrement est toujours la même (+3, la même que celle utilisée par Jules César): même si l'on constate de petits écarts dans les histogrammes, on retrouve un même motif (même pour le texte issu du roman *La Disparition* de G. Percec qui a été écrit sans le moindre E).

Message chiffré (clef+3)	Analyse fréquentielle	Message déchiffré
MH MHWWH DYHF JUDFH PRQ IHXWUH MH IDLV OHQWHPHQW O DEDQGRQ GX JUDQG PDQWHDX TXL PH FDOIHXWUH HW MH WLUH PRQ HVSDGRQ		Je jette avec grâce mon feutre, Je fais lentement l'abandon Du grand manteau qui me calfeutre, Et je tire mon espadon (Cyrano de Bergerac, E. Rostand)



<p>O HWUH KXPDLQ FURLUD  WRXMRXUV TXH SOXV  OH URERW SDUDLW  KXPDLQ SOXV LO HWV  DYDQFH FRPSOHAH HW  LQWHOOLJHQW</p>		<p>L'être humain croira toujours que plus le robot paraît humain, plus il est avancé, complexe et intelligent.  <i>(Les robots de l'aube, I. Asimov)</i></p>
<p>LO DEDQGRQQD VRQ URPDQ  VXU VRQ OLV LO DOOD D  VRQ ODYDER LO PRXLOOD  XQ JDQW TX LO SDVVD VXU  VRQ IURQW VXU VRQ FRX</p>		<p>Il abandonna son roman sur son lit. Il alla à son lavabo; il mouilla un gant qu'il passa sur son front, sur son cou.  <i>(La Disparition, G. Perec)</i></p>

Le chiffrement de César fait partie de la famille des chiffrements « par substitution monoalphabétique » : Il existe d'autres méthodes bien plus complexes : par exemple, pendant la Seconde Guerre mondiale, le chiffrement allemand Enigma (cassé par A. Turing et son équipe, voir page 15) se basait lui aussi sur des substitutions, mais la clef de chiffrement changeait régulièrement au fil du texte ! Historiquement, des méthodes de chiffrement qui n'étaient pas des substitutions monoalphabétiques ont aussi été utilisées, notamment en temps de guerre (par exemple, le chiffrement ADFGVX durant la Seconde Guerre mondiale).

Des méthodes de chiffrement bien meilleures existent et sont utilisées tous les jours pour protéger nos numéros de cartes de crédit, et autres secrets que nous souhaitons ne partager qu'avec un petit nombre de personnes...

# Signal vs information

Du latin *signum* («signe convenu»), le signal est un porteur (ou vecteur) d'information. L'information est portée par les fluctuations d'une grandeur (physique, le plus souvent), générées par un phénomène (naturel ou technologique).

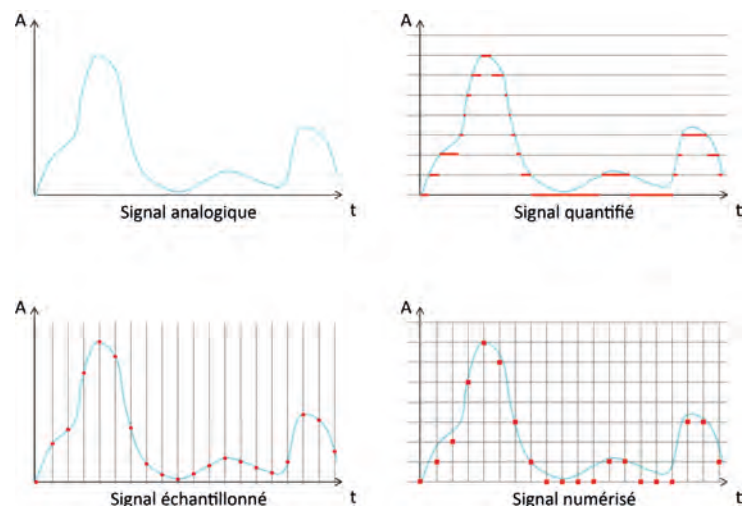
On peut classer les signaux de nombreuses façons, par exemple :

Type de signal	Support physique
<p><b>Signal analogique</b> : vecteur physique à évolution continue, en amplitude et dans le temps;</p> <p><b>Signal numérique</b> : vecteur physique discrétisé (voir plus loin);</p> <p><b>Signal logique</b> : vecteur qui ne porte qu'une information binaire, «vrai» ou «faux»;</p> <p><b>Signal de synchronisation</b> : signal dont l'information n'est pas portée par une fluctuation d'amplitude, mais par la chronologie des fluctuations (comme les bips d'un métronome).</p>	<p><b>Signal sonore</b> : le vecteur est une onde de compression (sonnerie, tambour...);</p> <p><b>Signal lumineux</b> : le vecteur est une onde électromagnétique dont peuvent varier la longueur d'onde (la couleur), la fréquence d'émission, l'amplitude... (phare, étoile, télégraphie sans fil, feu de circulation...);</p> <p><b>Signal optique</b> : ici, le vecteur est, comme pour le signal lumineux, une onde électromagnétique, cependant l'information est portée par la répartition spatiale du flux lumineux; comme dans une image, où seul l'agencement des pixels porte l'information et non chaque pixel individuellement (écran vidéo, signaux de fumée, télégraphe de Chappe...);</p> <p><b>Signal électrique</b> : le vecteur transite par un matériau électrique, électronique ou magnétique (ordinateur, disque dur, neurone...);</p> <p><b>Signal (bio)chimique</b> : le vecteur est une concentration de molécules chimiques naturelles ou artificielles (hormones...).</p>

Pour interpréter correctement un signal (et donc décoder l'information qu'il porte), émetteur et récepteur doivent préalablement choisir une convention commune<sup>9</sup>. Mais malgré cette évidente connivence, il faut traiter le signal pour l'utiliser. L'appellation «traitement du signal» est cependant assez vague, car elle recoupe de nombreux aspects :

- La **discrétisation** d'un signal analogique pour faciliter sa manipulation (en particulier sa numérisation sur un support digital). La discrétisation suppose à la fois de découper le signal en un nombre fini de valeurs à intervalles réguliers (de temps ou d'espace), puis de représenter chaque valeur de manière approchée pour que toutes se trouvent dans un ensemble fini de valeurs possibles. Enfin, on encode ces valeurs, par exemple en binaire, pour les représenter dans la machine.

On différencie parfois la discrétisation spatiale (qu'on appelle «pixellisation»),



Étapes de discrétisation (en rouge) d'un signal analogique (en bleu).

9. Parfois, l'émetteur est un système naturel (exemple : une étoile) et ne peut donc pas «choisir» une convention avec le récepteur, humain. C'est alors le travail du scientifique de modéliser le fonctionnement de cette étoile pour comprendre quel état physique est associé à quel signal émis.

la discrétisation d'amplitude (« quantification ») ou temporelle (« échantillonnage »). Cependant, il est courant d'employer, par abus de langage, le terme « échantillonnage » pour ces trois aspects: dans tous les cas, on prélève périodiquement des échantillons du signal.

- La **compression** du signal pour son stockage et sa transmission (plus le message est long, plus il y a de risques d'introduire des erreurs lors de l'émission, de la transmission, de la réception, du stockage, ou de l'interprétation du signal). Les méthodes de compression peuvent s'effectuer soit avec, soit sans perte: l'usage final dictera laquelle choisir. Le choix de la compression avec ou sans perte dépend bien de l'usage final, mais aussi du type d'information (sur une image ce peut être acceptable, pas sur du texte) et de l'espace de stockage/temps de transmission disponible. C'est donc un compromis.
- L'**amélioration** de sa qualité: on cherche souvent à séparer l'information (pertinente) du bruit (parasite) inhérent à tout phénomène physique. Pourtant, il n'existe pas de définition absolue du bruit: par exemple, le bruit de fond détecté par Penzias et Wilson en 1964 qui perturbait les échanges radio du laboratoire Bell s'est avéré être la signature du Big Bang recherchée par les astronomes (le fameux « fond diffus cosmologique », dont la découverte leur a valu le prix Nobel). On caractérise souvent la « qualité » d'un signal par un nombre appelé le « rapport signal/bruit ».
- L'**interprétation**: il s'agit ici d'extraire l'information pertinente au regard de l'usage que l'on veut en faire: par exemple, sur un feu de signalisation, c'est la couleur (rouge, vert, orange) qui nous importe, et pas l'intensité du flux lumineux émis par la lampe.

## Théories de l'information (Shannon vs Kolmogorov)

La numérisation de l'information a engendré de nouvelles disciplines, portées en particulier par Shannon et Kolmogorov.

Claude Elwood Shannon (dont nous avons, en 2016, fêté le centenaire de la naissance) est un ingénieur en génie électrique et un mathématicien américain. En parallèle de ses recherches au *Massachusetts Institute of Technology* (MIT), il travaille aux laboratoires Bell où il décrit la communication entre machines, introduisant le schéma désormais classique « source → encodeur → signal → décodeur → destinataire ». En 1948, Shannon (avec l'aide de son collègue Weaver) initie la « théorie mathématique de la communication de l'information » (TMCI, que la postérité a simplifiée en « théorie de l'information »). Il décrit précisément ce qui, dans un signal, tient de l'information, de la redondance (le surplus) et du bruit (les intrus), afin d'améliorer la communication. L'idée est de limiter la redondance pour que, malgré le bruit, l'information soit préservée, tout en ayant un message le plus court possible.

