

Sommaire

F oire aux questions	6
-----------------------------------	---

Les 20 questions que vous vous posez

U ne histoire de l'informatique	12
--	----

Des abaquas aux smartphones

Partie 1 S'informer, se former	19
---	----

Éléments pédagogiques	20
------------------------------------	----

La place dans les instructions officielles	20
--	----

L'intérêt d'une initiation à la programmation	20
---	----

Les compétences et connaissances développées	20
--	----

Les volumes horaires	21
----------------------------	----

Bibliographie / webographie	21
-----------------------------------	----

Présentation des logiciels	23
---	----

Le logiciel GéoTortue	23
------------------------------------	----

1. Présentation générale	23
--------------------------------	----

2. Description de l'interface	23
-------------------------------------	----

3. Installation	25
-----------------------	----

4. Séance de prise en main	26
----------------------------------	----

5. Trucs et astuces	27
---------------------------	----

Le logiciel Scratch	28
----------------------------------	----

1. Présentation générale	28
--------------------------------	----

2. Description de l'interface	28
-------------------------------------	----

3. Utilisation et installation	29
--------------------------------------	----

4. Séance de prise en main	30
----------------------------------	----

5. D'autres fonctionnalités	32
-----------------------------------	----

6. Trucs et astuces	32
---------------------------	----

Partie 2 Mettre ses élèves en activité 33

Préalables 34

Séances de prise en main 39

Modules et projets pour les élèves 45

Niveaux
recommandés

Modules utilisant GéoTortue

Module 1 Déplacements sur quadrillage 45 ◀ CE1-CE2

Module 2 Figures du plan 55 ◀ CE2

Module 3 Déplacements sur quadrillage 63 ◀ CM1-CM2

Module 4 Art et géométrie 71 ◀ CM1-CM2

Module 5 Les frises 85 ◀ CM1-CM2

Module 6 Les kaléidoscopes 95 ◀ CM2

Petits plus 105 ◀ CM1-CM2

Projets utilisant Scratch

Projet 1 La biographie de Van Gogh 113 ◀ CM1-CM2

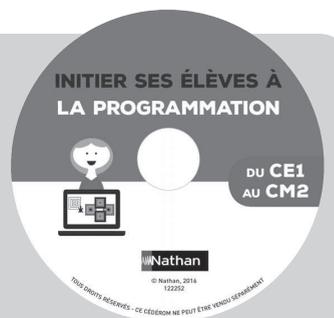
Projet 2 Le conte multimédia 119 ◀ CM1-CM2

Les fichiers du CD-Rom (voir page 37)

- ▶ Version élève d'« Une histoire de l'informatique ».
- ▶ Tutoriels à destination des enseignants (pour GéoTortue et pour Scratch).
- ▶ Fichiers nécessaires à la mise en place des modules et des projets :

- pour chaque séance, les fichiers destinés à l'affichage lors des phases collectives ;
- les fiches d'activités à imprimer pour les élèves ;
- des fichiers « réponse », qui donnent une solution possible parmi d'autres.

Tous les fichiers sont également téléchargeables sur le site initier-programmation.nathan.fr



1

Ça veut dire quoi, « programmer » ?

Avant toute chose, expliquons ce qu'est un **programme** : c'est ce qui permet de doter un ordinateur d'une certaine fonctionnalité, de lui faire exécuter des actions (calculer, dessiner, afficher une vidéo, chercher une information sur Internet, etc.). Les **programmes** informatiques sont partout autour de nous : dans les ordinateurs, évidemment, mais aussi dans les téléphones portables, les voitures, les GPS, les ascenseurs, etc.

On parle beaucoup, également, de **logiciels**. En fait, un logiciel est un ensemble de plusieurs programmes. Par exemple, votre tableur est un logiciel composé d'un programme qui permet de faire des calculs, d'un autre programme pour produire des graphiques, d'un autre pour mettre en forme un document, parmi une multitude d'autres programmes.

