

Trousse pour la formation des enseignants

Introduction :

Que contient cette trousse ?

Cette trousse est conçue pour les formateurs formant la formation des enseignants de la 6^e à la 9^e année en pensée informatique et programmation. La pensée informatique est un module du nouveau programme d'études Conception, compétences pratiques et technologies (CCPT) pour les élèves de la 6^e à la 9^e année. En 9^e année, le programme CCPT aborde l'apprentissage de la programmation textuelle dans le cadre du module Technologies de l'information et des communications (TIC).

Cette trousse contient notamment des ressources pour des ressources pour enseigner la pensée informatique et la programmation en classe, de la de la 6^e à la 9^e année. Ces ressources sont aussi conçues pour renforcer les compétences essentielles du programme d'études repensé de la C.-B. ainsi que les grandes idées et les compétences disciplinaires du programme CCPT.

Les ressources conçues pour les élèves sont réparties en deux modules :

- . Introduction à la pensée informatique
- . Approfondissement des découvertes

Chaque module comprend les éléments suivants :

- . Jusqu'à 10 heures de cours
- . Des Plans de cours en séquences et des activités pour les élèves
- . Une activité par projet
- . Des grilles d'évaluation

En complément de ces ressources, Des grilles d'évaluation trousse contient des permettant de mieux appréhender le contenu. Tout d'abord, elle donne un aperçu des compétences essentielles du programme repensé de la C.-B. Elle offre aussi des informations plus détaillées sur le programme CCPT, et des explications démontrant comment la pensée informatique est aussi grandement présente dans d'autres programme d'études de la C.-B. Finalement, on y retrouve une introduction de la pensée informatique , présentée comme un ensemble de stratégies de résolution de problèmes, ainsi que des détails sur la façon dont elle est reliée à la programmation.

En plus du contenu décrit ci-dessus, cette trousse propose une méthodologie de mise en place d'ateliers de formation au sein d'une commission scolaire ou d'une école. Elle comprend également des lignes directrices et des ressources pour aider pour aider les enseignants à intégrer la pensée informatique à pour aider cours.

En résumé, cette trousse contient les sections suivantes :

- . [Aperçu du programme d'études de la C.-B.](#)
- . [Pourquoi favoriser la pensée informatique ?](#)
- . [Guide pour les ateliers de formation](#)
- . [Ressources supplémentaires](#)

Aperçu du programme d'études de la C.-B.

La réforme du programme d'études de la C.-B. implantée en 2016 met l'accent sur l'apprentissage personnalisé axé sur les compétences et les concepts. Les compétences essentielles, ci-dessous, sont au cœur de l'ensemble des compétences et se retrouvent dans toutes les disciplines :

- Pensée créatrice
- Pensée critique
- Communication
- Identité personnelle et culturelle positive
- Conscience de soi et responsabilité personnelle
- Responsabilité sociale

Chaque discipline comprend trois éléments suivant le modèle savoir-faire-comprendre :

- Contenu, ce que les élèves doivent savoir
- Compétences disciplinaires, ce que les élèves doivent faire
- Grandes idées, ce que les élèves doivent comprendre

Aperçu du programme Conception, compétences pratiques et technologies

Le nouveau programme Conception, compétences pratiques et technologies (CCPT) est axé sur l'expérience et la pratique et permet aux élèves d'apprendre en créant.

Grandes idées

Les grandes idées abordent l'utilisation des processus de conception, des compétences pratiques et des technologies pendant la création. Les grandes idées de la 6^e à la 9^e année sont décrites ci-dessous :

Grandes idées	de la 6 ^e à la 9 ^e année	9 ^e année
Conception appliquée	La conception appliquée peut répondre à des besoins déterminés.	Des considérations sociales, éthiques et de durabilité influent sur la conception appliquée.
Compétences pratiques	L'exécution de tâches complexes nécessite l'acquisition de compétences supplémentaires.	L'exécution de tâches complexes exige un enchaînement de compétences pratiques.
Technologies appliquées	L'exécution de tâches complexes requiert parfois l'utilisation de plusieurs outils et technologies.	L'exécution de tâches complexes se fait à l'aide d'outils et de technologies variés, selon les étapes.