La notion de redondance peut s'illustrer facilement en linguistique: dans une phrase, le sens vient de l'agencement et du choix des mots, pas seulement des lettres et des syllabes qui les constituent... En effet, « ctt phrs rst prftmnt cmprhnsbl » (cette phrase reste parfaitement compréhensible): en limitant les redondances, on diminue la taille du message.

Le bruit est une dégradation du signal imposée par les contraintes physiques inhérentes à la communication: l'information est au mieux brouillée, au pire perdue (dans ce cas, les redondances peuvent contrer l'impact du bruit en offrant des pistes intéressantes pour la correction des erreurs). À ce titre, Shannon a démontré que, pour s'assurer de la qualité de la numérisation, il faut que l'échantillonnage adopte un « pas » deux fois plus petit que le motif le plus petit de l'information.

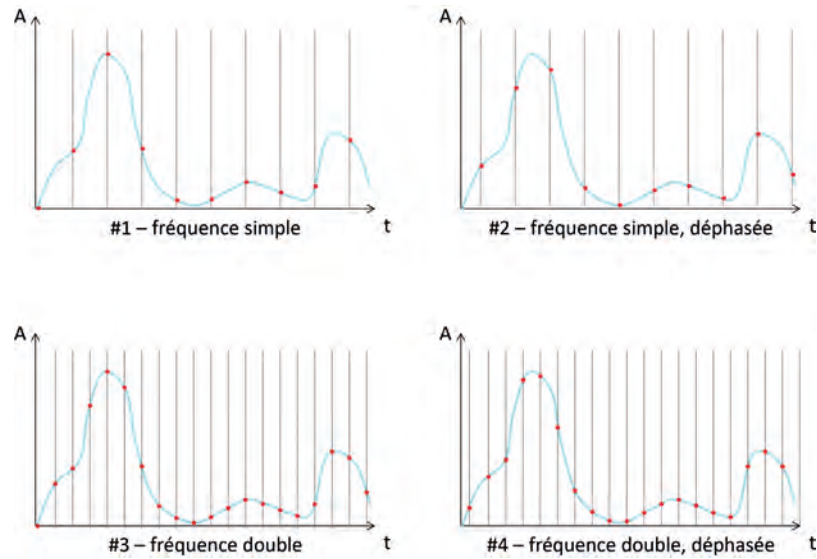


Illustration du théorème de Shannon: avec un échantillonnage de la même taille que les motifs à reproduire (#1 et #2), il est possible de sous-estimer grandement les pics selon le centrage des échantillons (l'épaulement initial manque en #1, le pic principal est tronqué en #2); par contre, avec une fréquence d'échantillonnage double (#3 et #4), quel que soit le centrage des échantillons, tous les pics sont toujours bien représentés.

Le théorème de Shannon peut également être utilisé pour distinguer bruit et information. En prenant l'exemple d'une image à l'écran, si un pixel montre une information différente de celle des pixels environnants, c'est probablement un bruit; si deux pixels contigus ont le même comportement dans un environnement distinct, alors il s'agit probablement d'un vrai motif.

Dans les années 1960, le mathématicien russe Andreï Nikolaïevitch Kolmogorov (aidé de ses collègues Solomonov et Chaitin) propose une approche différente de la TMCI de Shannon: la « théorie algorithmique de l'information ». Cette théorie vise à quantifier et qualifier le contenu en information d'un ensemble de données, sa « complexité ». Prenons par exemple ces quatre descriptions du blason provençal ci-contre :

- 1 - Un blason jaune et rouge
- 2 - D'or aux quatre pals de gueules
- 3 - Un blason jaune de 90 cm de largeur et 1 m de hauteur avec une rayure rouge verticale de 10 cm à 10 cm du bord gauche, une autre à 10 cm à droite de la précédente, encore une autre 10 cm à droite, et une dernière 10 cm à droite.
- 4 - Un blason jaune de 90 cm de largeur et 1 m de hauteur avec quatre rayures rouges verticales de 10 cm espacées de 10 cm les unes des autres.



La description #1 semble la plus courte de toutes (la moins complexe), mais elle est ambiguë (trop de blasons différents correspondent à ce descriptif). La description #2 est réservée aux connaisseurs : seuls les héraldistes savent l'interpréter « correctement » ; à vrai dire, ils ne peuvent peut-être pas reproduire à l'identique ce blason, mais l'information pertinente pour eux réside dans le nombre, la couleur et l'agencement des figures, pas dans leur taille précise au centimètre près.

La description #3 est très précise, mais très longue : elle est assurément plus complexe que la description #1, et reproduit fidèlement le motif initial. La description #4 reproduit un motif identique au cas #3, mais avec moins de mots : l'information en est aussi complexe, donc ; inversement, la seule longueur de



la description ne suffit pas à décréter si l'objet en question est complexe ou pas. On retrouve par contre dans cet exemple #4 l'origine du nom de cette théorie : on a recherché l'algorithme le plus simple qui permette de reproduire à l'identique le motif initial. Et c'est donc en comparant la complexité de ces différents algorithmes que l'on peut commencer à classer la complexité des informations.

# Informatique et enjeux sociétaux

## Enjeux actuels, scientifiques, économiques et sociétaux

La construction d'ordinateurs depuis les années 1940 a d'abord transformé le monde de la recherche : en permettant de simuler des phénomènes physiques complexes – telles les évolutions du climat – et en permettant de traiter de grandes masses de données, comme la séquence du génome humain, les ordinateurs ont transformé la méthode scientifique. Les ordinateurs et les réseaux ont parallèlement transformé la circulation des informations dans les entreprises. Enfin, l'arrivée des ordinateurs et des réseaux dans tous les foyers a transformé la manière dont nous communiquons dans notre vie privée, avec nos « amis » et nos familles.

On qualifie parfois de « monde numérique » le monde dans lequel nous vivons, qui se caractérise par l'omniprésence des objets informatiques. Ces objets sont partout, de manière visible : ordinateurs, tablettes, téléphones... et invisible. Dans l'informatique « embarquée » (dans les avions, les trains, les voitures, les régulateurs cardiaques, les appareils électroménagers...) des objets informatiques remplacent en effet de plus en plus de systèmes mécaniques ou électriques.

L'accès aux données, grâce à Internet, se fait désormais de façon extrêmement simple et rapide. Une donnée, dès lors qu'elle est numérisée, occupe une place négligeable, pour un coût minime, et peut être dupliquée instantanément. De ce fait, les systèmes informatiques sont dits « hypermnésiques » : ils sont capables de collecter des quantités énormes d'information et de les conserver sur de longues durées. Un exercice simple permet de prendre la mesure de cette hypermnésie : la Bibliothèque nationale de France contient 14 millions de volumes. Supposons que chaque volume comporte cinq cents pages et chaque page deux mille signes (oublions, pour cet exercice de pensée, les images) et qu'un signe s'exprime sur un octet. La « taille » de cette bibliothèque est donc de  $14\,000\,000 \times 500 \times 2000 = 14\,000\,000\,000\,000$  octets soit quatorze mille milliards d'octets, 14 téraoctets. Or un disque de 10 téraoctets coûte quelques centaines d'euros. On voit donc que, grâce aux techniques numériques, la capacité de stockage d'une petite entreprise peut rapidement dépasser la taille de la Bibliothèque nationale de France.

D'un point de vue économique, ce monde favorise bien évidemment l'industrie informatique, qui produit tous ces matériels et logiciels à haute valeur ajoutée, ainsi que la recherche en science informatique, dont les avancées théoriques vont de pair avec les avancées technologiques. Le monde numérique est destructeur d'emploi : acheminer le courrier d'un bout à l'autre de la Terre demandait des armées de facteurs, qui ne sont plus nécessaires aujourd'hui que le courrier est électronique. Mais il est aussi source d'emploi, car il faut sans cesse développer de nouvelles applications et de nouveaux logiciels pour suivre l'évolution rapide des besoins. Enfin, troisième pilier de l'importance économique

de l'informatique, auquel on ne pense pas forcément de prime abord : la formation professionnelle. Tous ceux qui manipulent, de près ou de loin, les ordinateurs et les logiciels (scientifiques, techniciens, commerciaux, juristes, enseignants, architectes, artistes...) ont besoin de formation, initiale ou continue. La formation professionnelle et le conseil en informatique représentent une part de plus en plus importante de l'activité liée au numérique.

Malgré ces opportunités, force est de constater que la France et l'Europe accusent un retard vis-à-vis de l'Amérique du Nord et de l'Asie. À l'exception de quelques niches (services informatiques, recherche, robotique...), nos pays ont trop longuement relégué l'informatique au rang d'outil, enseignant comment utiliser tel logiciel (logiciel qui devient souvent obsolète en quelques années) plutôt que d'enseigner la pensée informatique, étape indispensable pour favoriser l'émergence d'innovation. On assiste depuis quelques années à une prise de conscience de ce retard et à un souhait de le combler, notamment par l'éducation.

## Informatique et éthique

Le monde numérique offre une liberté sans pareille : s'exprimer, accéder à l'information, se former, communiquer, échanger... Mais là où il y a tant de libertés, les risques et les dangers sont grands. Il convient donc d'accompagner les jeunes en leur montrant les limites de ce nouveau monde.