Il existe énormément de logiciels prêts à l'emploi, c'est-à-dire réalisés par des **programmeurs**. Cependant, on peut vouloir faire quelque chose de bien spécifique à son ordinateur. Il faut alors pouvoir « communiquer » avec lui dans le but de le **programmer**. Malheureusement, le seul langage qu'un ordinateur comprenne est le langage binaire, une suite de 0 et de 1. Et écrire des ordres à son ordinateur en utilisant de longues séries de 0 et de 1 s'avère vite fastidieux... On peut alors utiliser un **langage de programmation** (le Cobol, le Fortran, le C, le Java, le HTML et, dans le monde de l'éducation, le LOGO, parmi de nombreux autres), sorte d'interprète compréhensible à la fois par nous, êtres humains, et par l'ordinateur.

2

« Coder », « programmer » : quelle est la différence ?

La différence entre « coder » et « programmer » est d'ordre lexical. Pour les spécialistes, « coder » désigne le fait de mettre en place un programme dans un langage de programmation précis, tandis que « programmer » renvoie à l'activité de conception, en amont du code. Dans le vocabulaire courant, « coder » est synonyme de « programmer ».

Il n'est sans doute pas nécessaire de distinguer les deux à l'école élémentaire. On emploiera donc l'un ou l'autre, « coder » ou « programmer », indifféremment. Les programmes, eux, utilisent le terme « programmer » (logique, non?).

3

Et l'« algorithmique » ?

Un **algorithme**, c'est une séquence d'actions réalisant un traitement. L'algorithmique est ainsi la discipline qui consiste à **analyser** des problèmes, à les **décomposer** pour les **résoudre**. Par exemple, si le problème posé est « comment sortir une voiture du garage ? », un algorithme peut être :

1. ouvrir la porte du garage

2. prendre la clef de la voiture
 3. ouvrir la porte avant-gauche de la voiture
 4. entrer dans la voiture
 5. vérifier que la voiture est au point mort
 6. mettre le contact
- ...

Il ne s'agit donc pas d'informatique mais bien de **réflexion** et d'**analyse**. Ceci étant, on rencontre souvent le terme d'« algorithme » dans le domaine de l'informatique : pour pouvoir élaborer un programme, avant d'utiliser un langage de programmation, il est de bon ton de mettre au point un algorithme, qui permettra d'avoir une vue d'ensemble plus claire du problème.

Par exemple, pour tracer un carré avec le logiciel GéoTortue, on peut écrire le programme :

```
rep 4 [av 100; td 90]
```

Auparavant, on aura élaboré l'**algorithme** :

Si je veux tracer un carré, je dois tracer 4 segments de 100 pixels formant des angles droits.

4

C'est quoi, un « logiciel de programmation » ?

Un logiciel de programmation est avant tout un logiciel. C'est donc, à ce titre, un ensemble de programmes écrits par un-e **programmeur-e**. Mais ce logiciel ne sert pas à naviguer sur Internet ou à regarder une vidéo : il permet de programmer, c'est-à-dire de **faire faire quelque chose à son ordinateur**. Autrement dit, c'est un ensemble de programmes qui permet d'écrire d'autres programmes...

5

Je n'ai jamais programmé. Puis-je me lancer avec mes élèves ?

Oui. Logiquement, après avoir lu cet ouvrage, vous devriez être suffisamment équipé-e pour mettre en place une séquence d'initiation à la programmation. En choisissant les bons logiciels, par exemple ceux qui sont présentés ici, qui offrent des environnements et des langages de programmation faciles d'accès, vous n'aurez besoin que de **logique** et de **rigueur** (les mêmes qualités demandées aux élèves devant une tâche de programmation).

6

D'ailleurs, même sans parler de programmer, je n'ai aucune compétence dans le domaine des nouvelles technologies !

Nul besoin d'être un-e *digital native* ou un-e *geek* pour initier ses élèves à la programmation. Cependant, aborder les nouvelles technologies sans crainte peut aider... Alors, domptez vos appréhensions, passez le cap ! Cela en vaut véritablement la peine.

