



MINISTÈRE
DE L'ÉDUCATION
NATIONALE

*Liberté
Égalité
Fraternité*



N°36

Janvier 2026

La pensée informatique

La pensée informatique est une démarche intellectuelle qui permet d'aborder un problème de manière structurée et efficace. Elle consiste à analyser une situation complexe, à en extraire les éléments essentiels, à repérer des régularités, puis à concevoir une suite d'étapes logiques permettant d'aboutir à une solution. C'est une compétence transversale qui **développe la rigueur, la clarté du raisonnement et la capacité à organiser une démarche.**

La pensée informatique (ou *computational thinking*) est donc une **façon de réfléchir et de résoudre des problèmes**, inspirée des méthodes utilisées en informatique... mais applicable dans toutes les disciplines. C'est une compétence transversale, au même titre que la lecture ou la résolution de problèmes. C'est une **méthode structurée de résolution de problèmes** qui consiste à analyser une situation complexe, à l'organiser de manière logique et à concevoir une solution sous forme d'étapes claires et exécutables.

Elle s'appuie sur quatre opérations mentales : **décomposer, repérer des régularités, abstraire l'essentiel et concevoir un algorithme.** C'est une façon de réfléchir qui peut être utilisée **avec ou sans ordinateur.** Elle aide les élèves à structurer leur pensée, à analyser une situation complexe et à imaginer des solutions efficaces.

SOMMAIRE

ACTUALITÉS	3
La pensée informatique : de quoi s'agit-il ?	3
À quoi sert la pensée informatique ?	3
En quoi est-elle utile pour l'élève ?	3
En quoi est-elle utile à l'enseignant ?	4
Les quatre piliers de la pensée informatique	4
NOUVEAUTÉS DES SERVICES NUMÉRIQUES ÉDUCATIFS	5
Class'Code	6
D-Clics numériques	6
Code - Decode	6
Declick	6
RETOURS D'USAGES NUMÉRIQUES	6
Mathématiques et pensée informatique	6
Construire la pensée algorithmique	7
Initiation à la programmation et à la pensée informatique : les robots souris en cycle 2	7
La pensée informatique en cycle 3	7

La pensée informatique : de quoi s'agit-il ?



La pensée informatique désigne une **démarche intellectuelle** structurée qui vise à aborder un problème de façon ordonnée et reproductible. Elle consiste à **analyser** une situation complexe, à en isoler les éléments pertinents, à repérer des régularités et à concevoir une suite d'actions claires permettant d'atteindre un objectif. Plutôt qu'un ensemble de techniques informatiques, il

s'agit d'un cadre de raisonnement : on formalise le problème, on explicite les étapes de la résolution et on prévoit des moyens de vérifier et d'ajuster la solution. Cette approche rend explicites des processus souvent implicites, ce qui facilite la communication, la collaboration et la transmission des méthodes entre élèves et enseignants.

À quoi sert la pensée informatique ?



La pensée informatique sert avant tout à rendre la résolution de problèmes plus efficace et plus transparente. En transformant une situation floue en une série d'étapes compréhensibles, elle permet de planifier, tester et corriger des solutions de manière systématique. Elle aide aussi à modéliser des phénomènes, à simuler des scénarios et à anticiper des erreurs, ce qui est utile tant pour des tâches

concrètes (organiser un projet, construire une expérience) que pour des raisonnements abstraits (exposer une démarche scientifique, argumenter). Enfin, elle favorise une **posture réflexive** : les élèves apprennent non seulement à trouver une solution, mais à expliquer pourquoi elle fonctionne et comment l'améliorer.

En quoi est-elle utile pour l'élève ?



Pour l'élève, la pensée informatique **développe des compétences transversales** essentielles : logique, capacité à décomposer un problème, sens de l'abstraction, persévérance face à l'erreur et aptitude au travail collaboratif. En pratiquant cette démarche, l'élève gagne en autonomie parce qu'il sait repérer les étapes à suivre

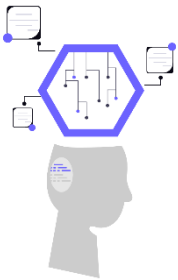
et **évaluer la pertinence** de ses choix ; il gagne en confiance parce qu'il apprend à tester et à corriger ses hypothèses. Ces compétences se transfèrent facilement d'une discipline à l'autre : un élève qui sait formaliser un protocole en sciences pourra structurer un exposé en français ou planifier une construction en technologie. La pensée informatique devient ainsi un **outil d'apprentissage durable**, qui prépare à des situations nouvelles et complexes.