Accéder à l'information de manière immédiate et gratuite aurait semblé une utopie aux bâtisseurs de la bibliothèque d'Alexandrie. Pourtant, comme nous l'avons vu plus haut, un disque dur de faible coût peut contenir presque autant de données que la BNF. Mais si les informations sont accessibles en quantité, sont-elles pour autant de qualité ? Quelle fiabilité donner aux informations que l'on trouve sur le Web ? Comment garder un certain esprit critique, vérifier les sources (elles aussi accessibles par Internet), croiser les témoignages... ? En classe, on peut par exemple faire réaliser une page Web par les élèves, afin qu'ils prennent conscience du fait qu'ils peuvent eux-mêmes écrire n'importe quoi sur le Web, et donc que, s'ils le peuvent, vraisemblablement d'autres le peuvent aussi. Les informations que l'on trouve sur le Web y ont été déposées par des gens comme eux.

Parmi les informations numérisées qui sont stockées ici et là, nous trouvons également nos données personnelles. Nos données de sécurité sociale sont stockées sur un serveur en France, mais nous n'y pouvons rien. Nos données bancaires sont stockées, probablement sur un serveur étranger, et nous n'y pouvons rien non plus. En revanche, nos photos de vacances sont postées sur Facebook, car nous les y avons mises nous-mêmes. Dans ces trois cas, la prudence s'impose : quel est le degré de sécurisation de nos données ? Qui peut avoir accès à cette collection astronomique d'informations personnelles ? Souvenons-nous que les ordinateurs sont hypermnésiques et que toute activité peut être enregistrée et traitée très rapidement. En 2008, l'entreprise Google a lancé le projet « Google Flu » : en accumulant, région par région, toutes les recherches des mots-clefs « grippe », « fièvre »... il est possible de décrire en temps réel la progression de l'épidémie, et même de la prévoir un peu à l'avance. De telles informations sont évidemment précieuses pour les autorités de santé, mais aussi pour les entreprises pharmaceutiques. De telles informations doivent-elles être rendues publiques, ou peuvent-elles être gardées privées et vendues aux compagnies les plus offrantes ? Ce qui est vrai pour la grippe l'est pour de nombreux autres sujets, et comme Google enregistre non seulement les requêtes mais aussi l'adresse IP de l'ordinateur émetteur il n'est plus trop difficile de transformer cette donnée d'apparence anonyme en véritable espionnage. Et que dire de toutes ces applications qui usent et abusent de la géolocalisation, permettant de nous localiser au mètre près ?

Tous ces outils, que nous utilisons pour nous simplifier la vie, recèlent des dangers insoupçonnés. Chaque pays doit donc légiférer en connaissance de cause. S'assurer que nos libertés ne sont pas compromises, mais garantir la sécurité de nos données voire un droit à l'oubli. Cela passe par la reconnaissance légale des délits numériques, la recherche en cryptologie et virologie informatique, la formation à la sécurité... Ce qui est déjà difficile à l'échelle nationale, faute de « bon sens informatique » ou de connaissances approfondies des risques, devient encore plus difficile sur la scène internationale. Les réseaux sont en effet supranationaux, les lois diffèrent d'un pays à l'autre et la vente de certains produits comme les drogues est autorisée dans certains pays et non dans d'autres. De même, la publication de certains textes – diffamatoires, blasphématoires... – est autorisée dans certains pays et non dans d'autres. Quelles lois régissent la publication d'un texte : celles du pays où est situé l'ordinateur qui héberge le texte ou celle du pays où est situé l'ordinateur qui permet d'accéder à ce texte ? L'éducation des jeunes mais aussi des adultes (citoyens comme législateurs) est nécessaire pour favoriser une utilisation éclairée et respectueuse de ces nouveaux outils.

## Les métiers de l'informatique

L'informatique et les sciences du numériques sont au cœur d'un bouleversement complet de notre économie. Dans tous les secteurs, des métiers disparaissent ou sont amenés à disparaître (dactylographe, par exemple... ou chauffeur de taxi d'ici quelques années), de nouveaux métiers émergent (*data-scientist...*), et presque tous sont profondément impactés (du boulanger à l'enseignant en passant par le médecin)<sup>10</sup>. Au-delà de cette évolution d'ensemble, quels sont les métiers spécifiques de l'informatique ? Ils sont eux-mêmes très variés et couvrent de nombreux secteurs d'activité, à tel point que le terme « informaticien », très en vogue dans les années 1980 pour désigner toute personne exerçant un métier en rapport avec l'informatique, est aujourd'hui tombé en désuétude. On lui préfère des termes plus spécialisés : analyste-programmeur, architecte de systèmes d'information, *game designer*...

L'association Pasc@line<sup>11</sup> classe les métiers du numérique selon 7 grandes catégories :

- la programmation et le développement ;
- la formation et l'assistance ;
- la conception et la gestion de projet ;
- les infrastructures et les réseaux ;
- la communication et le marketing ;
- le pilotage et le management ;
- l'expertise, l'enseignement et la recherche.

Si la plupart des recrutements actuels concernent des diplômés bac+5 (diplôme d'ingénieur, master...), de nombreux métiers sont accessibles depuis des formations de niveau bac + 2/3 (BTS, DUT, licence pro). Ci-dessous, nous proposons quelques éclairages sur certains de ces métiers, sous la forme de courts témoignages.

10. Lire, par exemple, le rapport de France Stratégie « Les métiers en 2022 » publié en avril 2015.

11. <http://www.assopascaline.fr>

## Focus : chercheur en informatique, un métier à multiples facettes

Chercheur en informatique est un terme générique qui recouvre des réalités très différentes selon que l'on travaille, par exemple, sur le big-data, les robots ou la cryptographie.

### Témoignage n° 1 : Florent Masegla, chercheur en analyse de données, Inria



Après une thèse soutenue en 2001, j'ai obtenu un poste de chercheur Inria à Sophia-Antipolis. J'y ai travaillé pendant 8 ans sur de l'analyse de données, principalement appliquée aux usages. Il s'agissait de comprendre des comportements utilisateurs pour optimiser des systèmes (par exemple un site Web) et les rendre plus adaptés à ces comportements. En 2010, j'ai déménagé pour Montpellier, toujours chez Inria, dans une équipe qui travaille avec des scientifiques de diverses disciplines. J'y fais toujours de l'analyse de données, mais au lieu d'aider les systèmes à s'adapter aux utilisateurs, j'aide les scientifiques à mieux comprendre leurs données (par exemple, pour un agronome, comment poussent ses plantes, alors qu'elles sont observées par milliers). La masse des données est telle qu'il faut distribuer les calculs sur des dizaines ou des centaines de machines et le challenge est passionnant.

J'aime beaucoup d'aspects de ce métier. D'abord, c'est un vrai travail d'équipe, où les compétences de chacun sont bien mises en œuvre. J'aime aussi le fait qu'on ne peut pas se reposer sur ses lauriers, il y aura toujours plus fort, plus malin, plus efficace... c'est stimulant. Il y a aussi le fait de me sentir utile pour la société. Enfin, j'investis beaucoup de temps dans la médiation scientifique. Pour expliquer mon domaine de recherche, ou l'informatique en général, au grand public. C'est très enrichissant d'écouter des personnes de tous horizons donner leur point de vue sur ce qu'on fait en recherche. Et puis, je suis payé avec leurs impôts alors... je leur dois bien ça !

### Témoignage n° 2 : Adeline Langlois, chercheuse en cryptographie, CNRS



Je suis chercheuse au CNRS depuis un peu plus d'un an. C'est un long parcours qui m'a permis d'avoir ce poste : après une classe préparatoire scientifique, j'ai intégré l'ENS de Rennes, où j'ai préparé une licence puis un master d'informatique, découvrant peu à peu la recherche pendant des stages. J'ai ensuite fait trois ans de doctorat à Lyon et une année de post-doctorat en Suisse.

Mon domaine de recherche est une nouvelle branche de la cryptographie qui pourrait devenir une alternative à celle que nous utilisons aujourd'hui, si, dans quelques années, celle-ci devait devenir obsolète à cause de l'avènement des ordinateurs quantiques. Je cherche donc à construire et à garantir la sécurité des échanges d'information de demain.

Il y a différents aspects que j'aime beaucoup dans ce métier, le premier est la stimulation intellectuelle due à la recherche : toujours se poser des questions, réfléchir à de nouvelles solutions... J'aime ensuite le côté collaboratif : la recherche avance bien mieux quand on travaille à plusieurs et quand on réfléchit ensemble, avec des étudiants ou d'autres chercheurs. Enfin il y a les voyages : la recherche n'a pas de frontières et permet de collaborer avec des chercheurs de toutes cultures et de tous horizons. Pendant ma thèse, j'ai eu la chance de passer 6 mois en Australie et plusieurs semaines à Singapour, mais aussi d'aller à des conférences partout dans le monde pour présenter mes travaux.

## Focus : quels sont les métiers derrière un site Web ?

Derrière un site Web se cachent de nombreux métiers bien différents : ergonomes, web-designers, intégrateurs, développeurs, administrateurs de bases de données, chargés de référencement, administrateurs réseau, responsables sécurité, chefs de projet, animateurs de communauté, webmasters...

### **Témoignage n° 3 : Olivier Pourvellarie, développeur Web indépendant**

Après 5 années d'études sur les réseaux, les langages de programmation et les bases de données, je me suis orienté vers le développement Web. Pour moi, le Web était une découverte car au début des années 2000, tout était à faire. Être un acteur de la révolution Internet est toujours aussi passionnant.