7

En fait, ça me fait peur...

Il ne faut pas. Muni-e de cet ouvrage et d'un peu de curiosité, vous n'aurez aucun problème à initier vos élèves à la programmation, soyez-en sûr-e.

8

Ça ne va pas révolutionner l'enseignement à l'école primaire!

Ce n'est pas le but. Ou plus précisément, ce n'est *plus* le but. Dans les années 1970 et 1980, cela l'a peut-être été pour certains, et notamment pour les concepteurs du langage LOGO, destiné au monde de l'éducation et basé sur des théories de Jean Piaget.

Aujourd'hui, si certaines pratiques d'enseignement *avec* le numérique visent à faire évoluer la pédagogie, à tirer des avantages des nouvelles technologies, l'initiation à la programmation a d'autres objectifs. Il ne s'agit pas d'enseigner *avec* le numérique mais **d'enseigner le numérique**.

Dans le premier cas, les pratiques évoluent en se servant de nouveaux outils : tableaux blancs interactifs (TBI), tablettes tactiles, environnements numériques de travail (ENT), par exemple.

Dans le second cas, l'ambition est de doter nos élèves de compétences solides en compréhension du numérique, de façon à ce qu'ils puissent s'adapter, aujourd'hui et dans leur vie d'adulte à venir, à leur environnement en perpétuelle mutation.

9

Est-il bien nécessaire d'intégrer la programmation au socle commun?

Débattre de la nécessité de cette intégration n'est pas l'objet de ce livre. Celui-ci entend plutôt proposer des outils pour mettre en œuvre un enseignement qui fait son entrée dans le socle commun et les programmes.

On peut toutefois se poser la question : n'est-il pas intéressant, comme on apprend les gestes de premiers secours, comme on éduque à la route, de munir les élèves du plus grand nombre d'éléments possible pour qu'ils réussissent leur intégration dans une société de plus en plus centrée sur le numérique, c'est-à-dire de leur permettre de jouer pleinement leur rôle de citoyen? Le titre d'un article publié sur le site de l'association Enseignement Public et Informatique¹ résume bien cette problématique : « programmer pour ne pas être programmé »... Quant au nombre d'heures à réserver à ces enseignements et à leur place par rapport aux enseignements dits « fondamentaux », chacun se fera son avis.

10

Ce nouvel enseignement doit-il être central dans ma pratique?

Certainement pas. Les programmes parlent bien d'une **initiation** à la programmation. Évidemment, si cela vous intéresse et que vous vous en sentez capable, vous pouvez mener cette initiation sous forme de projet à long terme (proposer et résoudre des « défis programmation » avec une autre classe ou réaliser tout au long de l'année, en programmant, des mandalas pour approvisionner la classe de maternelle de votre école, par exemple). Mais ce n'est pas ce qui vous est demandé dans les instructions officielles.

11

Qu'est-ce que programmer peut apporter à mes élèves?

Répetons-le : initier ses élèves à la programmation, c'est leur enseigner la **pensée informatique** pour qu'ils comprennent mieux le monde numérique qui les entoure.

1. <http://www.epi.asso.fr/revue/articles/a1504e.htm>

Mais c'est aussi leur apprendre à **raisonner** : pour résoudre un problème donné, ils devront **décomposer** ce problème en plusieurs petits problèmes, plus simples. Ils devront écrire un programme avec des règles, être **rigoureux**. Ils devront **tester** leur programme et le **corriger** pour arriver au résultat attendu. Par ailleurs, si vous mettez en place les recommandations de cet ouvrage, vous ferez travailler vos élèves par binômes : la **coopération** sera ainsi travaillée. Si vous choisissez un logiciel de programmation géométrique, vous ferez nécessairement appel à la **créativité** de vos élèves. Ils manipuleront des **angles** et seront amenés à **se repérer** en permanence dans l'espace de l'écran. Autrement dit, vos élèves mobiliseront des compétences fondamentales tout en développant leur compréhension des nouvelles technologies.