En quoi est-elle utile à l'enseignant ?



Pour l'enseignant, la pensée informatique offre **un cadre pédagogique** pour clarifier les objectifs d'apprentissage, concevoir des tâches progressives et différencier les parcours. En rendant visibles les **étapes d'un raisonnement**, elle facilite l'évaluation formative et permet d'identifier précisément où se situent les difficultés des élèves. Elle aide aussi à concevoir des activités interdisciplinaires et à intégrer le numérique de façon réfléchie : le codage, par exemple, peut servir d'outil pour mettre en œuvre des algorithmes conçus par les élèves, mais il n'est pas la condition première de la démarche. Ainsi, la pensée informatique enrichit la boîte à outils didactique de l'enseignant en offrant des méthodes pour enseigner la résolution de problèmes, la métacognition et la collaboration.

Les quatre piliers de la pensée informatique



On parle de quatre piliers parce qu'ils synthétisent, de façon pédagogique et opérationnelle, les opérations mentales centrales que mobilise l'informatique pour résoudre des problèmes. Ils rendent enseignable et transférable une méthode de pensée applicable dans toutes les disciplines. Les quatre piliers – **décomposer**, **repérer des motifs**, **abstraire** et **concevoir un algorithme** – forment une progression logique qui transforme un problème vague en une solution opérationnelle.

- **Décomposer** consiste à fragmenter une tâche complexe en sous-tâches indépendantes et gérables, ce qui permet d'attribuer, tester et corriger chaque partie séparément ;
Exemples : rédiger un plan détaillé d'un exposé en séparant recherche, sélection des sources, rédaction et répétition ; découper une expérience scientifique en préparation, protocole, observation et conclusion.
- **Repérer des motifs** revient à observer des répétitions ou des structures communes qui simplifient l'analyse et permettent de réutiliser des solutions déjà éprouvées ;
Exemples : identifier des structures récurrentes dans des problèmes mathématiques (par exemple, les types d'opérations) ou des tournures grammaticales qui se répètent dans des textes littéraires.
- **Abstraire** signifie isoler l'essentiel en éliminant les détails non pertinents pour construire un modèle généralisable ;
Exemples : extraire les données numériques pertinentes d'un énoncé de problème en ignorant le contexte narratif ; modéliser un phénomène (par ex. la croissance d'une plante) par une représentation simplifiée.

- Enfin **concevoir un algorithme** consiste à formaliser la solution sous forme d'une suite d'étapes claires, ordonnées et testables, en prévoyant des conditions d'entrée, des boucles éventuelles et des critères d'arrêt.

Travailler ces quatre opérations en classe aide les élèves à expliciter leur raisonnement, à anticiper les erreurs et à documenter leur démarche, et offre à l'enseignant des points d'appui concrets pour l'évaluation formative et la différenciation.

Exemples : écrire une recette de cuisine, formaliser les étapes d'un tri d'objets selon des critères précis, ou décrire pas à pas la procédure pour résoudre un type d'exercice, en incluant des tests et des conditions de correction.

Sources

- Elie Allouche (2 février 2022). Un point sur la pensée informatique (2022) : enjeux de recherche et d'éducation, définitions et repères. *Éducation, numérique et recherche*. Consulté le 17 janvier 2026 à l'adresse <https://doi.org/10.58079/o37u>
- Jeannette Wing (29 mai 2009). La pensée informatique. *Interstices*. Consulté le 17 janvier 2026 à l'adresse <https://interstices.info/la-pensee-informatique>
- Lettre [ÉduNum 1er degré n°35](#)
Cette lettre ÉduNum Premier degré met en lumière l'approche pédagogique STIAM (Sciences, Technologie, Ingénierie, Arts et Mathématiques). Cette approche est une priorité mondiale et européenne visant à développer chez les élèves la pensée critique, la créativité et la résolution de problèmes, les préparant ainsi aux défis du XXI^e siècle. Il s'agit de donner du sens aux apprentissages, encourager la collaboration et la préparation aux métiers de demain.

NOUVEAUTÉS DES SERVICES NUMÉRIQUES ÉDUCATIFS

La page éducol [Programmation et culture numérique](#) présente les **actions nationales** visant à développer chez les élèves – de l'école au lycée – des compétences en **programmation, pensée algorithmique**, culture numérique et création de contenus. Elle met aussi en avant plusieurs **projets et plateformes** soutenus par le ministère pour accompagner enseignants, animateurs et élèves. En ce qui concerne la pensée informatique, l'objectif de ces actions est de développer la pensée algorithmique et l'initiation aux langages de programmation.