Compétences disciplinaires

Les compétences disciplinaires définissent les pratiques des grandes idées et sont subdivisées en sous-titres correspondant aux étapes de la conception et de la réalisation suivantes :

- . Comprendre le contexte
- . Définir
- . Concevoir des idées
- . Assembler un prototype
- . Mettre à l'essai
- . Réaliser
- . Présenter

Le tableau ci-dessous établit la liste des compétences disciplinaires de la 6^e à la 9^e année correspondant à chaque étape :

	De la 6 ^e à la 9 ^e année	9 ^e année
Comprendre le contexte	Manifester de l'empathie envers des utilisateurs potentiels afin de déceler les problèmes et de découvrir les besoins et les possibilités de conception.	Entreprendre une période de recherche et d'observation empathique afin de comprendre les possibilités de conception.
Définir	Choisir une possibilité de conception. Identifier les caractéristiques principales ou les utilisateurs et leurs besoins. Établir les critères de réussite et indiquer toute contrainte existante.	Choisir une possibilité de conception. Déterminer les utilisateurs potentiels et développer les facteurs contextuels pertinents. Préciser les critères de réussite, l'effet recherché et toute contrainte existante.
Concevoir des idées	Formuler des idées et développer les idées des autres. Sélectionner les idées en fonction des critères et des contraintes. Examiner les conséquences sur le plan personnel, social et environnemental, et les facteurs liés à l'éthique. Choisir une idée à développer.	Prendre des risques créatifs en formulant des idées et développer les idées des autres de façon à les améliorer. Sélectionner les idées en fonction des critères et des contraintes. Analyser de façon critique et classer par ordre de priorité des facteurs opposés, y compris des considérations sociales, éthiques et de durabilité, afin de répondre aux besoins de la communauté dans des scénarios d'avenir souhaitables. Choisir une idée à développer, tout en demeurant ouvert à d'autres idées potentiellement viables.

<p>Assembler un prototype</p>	<p>Répertorier et utiliser des sources d'inspiration et d'information.</p> <p>Établir un plan indiquant les principales étapes et les ressources à utiliser.</p> <p>Examiner divers matériaux et en faire l'essai pour vérifier leur efficacité.</p> <p>Réaliser une première version du produit ou assembler un prototype, suivant le cas, en changeant s'il le faut les outils, les matériaux et les méthodes employés.</p> <p>Consigner les réalisations des versions successives du prototype.</p>	<p>Répertorier et utiliser des sources d'inspiration et d'information.</p> <p>Choisir la forme du prototype et établir un plan indiquant les principales étapes et les ressources à utiliser.</p> <p>Évaluer divers matériaux quant à leur efficacité, à leur biodégradabilité, et à la possibilité de les recycler et de les réutiliser.</p> <p>Assembler le prototype, en changeant, s'il le faut, les outils, les matériaux et les méthodes.</p> <p>Consigner les réalisations des versions successives du prototype.</p>
<p>Mettre à l'essai</p>	<p>Mettre la première version du produit ou du prototype à l'essai.</p> <p>Recueillir les commentaires et les idées d'un pair, d'un utilisateur ou d'un spécialiste.</p> <p>Apporter des modifications, régler un problème éventuel et refaire un essai.</p>	<p>Concevoir une formule d'essai appropriée du prototype.</p> <p>Recenser des sources de rétroaction.</p> <p>Procéder à l'essai ; compiler les données recueillies, évaluer les données et déterminer les modifications requises.</p> <p>Recréer le prototype ou abandonner l'idée de conception.</p>
<p>Réaliser</p>	<p>Trouver et utiliser les outils, les matériaux et les technologies nécessaires à la production.</p> <p>Établir un plan montrant les principales étapes de production ; l'exécuter en le modifiant au besoin.</p> <p>Utiliser les matériaux de façon à réduire le gaspillage au minimum.</p>	<p>Trouver et utiliser les outils, les matériaux, les technologies et les procédés nécessaires à la production.</p> <p>Établir un plan de production par étape ; l'exécuter en le modifiant au besoin.</p> <p>Utiliser les matériaux de façon à réduire au maximum le gaspillage.</p>

Présenter	Déterminer comment et à qui présenter son produit.	Déterminer comment et à qui présenter son produit et ses procédés.
	Faire une démonstration de son produit et expliquer, à l'aide de termes justes, le processus suivi et justifier ses choix de solutions et de modifications.	Montrer le fonctionnement de son produit à des utilisateurs potentiels, donnant, à l'aide de termes justes, la raison d'être de ses choix de solutions, de modifications et de méthodes.
	Évaluer son produit en fonction de ses critères et expliquer en quoi son concept est utile à l'individu, à la famille, à la communauté ou à l'environnement.	Évaluer, de façon éclairée, l'efficacité de son produit et expliquer en quoi son concept est utile à l'individu, à la famille, à la communauté ou à l'environnement.
	Réfléchir sur sa logique et ses processus de conception et évaluer sa capacité à travailler efficacement aussi bien seul qu'en faisant équipe avec d'autres, y compris sa capacité à partager et à garder en bon état un espace de travail coopératif.	Réfléchir à sa logique, à ses processus de conception et à sa capacité à travailler efficacement aussi bien seul qu'en faisant équipe avec d'autres, ainsi qu'à sa capacité à partager et à garder en bon état un espace de travail coopératif.
	Cerner d'autres problèmes de conception.	Cerner d'autres problèmes de conception.