Ce qui est intéressant dans le Web, c'est la diversité des langages de programmation que l'on y rencontre, comme le PHP (langage objet), le JavaScript (procédural), le HTML (balise), le CSS (styles), le SQL (les bases de données). C'est donc un métier où l'on ne reste pas à faire tout le temps la même chose.

Le développeur Web doit savoir travailler avec des équipes de designers, rédacteurs, communicants, etc., qui ne comprennent pas forcément son métier, et réciproquement. Il doit donc être pédagogue pour expliquer ses besoins, ses contraintes et former les utilisateurs. Cette occasion de s'immerger dans des domaines nouveaux est très enrichissante sur le plan personnel.



### **Témoignage n° 4 : Edouard Fajnzilberg, directeur technique, Kernel 42**

Pour moi, tout a commencé au collège quand j'ai voulu faire mes premiers sites Web. J'ai découvert HTML, puis PHP. Au lycée, j'ai utilisé des cours en ligne et des tutos pour apprendre d'autres langages (C, C++) et la modélisation, ce qui m'a beaucoup aidé quand j'ai intégré Epitech, une école très exigeante qui délivre un diplôme bac+5 en informatique. En 3<sup>e</sup> année, on a fondé notre entreprise (Kernel 42) avec Dorian Marchand, un copain de promo. C'était musclé, car il fallait gérer l'école et la boîte en même temps. Au début, on n'avait même pas de local, et j'étais tout seul à m'occuper du développement. On a intégré une pépinière d'entreprise, et une grosse boîte a investi pour qu'on se développe. Aujourd'hui, on a des bureaux à Paris et l'entreprise compte 8 salariés en CDI. Je ne fais presque plus de développement, mais de la conception, de l'architecture (logicielle, applicative, et matérielle). En tant que directeur technique, je veille aux processus d'industrialisation et je m'occupe aussi des recrutements et de la formation de nos développeurs.

Ce que j'aime le plus dans ce métier, c'est qu'on est sans cesse confrontés à de nouveaux problèmes qu'il faut chercher à résoudre. Et, comme les technologies Web changent tous les ans, il faut rester au top : on apprend tous les jours !



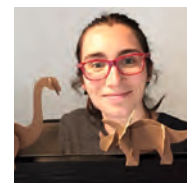


## Focus : quels sont les métiers derrière un jeu vidéo ?

Première industrie culturelle par son chiffre d'affaires en France, l'industrie du jeu vidéo regroupe des dizaines de métiers très variés, que l'on peut regrouper en quatre grandes familles :

- conception du jeu (game designer, scénariste, level designer, sound designer, chef de projet...);
- graphisme (animateur 2D/3D, graphiste, spécialiste effets spéciaux, directeur artistique...);
- programmation (développeur, testeur, directeur technique...);
- marketing (community manager, vendeur, data analyst...).

### Témoignage n° 5 : Gaëlle Oliveau, Game Designeuse, Manzalab



Venant d'une filière artistique et diplômée d'un master de Game Design en 2015, mon crédo est l'apprentissage par le jeu. J'ai décidé de me plonger dans le monde des jeux vidéo éducatifs suite à la découverte du jeu «Ludomedic» qui s'adresse à des enfants hospitalisés.

Je travaille pour des clients ou pour des projets de recherche français ou étrangers. Je conçois les «racines» des projets de jeu, en harmonisant le côté ludique et le côté éducatif, et je supervise les équipes Assurance Qualité qui permettent de garantir un contenu conforme au design attendu. Ce que j'aime dans le travail de Game Designer, c'est le fait de travailler avec des interlocuteurs différents : développeurs, graphistes, scénaristes, etc.

Cette galerie de portrait ne serait pas complète sans un témoignage lié à l'éducation : formation d'adultes, enseignement, médiation..., l'apprentissage des sciences informatiques dès le plus jeune âge mobilise de nombreux acteurs.

### Témoignage n° 6 : Romain Liblau, responsable pédagogique, Magic Makers



Après avoir passé mon adolescence à chercher des tutoriels de programmation (souvent mal écrits ou très scolaires), je me suis orienté vers une école d'ingénieur en informatique. L'autonomie et les capacités de création qu'offre la programmation m'ont toujours fasciné.

Je transmets aujourd'hui ces compétences aux enfants et adolescents qui partagent cette envie de créer et d'être maîtres de leur ordinateur. Avec les enfants, on est toujours à la recherche de projets ludiques. Ensemble, on relève les défis qu'amène la programmation de leurs jeux vidéo et le mois d'après, on s'embarque dans la construction d'un robot; en somme, on ne s'ennuie jamais! Dans ce métier, on doit rester au courant des dernières technologies et trouver comment les expliquer pour permettre aux plus jeunes de rendre réels leurs projets les plus fous. Transmettre ce pouvoir de création aux enfants est intellectuellement stimulant mais la vraie récompense est leur regard quand ils arrivent à faire marcher leur programme.

# Éclairage pédagogique

# L'informatique et les TICE, pour ne pas les confondre

## Les TICE : importantes, mais pas suffisantes

Au cours de ces 3 dernières décennies, la présence de l'informatique dans les enseignements s'est focalisée sur l'apprentissage de l'utilisation d'outils tels que l'ordinateur, la tablette et quelques logiciels courants : le traitement de texte, le tableur, le navigateur Web, etc.

Cette approche centrée sur les usages est celle des Technologies de l'information et de la communication pour l'enseignement, ou TICE. Elle présente un intérêt certain puisqu'elle permet aux enfants d'acquérir des compétences utiles dans de nombreuses disciplines scolaires ou dans la vie quotidienne, qu'il s'agisse de chercher de l'information, de communiquer, de calculer...

## Aller au-delà d'une simple utilisation : comprendre l'informatique

Cependant, l'approche centrée sur les usages ne prépare l'élève qu'à utiliser des outils standards sans comprendre les principes qui sous-tendent leur fonctionnement. L'élève ne s'en trouve ni équipé pour adapter les outils existants à ses besoins spécifiques, ni préparé à s'adapter aux évolutions rapides de ces outils, ni formé à imaginer et à créer de nouveaux outils. Ce sont pourtant ces compétences qui le rendront acteur du monde numérique qui l'entoure, plutôt que simple utilisateur.

L'informatique n'est pas qu'un ensemble d'outils, c'est une science, qui possède son histoire et ses concepts (largement racontés et détaillés dans l'éclairage scientifique, pages 9 et suivantes). En tant que science, l'informatique a toute sa place dans les enseignements, au même titre que la physique ou la biologie. Par ailleurs, les compétences travaillées en informatique, comme dans les autres sciences, concourent à développer chez l'élève l'autonomie, la prise d'initiative, le raisonnement, la créativité, etc. Les concepts de l'informatique survivront aux outils d'aujourd'hui, et leur maîtrise accompagnera durablement les élèves, tout au long de leur vie professionnelle et citoyenne. À titre d'exemple : nous jugeons plus formateur de guider les élèves dans la découverte de la façon dont sont encodées les images, au cours d'activités débranchées (pixellisation manuelle d'images) et branchées (création de petites images numériques) que de se contenter de les entraîner à manipuler des images numériques à l'aide de logiciels sophistiqués utilisés comme des boîtes noires. Ces logiciels seront bien mieux utilisés si les élèves ont compris les notions sous-jacentes comme le codage de l'information, le pixel, la résolution, la compression...

Il en est de même pour les notions propres à l'algorithmique, à la programmation ou à la robotique. Leur maîtrise donne du sens à l'usage : un élève ainsi initié à ces concepts ne verra plus jamais du même œil son logiciel de traitement d'images, son tableur et tout autre logiciel, ou même un robot ménager. En un mot, aborder l'informatique en tant que science n'exclut pas les TICE, mais se limiter aux TICE nous semble une vision très réductrice de l'informatique.

# Comment utiliser ce guide pédagogique ?

## Un tome 2 pour aller plus loin

En juin 2016 est paru le guide pédagogique *1, 2, 3... codez! – Enseigner l’informatique à l’école et au collège*. Ce guide (tome 1) propose une progression pédagogique complète en sciences informatiques pour les cycles d’enseignement 1 à 3, c’est-à-dire de la maternelle à la classe de 6<sup>e</sup>. Il permet une première approche des grands concepts informatiques, et il est tout à fait accessible à des élèves débutants dans ce domaine. La progression prévue pour le cycle 3 est tout à fait adaptée aux élèves de cycle 4 qui débutent en science informatique.

En publiant le présent volume *1, 2, 3... codez! – tome 2: 6 projets d’informatique pour le collège*, nous anticipons l’élévation du niveau de science informatique que va progressivement engendrer l’introduction depuis la rentrée 2016 de cette discipline dans les programmes d’enseignement du premier et du second degrés. Dans les années à venir, une proportion croissante des élèves de collège aura déjà vécu un projet de science informatique, et sera en mesure – et en demande – d’aller plus loin. Aussi ce tome 2 propose-t-il 6 projets d’informatique indépendants les uns des autres, dont certains sont accessibles à des élèves collégiens quasiment débutants, et d’autres réservés à des élèves plus aguerris. La diversité de ces projets permet une programmation à l’échelle d’un cycle: un projet tiré du tome 1 peut servir, en classe de 6<sup>e</sup>, à introduire les concepts de base de l’informatique, tandis que des projets du tome 2 peuvent permettre un renforcement de ces concepts et un approfondissement, en classes de 5<sup>e</sup>, 4<sup>e</sup> et 3<sup>e</sup>, dans le cadre du cours de mathématiques, du cours de technologie ou à l’occasion d’un EPI.

