

GRANDES IDÉES

La décomposition et l'**abstraction** aident à résoudre des problèmes difficiles en les simplifiant.

Les **algorithmes** sont essentiels pour résoudre des problèmes au moyen de l'informatique.

La programmation est un outil qui permet de mettre en pratique la **pensée informatique**.

Résoudre des problèmes est un processus créatif.

La **représentation des données** aide à comprendre et à résoudre efficacement des problèmes.

Normes d'apprentissage

Compétences disciplinaires	Contenu
<p><i>L'élève sera capable de :</i></p> <p>Raisonnement et modéliser</p> <ul style="list-style-type: none"> • Développer une réflexion aisée, souple et stratégique pour analyser et créer des algorithmes • Explorer, analyser et appliquer des idées mathématiques et des concepts informatiques au moyen du raisonnement, de la technologie et d'autres outils • Modéliser au moyen des mathématiques dans des situations contextualisées • Faire preuve de pensée créatrice et manifester de la curiosité et de l'intérêt dans l'exploration de problèmes <p>Comprendre et résoudre</p> <ul style="list-style-type: none"> • Développer, démontrer et appliquer sa compréhension des concepts par des expériences, l'investigation et la résolution de problèmes • Explorer et représenter des concepts et des relations informatiques par la visualisation • Appliquer des approches flexibles et stratégiques pour résoudre des problèmes • Résoudre des problèmes avec persévérance et bonne volonté • Réaliser des expériences de résolution de problèmes qui font référence aux lieux, aux histoires, aux pratiques culturelles et aux perspectives des peuples autochtones de la région, de la communauté locale et d'autres cultures 	<p><i>L'élève connaîtra :</i></p> <ul style="list-style-type: none"> • Accès à des variables dans une mémoire • Différentes organisations de la structure des données dans une mémoire • Différents usages d'ensembles multidimensionnels • Algorithmes classiques, dont le tri et la recherche • Utilisation de la notation grand O pour prédire la performance d'exécution • Résolution de problème par récursivité • Mémoire persistante • Encapsulation de données • Différentes manières de modéliser des problèmes mathématiques

Normes d'apprentissage (suite)

Compétences disciplinaires	Contenu
<p>Communiquer et représenter</p> <ul style="list-style-type: none"> • Expliquer et justifier des concepts et des décisions informatiques de plusieurs façons • Représenter des concepts informatiques sous formes concrète, graphique et symbolique • Utiliser le vocabulaire et le langage de l'informatique et des mathématiques pour participer à des discussions en classe • Prendre des risques en proposant des idées dans le cadre du discours en classe <p>Faire des liens et réfléchir</p> <ul style="list-style-type: none"> • Réfléchir sur l'approche mathématique et informatique • Faire des liens entre différents concepts mathématiques et informatiques, et entre ces concepts et d'autres domaines et intérêts personnels • Voir les erreurs comme des occasions d'apprentissage • Incorporer les visions du monde, les perspectives, les connaissances et les pratiques des peuples autochtones pour établir des liens avec des concepts informatiques 	

Grandes idées – Approfondissements

• **abstraction :**

- simplifier, en représentant les éléments essentiels et en écartant les détails du contexte ou les explications

Questions pour appuyer la réflexion de l'élève :

- Comment décide-t-on s'il faut représenter un objet sous forme abstraite?
- Comment choisir les éléments publics?
- Comment choisir les éléments qui seront affichés?
- Comment le fait d'écarter les détails du contexte simplifie-t-il le processus de résolution de problème?

• **algorithmes :**

Questions pour appuyer la réflexion de l'élève :

- Comment comparer des algorithmes et déterminer lequel est le plus efficace?
- Un algorithme élégant peut-il être efficace?
- Comment formule-t-on un algorithme?
- Qu'est-ce qui fait qu'un algorithme est meilleur qu'un autre?
- Quelle est la relation entre un algorithme élégant et un algorithme efficace?
- Est-ce que tous les problèmes peuvent être résolus par une série d'étapes prédéfinies?

• **pensée informatique :**

- processus de réflexion qui se fonde sur la reconnaissance des régularités et sur la décomposition dans le but de produire un algorithme exécutable par un ordinateur

Questions pour appuyer la réflexion de l'élève :

- Comment choisir le langage informatique à employer pour résoudre un problème donné?
- Pourquoi la lisibilité du code est-elle importante?
- Quels facteurs influent sur la lisibilité du code?
- Quelle quantité de documentation du code source est nécessaire?
- La solution présente-t-elle des régularités qui pourraient être généralisées?
- Comment reconnaître des régularités?