Enfin, même si cela est moins évident, les bénéfices d'une initiation à la programmation sont également présents du côté du **langage**. L'ordinateur ne comprendra absolument rien à un élève peu rigoureux qui oubliera une virgule ou qui ajoutera un espace superflu. La nécessité de règles, le caractère conventionnel du langage informatique est directement transférable au langage humain. Ce transfert se fera de façon consciente pour certains enfants, de façon inconsciente pour d'autres.

Sur le site d'un collègue passionné, on pouvait lire, il y a quelque temps, et cela semble tout à fait pertinent : « apprendre à programmer, programmer pour apprendre »².

12

Ça n'est pas un peu compliqué, la programmation, pour des enfants de 8 ans ?

Non, absolument pas. Évidemment, comme pour toute activité, les choix de l'enseignant-e sur les objectifs à atteindre, les modalités et le matériel sont déterminants. Comme pour toute activité, une progression dans les apprentissages est indispensable : on n'apprendra pas à un enfant de CE1 à programmer en HTML – cela n'aurait d'ailleurs aucun intérêt. Comme en grammaire, en anglais, en histoire, il faut proposer en programmation des activités en rapport avec les capacités des élèves.

Dans les médias, le monde de la programmation - de l'informatique de manière générale - est présenté comme un univers très élitiste, avec son jargon et son côté mystérieux. On entend souvent parler d'adolescents « génies de l'informatique ». En réalité, des enfants de moyenne section sont capables de programmer un petit robot qui se déplace. Il suffit, à chaque âge, de proposer les bons outils et les bonnes situations.

13

La programmation ? Mais c'est vieux comme le monde ! J'en ai fait avec mes élèves dans les années 80 !

C'est tout à fait possible. Dans les années 1980, le « Plan Informatique pour Tous » voulait initier les écoliers à l'outil informatique en dotant les écoles de matériel dédié et en formant les enseignants à son usage. Ce fut l'occasion d'un premier accès à l'informatique et d'une première approche de la programmation pour de nombreux enfants. Pour de multiples raisons, d'ordres politique et économique principalement, ce grand plan a été vivement critiqué.

Aujourd'hui, l'initiation à la programmation refait son apparition à l'école. Peut-être pour des raisons politiques et économiques – dont il n'est pas question de débattre ici – mais certainement aussi par bon sens.

2. <http://www.algo.be/logo.html>

14

Qu'est-ce qui peut se faire en matière de programmation à l'école élémentaire ?

Il existe beaucoup de logiciels, plus ou moins confidentiels, qui permettent de faire programmer des enfants. Beaucoup d'entre eux n'ont pas été pensés pour un usage pédagogique ; ils peuvent être mis entre les mains d'un enfant à la maison, pour lui faire découvrir le principe de la programmation, éventuellement pour en faire un futur « codeur ». Quelques-uns seulement ont été pensés pour un usage en classe, dont ceux qui sont utilisés dans cet ouvrage.

Depuis longtemps, il existe des logiciels utilisant le langage LOGO – et donc bien souvent le principe d'une petite tortue à déplacer pour créer des figures géométriques – dont **GéoTortue**, un logiciel libre et gratuit développé par l'IREM Paris-Nord.

Scratch, plus récent, a été conçu au *Massachusetts Institute of Technology* (MIT). Il propose une programmation par manipulation de briques de couleurs. C'est également un logiciel libre et gratuit.

Dans le même esprit, la plateforme Internet code.org met à disposition des exercices ciblés qui permettent d'appréhender très rapidement l'univers de la programmation graphique.

On trouve également des applications pour tablettes : *Robot School* et *Cargo Bot*, par exemple. Les *Lego Mindstorms*, eux, sont des kits robotiques à construire et à programmer. Ils permettent une initiation à la programmation très motivante car concrète.

Enfin, il est tout à fait possible, et intéressant, de se passer d'ordinateur en utilisant des petits robots programmables, comme le camion *Big Trak* des années 1980 et les plus récents *Bee-Bots*. L'achat de robots à programmer représente un coût pour une école. Cet ouvrage fait donc l'impasse sur ces derniers et choisit de se concentrer sur les logiciels téléchargeables librement et gratuitement.