Class'Code



Projet qui consiste à fédérer les acteurs et former des intervenants à distance et en proximité, grâce à une plateforme d'apprentissage diffusant un contenu du type « MOOC - Cours en ligne ouvert à tous » tutoré. Il engage à maîtriser la pensée informatique pour la transmettre.

D-Clics numériques



Projet d'éducation active auprès d'enfants de 8 à 14 ans, proposant également des formations courtes adaptées aux enseignants et aux animateurs des collectivités et des associations gestionnaires de structures de loisirs périscolaires, ainsi que des actions d'accompagnement et de diffusion des outils.

Code - Decode



Projet visant à initier et sensibiliser les enfants de primaire (dans un cadre périscolaire) et du collège (dans le cadre scolaire) à la programmation, la littératie numérique et la culture du code.

Declick

Projet bâti avec le soutien apporté par le dispositif Édu-Up, conçu pour initier les 7-15 ans au code et à la création numérique, à travers la programmation écrite. La plateforme constitue ainsi une étape intermédiaire entre des systèmes de programmation graphique (type Scratch) et les vrais environnements de programmation. Il repose sur des parcours progressifs ainsi qu'un logiciel libre et gratuit.

RETOURS D'USAGES NUMÉRIQUES

Mathématiques et pensée informatique



[Le projet EXPIRE](#) (EXpérimenter la Pensée Informatique pour la Réussite des Élèves) a pour objectif de contribuer au développement de l'enseignement de la pensée informatique et des mathématiques à l'école primaire (CM1 et CM2). Il est lauréat du programme e-FRAN soutenu par la caisse des dépôts et le ministère de l'Éducation nationale. Il propose une nouvelle approche s'appuyant sur Scratch et l'algorithmique.

Construire la pensée algorithmique



[Des activités vidéo](#) pour comprendre l'informatique débranchée sont proposées par [Marie Duflot-Kremer](#), maîtresse de conférences en informatique à l'Université de Lorraine. Elle a travaillé dans le cadre du projet Pixees

« Sciences et recherches participatives » à la réalisation de tutoriels dont l'objectif est d'illustrer des activités débranchées pour s'initier aux algorithmes et aux structures de données.

Initiation à la programmation et à la pensée informatique Les robots souris en cycle 2



[Cette séquence](#) constitue une **entrée progressive, structurée et clé en main** pour initier les élèves de cycle 2 à la programmation à travers des activités ludiques et manipulatoires. Elle s'appuie sur une démarche rigoureuse, issue d'un travail collaboratif entre enseignants et chercheurs, et vise à développer chez les élèves les bases de la pensée informatique. Les objectifs pédagogiques en sont : découvrir le langage informatique, comprendre ce qu'est un algorithme, apprendre à programmer, coder et décoder des instructions.

La pensée informatique en cycle 3



[Cette séquence pédagogique](#) initie les élèves de cycle 3 à la pensée informatique. Elle offre six séances progressives pour découvrir les caractéristiques d'un robot, comprendre ses comportements, apprendre à le programmer et relever des défis. Les objectifs principaux en sont : identifier les représentations initiales des élèves sur les robots, découvrir les capteurs, effecteurs et mémoire, utiliser des codes pour programmer des déplacements, résoudre des défis en anticipant le comportement du robot.

Sources

- éducol : [Initiation à la programmation aux cycles 2 et 3](#)
- éducol : [Pratique de l'informatique aux cycles 3 et 4](#)
- éducol : [Initier les élèves à la pensée informatique et à la programmation avec Scratch](#)

**Lettre ÉduNum proposée par la direction du numérique pour l'éducation
Sous-direction de la transformation numérique (DNE – TN3)**

 [Contact courriel](#)

Vous recevez cette lettre car vous êtes abonné à la lettre ÉduNum Premier degré

Souhaitez-vous continuer à recevoir la lettre ÉduNum Premier degré ?

[Abonnement/Désabonnement](#)

À tout moment, vous disposez d'un droit d'accès, de rectification et de suppression des données qui vous concernent (articles 15 et suivants du RGPD).

Pour consulter nos mentions légales, [cliquez ici](#).

Pictogrammes : [unDraw](#) - 2026 Katerina Limpitsouni

ISSN 2739-882X (en ligne)