• **résoudre des problèmes :**

Questions pour appuyer la réflexion de l'élève :

- De combien de façons peut-on résoudre tel ou tel problème?
- Comment juger quelle solution est la meilleure?
- Comment procéder pour résoudre un problème de plusieurs façons différentes?
- Si aucune solution n'est suggérée, comment commencer à résoudre un problème?

Grandes idées – Approfondissements

• **représentation des données :**

- une méthode de stockage et d'organisation des données dans un support

Questions pour appuyer la réflexion de l'élève :

- Dans quelles circonstances est-il utile de créer notre propre type de données?
- Comment les ordinateurs utilisent-ils l'électricité pour représenter les données?
- Comment organiser plus efficacement nos types de données?
- Comment choisir les types de données à utiliser?

Compétences disciplinaires – Approfondissements

• **réflexion aisée, souple et stratégique :**

- savoir comparer l'efficacité de différents algorithmes pour résoudre le même problème, trouver l'équilibre entre performance et élégance

• **analyser :**

- examiner la structure des concepts mathématiques et les liens entre eux (p. ex. analyse grand O)

• **raisonnement :**

- raisonnement inductif et déductif
- prédictions, généralisations et conclusions tirées d'expériences (p. ex. en programmation)

• **technologie :**

- technologie graphique, géométrie dynamique, calculatrices, matériel de manipulation virtuelle, applications conceptuelles
- usages très variés, notamment :
 - exploration et démonstration de relations mathématiques
 - organisation et présentation de données
 - formulation et mise à l'épreuve de conjectures inductives
 - modélisation mathématique

• **autres outils :**

- environnements de développement intégrés (EDI)
- débogueur EDI pour dépanner un programme en exécution
- bibliothèques externes
- utilitaires visuels de comparaison pour visualiser les différences de code (p. ex. Meld)
- utilitaires d'analyse de mémoire pour détecter des pertes de mémoire
- systèmes de gestion des versions pour partager du code source entre les membres d'une équipe (p. ex. git)

Compétences disciplinaires – Approfondissements

- **Modéliser :**
 - à l'aide de concepts et d'outils mathématiques, résoudre des problèmes et prendre des décisions (p. ex. dans des scénarios de la vie quotidienne ou abstraits)
 - choisir les concepts et les outils mathématiques nécessaires pour déchiffrer un scénario complexe et essentiellement non mathématique
- **situations contextualisées :**
 - par exemple, des scénarios de la vie quotidienne et des défis ouverts qui établissent des liens entre les mathématiques et la vie quotidienne
- **pensée créatrice :**
 - être ouvert à l'essai de stratégies différentes
 - on fait référence ici à une réflexion mathématique créatrice et innovatrice plutôt qu'à une représentation créative des mathématiques, p. ex. par les arts ou la musique
- **curiosité et de l'intérêt :**
 - poser des questions pour approfondir sa compréhension ou pour ouvrir de nouvelles voies d'investigation
- **investigation :**
 - investigation structurée, orientée et libre
 - observer et s'interroger
 - relever les éléments nécessaires pour comprendre un problème et le résoudre
- **visualisation :**
 - visualiser graphiquement des structures de données
 - utiliser des organigrammes
 - se servir d'utilitaires ou de sites Web de visualisation de code (p. ex. <http://pythontutor.com/>)
- **approches flexibles et stratégiques :**
 - utiliser des algorithmes différents pour résoudre un même problème
 - concevoir des algorithmes capables de résoudre une catégorie de problèmes plutôt qu'un seul problème
 - choisir les régularités de programmation appropriées et des algorithmes éprouvés pour résoudre un problème
 - choisir une stratégie efficace pour résoudre un problème (p. ex. essai-erreur, modélisation, résolution d'un problème plus simple, utilisation d'un graphique ou d'un diagramme, jeu de rôle)
- **résoudre des problèmes :**
 - interpréter une situation pour cerner un problème
 - appliquer les mathématiques à la résolution de problème
 - analyser et évaluer la solution par rapport au contexte initial
 - répéter ce cycle jusqu'à ce qu'une solution vraisemblable ait été trouvée
- **persévérance et bonne volonté :**
 - ne pas abandonner devant les difficultés
 - résoudre les problèmes avec dynamisme et détermination