15

Il paraît que c'est bien pour les surdoués !

C'est possible. L'aspect logique de la programmation, la rigueur, l'impossibilité de tricher avec l'ordinateur, la dimension de défi, tout cela serait assez attirant pour les élèves à haut potentiel. Mais pour toutes ces raisons et pour d'autres, la programmation est un domaine intéressant pour tous les élèves, quel que soit leur profil. Elle convient aussi tout à fait aux élèves en difficulté, qui apprécient un résultat assez rapide et valorisant, et aux élèves « moyens » qui suivent avec plaisir cette initiation à un nouvel aspect du monde numérique qui les entoure. Pas plus que tout autre enseignement, l'initiation à la programmation ne doit être pensée ni réservée à un type d'élèves.

16

Ça va être super pour mon groupe de garçons !

C'est une idée reçue qui ne semble pas fondée. L'expérience montre que, au cours de leurs études, les filles ne sont ni plus ni moins intéressées, ni plus ni moins performantes face à la programmation que les garçons. Dans le monde du travail, de nombreuses femmes informaticiennes démontrent au quotidien qu'elles sont au moins aussi compétentes que leurs homologues masculins ! Le « premier programmeur » de l'histoire de l'informatique fut d'ailleurs une femme, Ada King, comtesse de Lovelace, qui a donné son nom à un langage de programmation, l'ADA... Enfin, rappelons au passage que l'École a un rôle central à jouer dans l'égalité filles-garçons.

17

Ils sont déjà suffisamment exposés aux écrans!

C'est vrai. Mais il ne faut pas voir dans cette initiation à la programmation une exposition supplémentaire aux écrans et au monde virtuel. Il faut y voir une exposition différente, où l'élève est **plus actif** et se construit un référentiel pour comprendre l'environnement numérique qui l'entoure. Les activités proposées dans cet ouvrage représentent toutes des **temps d'expérimentation et de recherche**; elles permettront d'approfondir les compétences transversales des élèves.

18

Je suis altermondialiste, de toute façon!

Pensez donc aux *FabLabs* – laboratoires de fabrication ou fabuleux laboratoires, en français, ces ateliers collaboratifs de plus en plus nombreux dont les membres veulent se réapproprier la technologie et ne plus en dépendre, en la comprenant. Pensez à ces objecteurs de croissance qui programment des machines-outils et des imprimantes 3D pour construire leur monde de demain...

19

Je n'ai pas de connexion Internet dans ma classe...

Ce n'est pas très grave. Évidemment, il va falloir installer en classe le logiciel sur lequel les élèves vont travailler. Vous pouvez télécharger ce logiciel sur Internet, chez vous, ou le récupérer auprès d'un-e collègue bienveillant-e ou de votre EATice. Une fois enregistré (par exemple sur une clé USB), le logiciel peut être installé sur les postes de la classe.

20

Je n'ai que 4 postes en fond de classe...

L'aspect matériel peut évidemment être bloquant. Il sera nettement plus facile d'initier ses élèves à la programmation si l'on dispose d'un poste de travail pour 2 élèves. Néanmoins, sans se lancer dans des roulements alambiqués et sans compter plus que de raison sur les capacités d'autonomie d'un grand nombre d'élèves à la fois, les séquences présentées dans la suite de ce livre ont été pensées et testées avec un ratio d'**un ordinateur pour 4 élèves**. À titre d'exemple, pour une classe de 24 élèves, une initiation à la programmation pourra être réalisée dans de bonnes conditions avec 6 ordinateurs: il s'agit de mettre la moitié de la classe au travail, soit 12 élèves, à raison de 2 élèves par ordinateur, et d'opérer un roulement à la fin de la séance. Les séances proposées durant de 30 à 45 minutes, l'autre moitié de la classe peut raisonnablement être mise en autonomie pendant ce temps-là.