## Six projets clefs en main pour le collège

Les 6 projets pour le collège décrits dans ce guide peuvent être considérés comme « clefs en main »: en effet, leur description est suffisamment détaillée pour permettre à un enseignant, même peu familiarisé avec les sciences informatiques, de s’auto-former avant de conduire son enseignement. Pour chaque séance, on précise la durée de l’activité, le matériel nécessaire, le questionnement initial, les notions abordées, les difficultés potentielles, la conclusion visée...

Les projets ont été conçus par une équipe pluridisciplinaire composée d’enseignants, de formateurs et de scientifiques, puis testés par 16 enseignants dans 13 collèges, auprès d’un total de 23 classes. Parmi les 6 projets proposés dans cet ouvrage, certains sont conçus pour être menés par un professeur de mathématiques, d’autres par un professeur de technologie, d’autres encore ont une forte composante interdisciplinaire et supposent l’implication de plusieurs enseignants (par exemple dans le cadre des EPI). Certains projets sont accessibles à des élèves quasiment débutants en sciences informatiques, alors que d’autres sont réservés à des élèves plus aguerris.

Le premier travail de l'utilisateur de ce guide est donc d'identifier le ou les projets pertinents pour les classes avec lesquelles il souhaite travailler, et de choisir le projet le plus adapté. Ce choix est facilité par :

- le tableau synthétique des pages 66-67, qui offre une vue d'ensemble des 6 projets;
- le résumé des séances et le scénario conceptuel proposés en pages d'ouverture de chacun des projets.

Encore une fois, dans le cas d'élèves tout à fait débutants, ne pas oublier d'envisager de choisir la séquence Cycle 3 décrite dans le tome I.

Une fois le projet choisi, il est essentiel que le professeur se l'approprie, ce qui suppose qu'il réalise lui-même, à l'avance, les activités proposées. Il est alors en mesure d'adapter le projet, en fonction du temps qu'il souhaite y consacrer et de ses objectifs pédagogiques.

Les projets sont décrits d'une façon à la fois très détaillée et relativement linéaire, qui facilite cette appropriation par le professeur. Mais lors de leur mise en œuvre concrète dans la classe, il se peut que les élèves partent dans une direction différente de celle prévue. Le professeur doit donc, le cas échéant, adapter la progression aux propositions des élèves.

## En plus du guide pédagogique, des possibilités d'accompagnement

Afin d'accompagner l'introduction de l'informatique à l'école et au collège, la Fondation *La main à la pâte* et ses partenaires proposent différents services aux enseignants et aux formateurs :

- Formations en présentiel :
  - L'équipe de *La main à la pâte* anime des formations à la demande des autorités éducatives locales (circonscriptions, rectorats, ESPE...). Se renseigner sur le site Internet du projet (cf. page 404).
  - Le réseau des Maisons pour la science propose également des formations dans les régions où les Maisons sont implantées (<http://www.maisons-pour-la-science.org/>).
- Formation à distance et hybride
  - Le consortium Class'Code, qui rassemble de nombreux acteurs institutionnels et associatifs, propose une formation hybride associant un MOOC et un accompagnement en présentiel sur l'enseignement de l'informatique (<http://classcode.fr>).
  - France IOI propose, sur son site Internet (<http://www.france-ioi.org/>) de nombreuses ressources, mais aussi des formations en ligne et un accompagnement à distance.
- Accompagnement à distance
  - Le site Internet « 1,2,3... codez! » (cf. page 404) se veut un outil d'accompagnement des enseignants, qui peuvent poser des questions, témoigner de leurs pratiques en classe, etc.



# Comment enseigner l'informatique ?

## Quels contenus enseigner ?

L'enseignement de l'informatique est souvent compris comme celui de la programmation. Comme nous allons le voir, la programmation est essentielle, mais loin d'être suffisante dans un tel enseignement :

- Il est indispensable de transmettre aux élèves les rudiments de la programmation assez tôt. Tout d'abord, parce qu'il est presque impossible de comprendre ce qu'est un programme sans en écrire quelques-uns soi-même. Ensuite, parce que donner aux élèves une maîtrise minimale de la programmation est essentiel pour leur permettre d'avoir une démarche d'apprentissage active. Un élève qui ne sait pas programmer et qui cherche à comprendre, par exemple, le code de César, se retrouve rapidement dans une situation de passivité : il peut, peut-être, comprendre les principes de cette méthode de chiffrement, il peut comprendre que d'autres personnes l'utilisent pour chiffrer et déchiffrer des messages, mais il ne peut pas le faire lui-même. En revanche, dès qu'il comprend les rudiments d'un langage de programmation, il peut écrire lui-même un programme de quelques lignes, qui permet de chiffrer et de déchiffrer des messages, et s'approprier cette méthode par une démarche active.

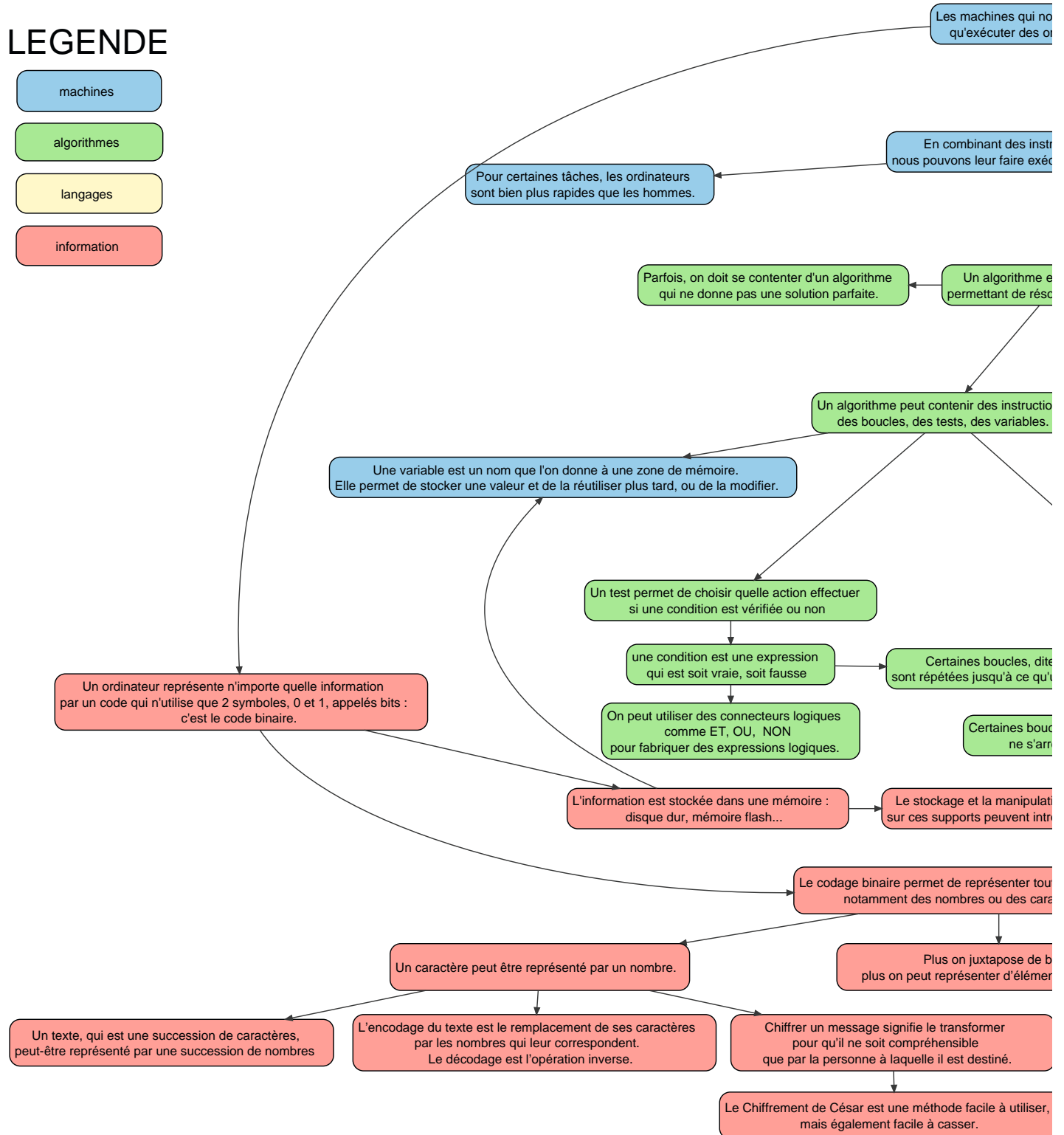
- Toutefois, connaître l'informatique ne se limite pas à savoir programmer, de même que connaître l'électricité ne se limite pas à savoir réaliser un circuit électrique. Il est donc essentiel dans la conception d'un enseignement de l'informatique d'équilibrer différents objectifs :

- que les élèves comprennent des algorithmes et les principes qui permettent de concevoir des algorithmes, par exemple le principe de dichotomie (au collège),
- que les élèves comprennent ce qu'est un langage de programmation, en quoi il se distingue d'une langue naturelle, en quoi il se rapproche de la notation musicale, du langage qui permet d'exprimer les nombres avec des chiffres arabes ou romains, etc.
- que les élèves comprennent que les objets informatiques traitent des données représentées de manière symbolique (textes, images, sons...) et comment ces données peuvent être compressées, chiffrées, corrigées, etc.
- que les élèves comprennent que les machines à traiter de l'information sont diverses : ordinateurs, téléphones, robots, réseaux... mais que cette diversité cache une profonde unité : toutes ces machines traitent de l'information en exécutant des algorithmes, exprimés sous forme de programmes, dans un langage de programmation.