Compétences disciplinaires – Approfondissements

- **qui font référence :**
 - aux activités quotidiennes, aux pratiques locales et traditionnelles, aux médias populaires, aux événements d'actualité et à l'intégration interdisciplinaire
 - en posant et en résolvant des problèmes, ou en posant des questions sur les lieux, les histoires et les pratiques culturelles
- **Expliquer et justifier :**
 - utiliser des arguments mathématiques pour convaincre
 - prévoir des conséquences
- **décisions :**
 - demander aux élèves de choisir parmi deux scénarios, puis de justifier leur choix
- **plusieurs façons :**
 - par exemple : orale, écrite, pseudocode, visuelle, au moyen de technologies
 - communiquer efficacement d'une manière adaptée à la nature du message et de l'auditoire
- **Représenter :**
 - à l'aide de pseudocodes (p. ex. modèles, tables, organigrammes, mots, nombres, symboles)
 - en établissant des liens de sens entre plusieurs représentations différentes
 - au moyen de matériels concrets et d'outils technologiques interactifs dynamiques
- **discussions :**
 - dialogues entre pairs, discussions en petits groupes, rencontres enseignants-élèves
- **discours :**
 - utile pour approfondir la compréhension des concepts
 - peut aider les élèves à clarifier leur réflexion, même s'ils doutent quelque peu de leurs idées ou si leurs prémisses sont erronées
- **Réfléchir :**
 - présenter le résultat de son raisonnement mathématique et informatique et partager celui d'autres personnes, y compris évaluer les stratégies et les solutions, développer les idées et formuler de nouveaux problèmes et de nouvelles questions
- **Faire des liens entre différents concepts mathématiques et informatiques :**
 - s'ouvrir au fait que l'informatique peut aider à se connaître et à comprendre le monde autour de soi (p. ex. activités quotidiennes, pratiques locales et traditionnelles, médias populaires, événements d'actualité, justice sociale et intégration interdisciplinaire)
- **erreurs :**
 - de syntaxe, de sémantique, d'exécution et de logique
- **occasions d'apprentissage :**
 - en :
 - analysant ses erreurs pour cerner les éléments mal compris
 - apportant des correctifs à la tentative suivante (p. ex. débogage)
 - relevant non seulement les erreurs, mais aussi les parties d'une solution qui sont correctes

Compétences disciplinaires – Approfondissements

- **Incorporer :**
 - en :
 - collaborant avec les Aînés et les détenteurs du savoir parmi les peuples autochtones de la région
 - explorant les principes d'apprentissage des peuples autochtones (<http://www.fnesc.ca/wp/wp-content/uploads/2015/09/PUB-LFP-POSTER-Principles-of-Learning-First-Peoples-poster-11x17.pdf> : l'apprentissage est holistique, introspectif, réflexif, expérientiel et relationnel [axé sur la connexité, les relations réciproques et l'appartenance]; l'apprentissage demande temps et patience)
 - faisant des liens explicites avec l'apprentissage des mathématiques
 - explorant les pratiques culturelles et les connaissances des peuples autochtones de la région, et en faisant des liens avec les mathématiques
- **connaissances :**
 - connaissances locales et pratiques culturelles qu'il est convenable de partager et qui ne relèvent pas d'une appropriation
- **pratiques :**
 - pratiques culturelles selon Bishop : compter, mesurer, localiser, concevoir, jouer, expliquer (http://www.csus.edu/indiv/o/oreyd/ACP.htm_files/abishop.htm)
 - ressources sur l'éducation autochtone (www.aboriginaleducation.ca)
 - *Teaching Mathematics in a First Nations Context*, FNESC (<http://www.fnesc.ca/resources/math-first-peoples/>)

Contenu – Approfondissements

- **Accès à des variables :**
 - passage par valeur et passage par référence, données variables et données immuables
- **structure des données :**
 - vecteurs, listes, files d'attente, dictionnaires, cartes, arbres, piles
- **usages :**
 - jeux de société, traitement d'image, représentation de données tabulaires ou de matrices
- **tri et recherche :**
 - algorithmes de tri (p. ex. par bulles, par insertion, par sélection, rapide, par fusion)
 - algorithmes de recherche (p. ex. recherche binaire, traversée de structure de données)
- **performance :**
 - analyser des algorithmes pour prédire et comparer la complexité d'exécution d'un programme
 - travailler avec de grands ensembles de données
- **Résolution de problème par récursivité :**
 - reconnaître la récursivité dans un problème ou une régularité
 - suite de Fibonacci, exposants, factorielles, palindromes, combinaisons, plus grand commun diviseur, fractales
- **Mémoire persistante :**
 - lecture à partir d'un fichier et écriture dans un fichier
- **Encapsulation :**
 - créer son propre type, sa propre classe ou sa propre structure de données, variables publiques et privées, variables statiques et de classe
- **modéliser des problèmes mathématiques :**
 - estimer une probabilité théorique par la simulation
 - représenter des suites et des séries arithmétiques finies
 - résoudre un système d'équations linéaires, croissance/décroissance exponentielle
 - résoudre une équation polynomiale
 - calculer les valeurs statistiques de grands ensembles de données (p. ex. fréquence, tendance centrale, écart-type)