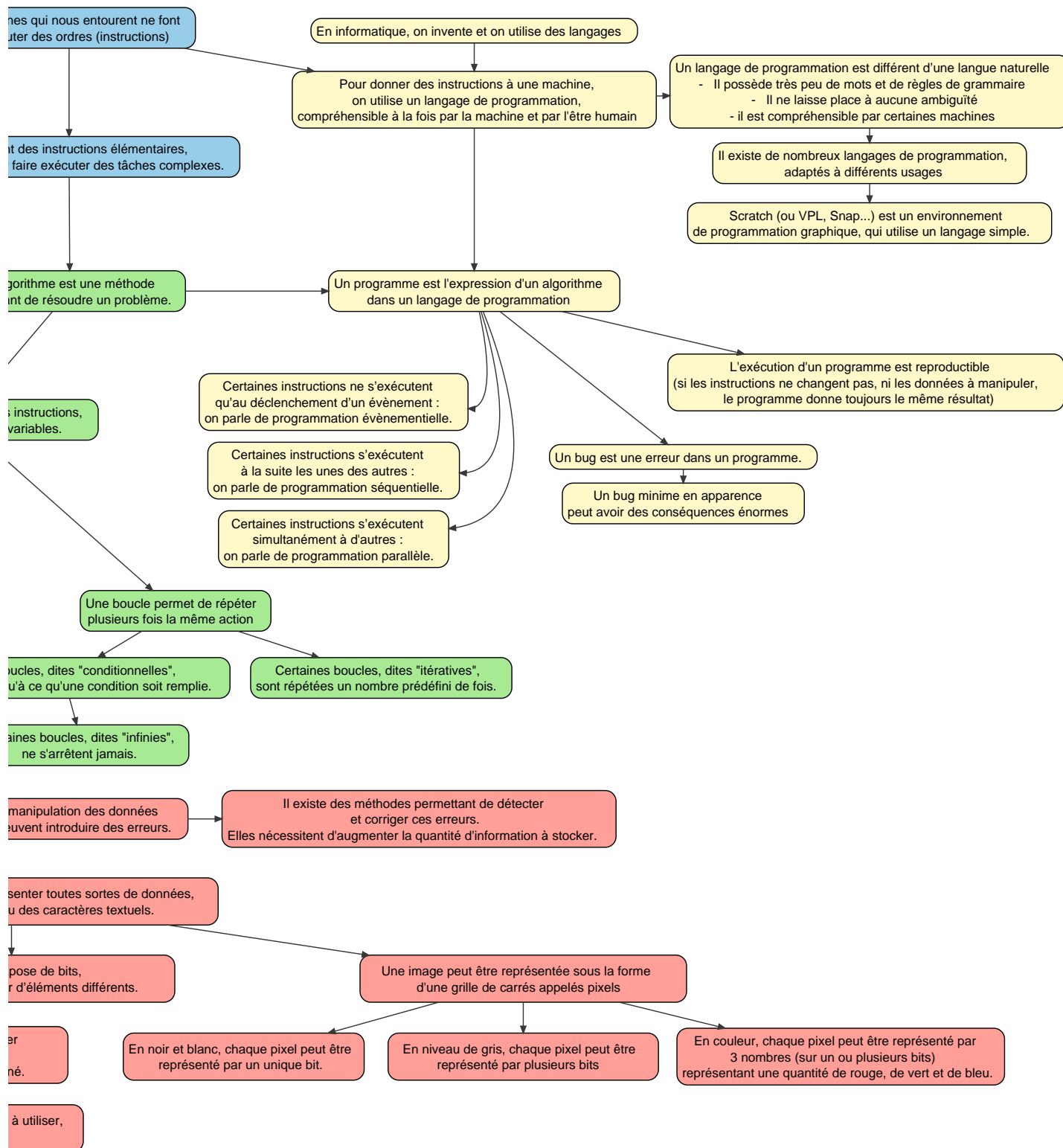
Les guides pédagogiques « 1, 2, 3... codez ! » (tomes 1 et 2) permettent aux élèves de s'initier aux 4 concepts de base de l'informatique cités ci-dessus (machine, algorithme, langage, information). Ces concepts peuvent eux-mêmes être déclinés en notions plus élémentaires, dont l'articulation est visible dans les scénarios conceptuels proposés pour chaque séquence pédagogique du tome 1 et pour chaque projet pour le collège du tome 2. Ci-dessous, nous rappelons pour mémoire le scénario conceptuel proposé pour le cycle 3 dans le tome 1. Les projets du tome 2 reprennent le cœur de ce scénario conceptuel et permettent d'en approfondir certains aspects, ou de développer de nouveaux aspects.

# LEGENDE

- machines
- algorithmes
- langages
- information



Scénario conceptuel proposé pour le cycle 3 dans le tome 1. Chaque projet du présent tome 2 s'appuie sur un scénario conceptuel qui reprend le cœur du scénario cycle 3 et le complète.



Les nouveaux programmes de l'école et du collège, applicables à la rentrée 2016, sont moins précis en termes de concepts mais distinguent clairement d'une part les activités de programmation et d'autres part celles relevant de l'algorithmique ou de la représentation de l'information (cf. page 64 pour plus de détails).

## L'intérêt des activités débranchées

Le tome 1 s'appuie systématiquement sur une combinaison d'activités dites « débranchées », au sens où elles ne nécessitent pas de machine informatique (ordinateur, robot...) et d'activités « branchées ». Cette approche nous semble intéressante à plusieurs titres. Premièrement, elle montre que de nombreuses notions importantes en informatique existent indépendamment des machines (même si, en pratique, elles sont utiles pour exploiter les machines). Deuxièmement, elle permet de dissocier la difficulté de compréhension d'un concept et la difficulté à utiliser ce concept sur une machine. Troisièmement, d'un point de vue pratique, elle permet de gérer une éventuelle pénurie de matériel.

Dans le tome 2, certains projets reprennent cette articulation entre activités « branchées » et « débranchées » (projet « cryptographie », projet « conception et fabrication d'un synthétiseur »). D'autres projets ne proposent pas d'activités débranchées, soit parce que ce n'est pas pertinent (cas des projets « robotique avec Thymio » et « domotique avec Arduino »), soit parce que des activités débranchées proposées dans le tome 1 peuvent être reprises ou adaptées à ces projets (cas des projets « Programmation d'un jeu vidéo », qui peuvent bénéficier d'activités débranchées sur les opérateurs logiques ou les variables, par exemple).

## L'intérêt pédagogique de la robotique

La robotique offre la possibilité de lier le monde numérique et le monde physique. C'est un champ privilégié d'application de l'algorithmique et de la programmation souvent plus motivant et plus rassurant qu'un simple écran d'ordinateur. C'est également l'occasion d'initier à une technologie majeure d'aujourd'hui, qui illustre l'intégration de l'informatique dans les objets physiques.

Plusieurs études montrent l'impact très positif de la robotique sur l'apprentissage de concepts informatiques, entre autres<sup>12</sup>. L'impact en est encore renforcé si on y associe une pédagogie active où les élèves expérimentent par eux-mêmes, s'appuyant sur une méthode d'investigation scientifique et une approche coopérative (voir page 60).

L'apprentissage de la démarche scientifique, de l'argumentation, du débat, sont soutenus par l'aspect tangible qu'apporte la robotique. La manipulation d'un objet physique est un élément motivant et un atout supplémentaire dans la compréhension. L'aspect coopératif induit par les activités robotiques offre également un nouvel espace d'expression aux élèves. La possibilité de tester des programmes sur des objets physiques, de valider ou d'invalider des hypothèses en leur donnant une expression tangible, contribue à débarrasser la notion d'erreur du sentiment de sanction intellectuelle, à redonner à l'erreur un statut positif, une étape dans le processus d'apprentissage.

---

12. Lire par exemple « IniRobot: a pedagogical kit to initiate children to concepts of robotics and computer science », D. Roy et al., RIE 2015. <https://hal.inria.fr/hal-01144435>

## Thymio II, un robot taillé pour l'éducation

Nous proposons dans ce guide pédagogique de nombreuses activités de robotique utilisant le robot Thymio II. Pourquoi celui-ci plutôt qu'un autre ?

Le Thymio II est un petit robot (11 x 11 x 5 cm), autonome, mobile et robuste, conçu spécifiquement pour l'éducation, particulièrement adapté de la maternelle au collègue. Le Thymio II est *open source* matériel et logiciel et dispose d'une interface utilisateur riche et ergonomique. Il peut être utilisé directement dès sa sortie de la boîte, il n'y a rien à construire pour le faire fonctionner, mais des plots Lego® sur le dessus et sur les roues permettent de nombreuses compositions supplémentaires.

En entrée, il dispose de cinq capteurs de proximité à l'avant, de deux à l'arrière, et de deux capteurs en dessous qui mesurent la luminosité du sol et peuvent ainsi distinguer des zones de différentes couleurs (pour suivre une ligne noire, par exemple). Il y a cinq boutons sensitifs sur le dessus, un microphone, un accéléromètre à trois axes pour mesurer des inclinaisons ou des chocs, un capteur infrarouge pour une télécommande extérieure et un thermomètre. En sortie, en plus des deux moteurs pour entraîner les roues, 39 LED sur l'ensemble du corps, dessus, dessous et sur les côtés permettent au robot d'utiliser de très nombreuses combinaisons de couleurs. Le robot dispose également d'un haut-parleur, d'un synthétiseur de sons et d'un emplacement micro SD pour stocker des fichiers musicaux par exemple. Tous les capteurs et actionneurs évoqués ci-dessus sont accessibles par programmation.

La programmation du robot se fait :

- soit par une interface de programmation visuelle très intuitive, accessible aux débutants, même aux enfants non lecteurs,
- soit par une interface de programmation textuelle utilisant un langage script simple,
- soit par les logiciels de programmation visuelle/textuelle *Scratch*, *Snap!* et *Blockly*. Pour programmer Thymio à l'aide de *Scratch* voir par exemple le site suivant : <https://dm1r.inria.fr/t/piloter-un-thymio-ii-avec-scratch/161>

La combinaison de la robustesse, de la richesse des capteurs et des actionneurs, de l'ergonomie de l'interface, de la possibilité d'une programmation totalement visuelle mais aussi de la programmation avec de nombreux autres langages font de Thymio un robot remarquablement adapté à l'éducation.



Le robot Thymio II (à gauche) et l'environnement de programmation graphique VPL 1.5 (au centre et à droite)



# Enseigner l'informatique par une pédagogie active : démarche d'investigation et démarche de projet

Le projet pédagogique « 1, 2, 3... codez! », pluridisciplinaire, met en avant l'activité des élèves par le questionnement, l'expérimentation, l'observation, le tâtonnement, la programmation ou le débat. Cette « pédagogie active » peut prendre des formes sensiblement différentes, notamment la démarche d'investigation ou la démarche de projet.

L'informatique, contrairement aux sciences de la nature (physique, biologie...), n'étudie pas un monde qui préexiste, mais un monde créé par l'homme. Si la démarche d'investigation s'applique aussi bien à l'étude des objets naturels que des objets artificiels, l'apprentissage d'une « science des objets artificiels », telle que l'informatique, demande aussi d'apprendre à en construire par soi-même. C'est pour cette raison que la démarche d'investigation doit être complétée par une démarche de projet.

Certaines séances du projet « 1, 2, 3... codez! » visent plutôt l'appropriation de concepts et s'inscrivent clairement dans une démarche d'investigation, telle qu'on peut la concevoir dans l'enseignement des sciences « classiques ». D'autres séances ciblent davantage le développement de compétences (ce qui ne les empêche pas de permettre la compréhension de concepts!) : c'est particulièrement le cas des séances de programmation sous *Scratch*. Ces séances s'inscrivent davantage dans une démarche de projet. Les séances de robotique sont intermédiaires : le robot peut donner lieu à la réalisation d'un projet, mais aussi constituer un champ d'investigation en lui-même.

## Qu'est-ce qu'une démarche d'investigation ?

La démarche d'investigation, mise en avant notamment par *La main à la pâte* depuis 20 ans, est désormais bien implantée à l'école primaire et fait son chemin au collège. Tout en reconnaissant qu'une schématisation de cette démarche est forcément réductrice, nous pouvons tout de même identifier 3 phases en général présentes :

- un questionnement, initié par le professeur ou les élèves, donnant lieu à la formulation d'hypothèses ;
- une recherche qui peut être une expérience, une observation, une étude documentaire...
- une structuration des savoirs entraînant, à son tour, un nouveau questionnement, une nouvelle recherche, etc.

Les paragraphes qui suivent éclairent rapidement les grandes phases d'une démarche d'investigation.

### La phase de questionnement

De la diversité des réponses fournies à ce questionnement, de leur confrontation, voire de leurs divergences, va se dessiner un problème que les élèves auront à résoudre. Le rôle du professeur est de conduire la discussion qui amènera les élèves à prendre conscience du problème, de ce qu'ils cherchent à savoir ou à montrer. Pour cela, il encourage la communication entre les élèves et les guide dans leur réflexion : « Et vous, que diriez-vous ? Qu'en pensez-vous ? »

### La formulation des hypothèses

En s'appuyant sur son expérience ou ses connaissances, l'élève donne des explications qu'il juge plausibles : ses hypothèses. C'est en passant par l'investigation, via une recherche documentaire, une expérimentation et/ou une modélisation que les élèves vont pouvoir confirmer ou infirmer ces hypothèses. La phase de recherche est ainsi suscitée par la nécessité de tester la pertinence d'une hypothèse.

La formulation des conceptions ou des hypothèses des élèves (ce qu'ils pensent savoir, ce qu'ils pensent comprendre et pouvoir expliquer d'un phénomène) peut être faite de façon individuelle ou collective.

- À l'écrit sous la forme :
  - d'un dessin ou d'un schéma légendé ;
  - d'un texte argumenté ;
  - d'une liste élaborée collégalement.
- À l'oral, sous la forme d'une discussion collective argumentée entre les élèves.

### **La phase de recherche**

Lors de cette phase, toujours guidé par le professeur, l'élève s'investit, seul ou en groupe, dans la recherche de solutions au problème posé. Il s'agit de mettre à l'épreuve les « hypothèses » retenues. Le professeur veille à ce que les modalités de recherche soient trouvées par les élèves eux-mêmes, ceux-ci ne devant pas être de simples exécutants. Il peut toutefois les aider en cas de blocage, par exemple en leur présentant le matériel disponible.

Lorsque l'expérience, la modélisation ou l'observation directe ne sont pas possibles, la recherche documentaire ou même l'interview d'un « savant » (qui peut être le professeur) permettent aux élèves de valider ou de réfuter les hypothèses précédemment émises.

### **La structuration des savoirs**

On a vu combien le questionnement tient une place essentielle tout au long de l'investigation, qu'il s'agisse de poser un problème, d'interpréter le résultat d'une expérience, de confronter des points de vue... Il faut parfois plusieurs allers-retours entre questionnement et recherche avant de pouvoir répondre au problème et construire ainsi de nouvelles connaissances.

C'est lors de la phase orale collective que la classe construit véritablement un savoir commun. Le débat y tient une place primordiale. Cette mise en commun ne doit pas être vue comme un dialogue entre élèves et enseignant, mais comme un dialogue entre élèves, facilité par le professeur.

Toute la classe participe à l'élaboration d'une trace écrite collective, qui fait consensus et qui résume ce qui a été appris et compris. Cette conclusion permet également de prendre de la distance avec l'activité réalisée afin de pouvoir commencer à généraliser et à conceptualiser. La précision du vocabulaire devient ici centrale. La trace écrite collective est souvent textuelle, mais peut s'agrémenter d'autres formes de présentation : graphique, schéma, frise chronologique...

La conclusion de la classe fait consensus..., mais cela ne signifie pas qu'elle soit valable ! On peut être tous d'accord et tous se tromper ! Une étape essentielle, souvent oubliée, de l'investigation est la nécessaire confrontation du savoir construit en classe (nos conclusions) avec le savoir établi (ce que savent les scientifiques). Cette confrontation se fait à l'aide de livres, de documents... ou même avec le professeur qui est, lui aussi, dépositaire du savoir établi.

Dans le module pédagogique « 1, 2, 3... codez ! » des conclusions types sont présentées à la fin de chaque séance. Il s'agit bien entendu d'exemples (basés sur des tests réalisés en classe) destinés à guider le professeur. Il serait cependant dommage que ces conclusions soient utilisées telles quelles. Nous recommandons aux enseignants de laisser leurs élèves élaborer leurs propres conclusions, basées sur le travail effectivement réalisé en classe.

### **Les rôles du professeur dans la démarche d'investigation**

Même si l'activité de l'élève est prépondérante et favorisée, le professeur a un double rôle, essentiel. Il n'est plus seulement celui qui transmet des connaissances, mais aussi celui qui aide les élèves dans un

cheminement vers la construction, par eux-mêmes, de connaissances et l'acquisition de savoir-faire et de savoir-être.

Pour cela, il s'appuie sur la connaissance qu'il a des capacités de ses élèves et aussi de l'état de la progression de l'ensemble de la classe. Il est attentif à l'atmosphère générale comme au rythme de travail de chacun ou des groupes, apporte son soutien ou relance la réflexion quand cela est nécessaire et joue un rôle modérateur (« Et toi, à ton avis... », « Êtes-vous d'accord avec ce qui a été dit? »...). Il décide ou non de passer à une autre activité, à des moments de recentrage ou de généralisation. Pour tout cela, il est dit « tuteur » de la classe.

Le professeur est aussi le garant des « faits » observés, de leur normalité, en tant qu'intermédiaire entre la science « officielle » (celle des savants) et les élèves. Il décide aussi, en se justifiant, de la prise en compte ou non des propositions des élèves, de leur traitement et, finalement, en tant qu'expert ou référent, de la qualité scientifique des résultats des travaux de la classe. Et pour cela, il est dit « médiateur » scientifique de la classe.

## Qu'est-ce qu'une démarche de projet ?

La démarche de projet s'inscrit pleinement dans le courant des pédagogies actives. Formalisée au début du XX<sup>e</sup> siècle (en premier lieu par John Dewey<sup>13</sup>, qui est aussi à l'origine de la démarche d'investigation), elle a longtemps été cantonnée à l'école primaire, avant de s'étendre progressivement dans l'enseignement secondaire et supérieur.

Le guide pédagogique « 1, 2, 3... codez ! » propose trois séquences entièrement construites selon cette démarche ; il s'agit des séquences dédiées à la programmation d'un jeu vidéo en *Scratch* (jeu d'arcade et jeu de plateforme), ainsi que de la séquence « Domotique avec *Arduino* ».

Ce qui caractérise cette démarche :

- Les élèves travaillent autour d'une tâche complexe (un problème autour duquel mener une enquête, pour le résoudre ; une question pour laquelle il s'agit de chercher une réponse). Dans le cas du projet « 1, 2, 3... codez ! » l'objectif de la démarche projet est de permettre aux élèves d'apprendre à programmer à partir d'une question ouverte et non pas d'exercices répétitifs (des exercices sont cependant nécessaires, en début de projet, pour s'approprier l'environnement de programmation) ;
- La tâche peut être concrète, et mener à une production (jeu vidéo, maquette de maison...) ; ou plutôt conceptuelle et mener à la construction d'arguments pour une discussion, pour justifier un point de vue (par exemple, un atelier philo).
- L'étendue du projet dans le temps permet aux élèves, en groupe ou individuellement, de mener une vraie enquête pour répondre aux questions ou aux défis qui se dégagent de la tâche. Ces questions sont souvent ouvertes, au sens où elles n'ont pas nécessairement de réponse unique et pré-écrite.
- Le projet est géré par le groupe classe et non par le professeur seul (qui anime mais ne prend pas de décision seul) ;
- Cette tâche complexe peut être découpée en tâches plus élémentaires dans lesquelles les élèves peuvent avoir un haut degré d'autonomie et s'impliquer activement.

---

13. Dans la littérature anglo-saxonne, cette approche est nommée *Project Base Learning* (PBL).

Les bénéfiques d'une telle démarche sont nombreux :

- L'élève est conscient de ce qu'il fait... et de pourquoi il le fait!
- Le projet, par son caractère concret, est particulièrement motivant. Il est important que la tâche soit engageante pour les élèves et qu'elle fasse sens *de leur point de vue*, pour qu'ils aient envie de résoudre le problème. Mais aussi qu'elle soit perçue comme réalisable par eux, que le défi ne leur apparaisse pas hors de leur portée. Trop simple, elle est sans intérêt; trop difficile, elle est décourageante. Le fait de vivre une expérience « authentique » (on ne résout pas un exercice scolaire, on fabrique un vrai jeu vidéo!) est source de grande satisfaction.
- L'apprentissage des concepts est facilité par le fait que ceux-ci sont contextualisés et prennent sens. La question « comment, dans notre jeu vidéo, compter notre score ou notre nombre de vies ? » va mener les élèves vers la notion de variable, ainsi que les actions qui vont avec : comment créer une variable ? pourquoi faut-il l'initialiser ? comment le faire ? comment changer sa valeur ? comment utiliser cette valeur dans une autre partie du programme (par exemple, quand le nombre de vies vaut zéro, il faut mettre fin au jeu) ? Etc. Les connaissances et compétences mobilisées au cours du projet sont très diverses, sans découpage disciplinaire arbitraire.
- Les élèves changent leur rapport à l'erreur, qui devient partie intégrante du processus d'apprentissage. En effet, ils sont très fortement incités à tâtonner, à procéder par essai-erreur. D'autant plus quand le projet consiste à programmer un jeu : se tromper ne présente aucun danger, il n'y a pas de fausse manip possible (un programme erroné n'abîme pas l'ordinateur!) et le verdict tombe immédiatement. C'est d'ailleurs quelque chose que les élèves apprécient particulièrement : on sait tout de suite si ce qu'on a proposé marche ou pas, et c'est la machine qui le dit, pas le professeur ou le camarade. La facilité de l'essai et l'absence de conséquence dramatique de l'erreur facilitent l'autonomie et la prise de décision. L'autonomie, dans un projet de programmation, n'est pas feinte : il existe souvent plusieurs méthodes (plus ou moins élégantes) de résoudre un même problème.
- Le projet permet d'exercer de nombreuses compétences transversales, comme la prise de décision, la planification, etc. Même s'il est possible de suivre un cours sur la conduite de projet, il s'agit d'un domaine dans lequel l'expérience prime : on apprend à mener un projet à bien en le faisant.
- L'intelligence collective est mise en valeur : la coopération se révèle indispensable car, si le projet est bien choisi, chaque élève sera confronté à un moment ou à un autre à une tâche qu'il ne sait pas résoudre seul. Démarrer l'année par un projet est d'ailleurs un excellent moyen pour créer du lien social dans la classe, pour souder le groupe à travers une activité fédératrice. Le projet permet aussi, par son côté concret et « non scolaire », de révéler certains élèves en difficulté qui, d'habitude sont plutôt en retrait.

### **Les rôles du professeur dans la démarche projet**

Tout comme pour la démarche d'investigation, la démarche projet met en avant l'activité des élèves. Cela ne signifie pas pour autant que le professeur n'a pas un rôle à jouer ; bien au contraire !

Le professeur est le garant de l'intérêt scientifique et pédagogique du projet, il aide à sa définition et veille à ce que ses objectifs soient atteignables par les élèves. Il anticipe les différentes étapes et les traduit en tâches plus élémentaires dans lesquelles les élèves pourront être davantage autonomes.

Quand cela s'avère nécessaire, le professeur supervise la répartition des tâches entre les élèves.

Le professeur est aussi facilitateur : il recentre l'activité ou la discussion sur le problème à régler, rappelle l'objectif à atteindre, et ne porte pas de jugement a priori sur les propositions des élèves afin de les

encourager. De même, il veille à ce que chaque élève soit impliqué dans le projet. Concrètement, dans les séquences de programmation, nous recommandons au professeur de faire en sorte que les élèves travaillent en binôme, mais en changeant de place toutes les 10-15 minutes de façon à ce que chacun ait « la main » (c'est-à-dire tienne le clavier et la souris) à tour de rôle.

Le professeur est enfin régulateur lors des échanges en classe entière, en distribuant la parole, en aidant à la formulation de synthèses... Enfin, il peut jouer un rôle de conseiller / d'expert en proposant des pistes de solutions ou nouvelles connaissances dans le cas où les élèves sont dans une impasse.

## L'informatique dans le Socle Commun

Pour la première fois, les sciences informatiques sont apparues à la rentrée 2016 dans les programmes des cycles 1 à 4. Les séquences proposées dans le tome 2 de « 1, 2, 3... codez! » ont été conçues pour aider à la mise en œuvre de ces nouveaux programmes au cycle 4 (ou éventuellement en 6<sup>e</sup>, pour les projets « Jeu d'arcade » et « Domotique avec Arduino »). La diversité des contenus et des démarches permet également l'acquisition de connaissances et de compétences conformes aux attentes du socle commun, notamment les compétences langagières et collaboratives.

Afin de faciliter la mise en relation de ce module avec le Socle Commun, nous présentons ci-dessous les capacités telles qu'elles y sont formulées. Nous ne listons cependant que les compétences directement liées à la science informatique, en excluant volontairement les compétences transverses liées à la démarche (écrit, oral, production, débat, expérimentation, TICE, etc.) bien qu'elles soient largement travaillées tout au long du projet.

Par ailleurs, le lecteur trouvera au début de chacun des 6 projets la liste des points du programme du cycle 4 spécifiquement traités, pour les diverses disciplines concernées.

### **Domaine 1 : les langages pour penser et communiquer**

#### **Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques**

L'élève [...] lit des plans, se repère sur des cartes. Il produit et utilise des représentations [...] telles que schémas, croquis, maquettes, [...]. Il lit, interprète, commente, produit des tableaux, des graphiques et des diagrammes [...]. Il sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples.

### **Domaine 4 : les systèmes naturels et les systèmes techniques**

#### **Démarches scientifiques**

L'élève sait mener une démarche d'investigation. Pour cela, il décrit et questionne ses observations; il prélève, organise et traite l'information utile; il formule des hypothèses, les teste et les éprouve; il manipule, explore plusieurs pistes, procède par essais et erreurs; il modélise pour représenter une situation; il analyse, argumente, mène différents types de raisonnements (par analogie, déduction logique...); il rend compte de sa démarche. Il exploite et communique les résultats de mesures ou de recherches en utilisant les langages scientifiques à bon escient.

#### **Conception, création, réalisation**

L'élève imagine, conçoit et fabrique des objets et des systèmes techniques. Il met en œuvre observation, imagination, créativité, sens de l'esthétique et de la qualité, talent et habileté manuels, sens pratique, et sollicite les savoirs et compétences scientifiques, technologiques et artistiques pertinents.

### **Domaine 5 : les représentations du monde et l'activité humaine**

#### **Invention, élaboration, production**

L'élève imagine, conçoit et réalise des productions de natures diverses, [...]. Pour cela, il met en œuvre des principes de conception et de fabrication d'objets ou les démarches et les techniques de création. [...] Il mobilise son imagination et sa créativité au service d'un projet personnel ou collectif.

Pour mieux connaître le monde qui l'entoure [...], l'élève pose des questions et cherche des réponses en mobilisant des connaissances sur :

- les éléments clés de l'histoire des idées [...];
- les grandes découvertes scientifiques et techniques et les évolutions qu'elles ont engendrées, tant dans les modes de vie que dans les représentations