Contenu

Finalement, voici les normes d'apprentissage des modules de pensée informatique et de programmation de la 6^e à la 9^e année auxquelles s'alignent les ressources destinées aux élèves.

Normes d'apprentissage du contenu des modules de pensée informatique et de programmation :	
De la 6 ^e à la 7 ^e année	<ul style="list-style-type: none"> . Les algorithmes simples qui traduisent la pensée informatique . Les représentations visuelles de problèmes et de données . La programmation visuelle
8 ^e année	<ul style="list-style-type: none"> . Savoir en quoi les programmes d'ordinateur sont des séquences d'instructions précises comportant des algorithmes que d'autres peuvent répéter fidèlement . Déboguer des algorithmes et des programmes en décomposant les problèmes en une série de sous-problèmes
9 ^e année	<ul style="list-style-type: none"> . Programmation textuelle

La pensée informatique dans le programme d'études de la C.-B.

Dans le nouveau programme d'études de la C.-B., la pensée informatique fait partie des normes d'apprentissage du contenu du programme CCPT de la 6^e à la 8^e année. De la 6^e à la 7^e année, le module de pensée informatique comprend des algorithmes simples pour refléter la pensée informatique et faire des représentations visuelles de problèmes et de données. En 8^e année, le module de pensée informatique aborde le débogage d'algorithmes en divisant des problèmes en sous-problèmes.

Bien que ces concepts de pensée informatique fassent partie du programme CCPT, ils sont également présents dans d'autres disciplines du programme de la C.-B. Par exemple, dans le programme d'éducation artistique de la 6^e et de la 7^e année, la notation en musique et en danse et la formation de séquences sont un exemple de création d'algorithmes simples.

Les tableaux ci-dessous offrent des exemples de normes de disciplines de la 6^e à la 8^e année correspondant aux normes d'apprentissage du programme CCPT.

Normes d'apprentissage - 6e année

Matière	Norme d'apprentissage	Lien avec la norme d'apprentissage du programme CCPT
Éducation artistique	La notation en musique et en danse sert à représenter les sons, les idées, les mouvements, les éléments et les actions.	La notation et les séquences sont un exemple d'algorithmes simples.
English Language Arts	Caractéristiques, structures et conventions linguistiques, particulièrement l'organisation des paragraphes.	La création de paragraphes cohérents, développés et cohérents est équivalente à la création d'algorithmes.
Mathématiques	Modéliser les objets et les relations mathématiques dans des expériences contextualisées.	Les mettre en pratique à l'aide d'un contenu concret (p. ex. : en manipulant) ; dessiner des images et des diagrammes, créer des tableaux et des graphiques pour représenter des données visuellement.

	La priorité d'opérations avec des nombres entiers.	La priorité d'opérations avec des nombres entiers est un algorithme simple pour évaluer des expressions.
Éducation physique et santé	Les façons de participer à différents types d'activités physiques, y compris les activités individuelles et à deux, les activités rythmiques et les jeux.	Les jeux comprennent habituellement des règles, qui sont des algorithmes simples.
Sciences	Élaborer et utiliser une variété de méthodes, notamment des tableaux, des graphiques et des technologies numériques, selon les besoins, pour représenter des régularités ou des relations dans les données.	Représenter visuellement des données pour démontrer des régularités ou des relations.
	Les structures et les fonctions de base des systèmes du corps : <ul style="list-style-type: none"> . Système urinaire . Système reproducteur . Système hormonal . Système nerveux 	Les fonctions des systèmes du corps suivent des algorithmes simples.
Sciences humaines	Utiliser les compétences et les processus d'investigation des sciences humaines pour poser des questions, recueillir, interpréter et analyser des idées et communiquer des conclusions et des décisions	La localisation et l'identification de continents, de fleuves et d'océans à l'aide de grilles, d'échelles et de légendes et la présentation de la même information en deux ou plusieurs formes différentes (p. ex. : graphiques, tableaux, cartes thématiques) sont des exemples d'une représentation visuelle des problèmes et des données.
	Ordonner des objets, des images et des événements, et relever les aspects positifs et négatifs des continuités et des changements dans le passé et le présent (continuité et changement).	L'ordonnance d'objets est essentielle à la création d'algorithmes simples.

Normes d'apprentissage de la 7^e année

Matière	Norme d'apprentissage	Lien avec la norme d'apprentissage du programme CCPT
Éducation artistique	La notation en musique et en danse sert à représenter les sons, les idées, les mouvements, les éléments et les actions.	La notation et les séquences sont un exemple d'algorithmes simples.
English Language Arts	Caractéristiques, structures et conventions linguistiques, particulièrement l'organisation des paragraphes.	La création de paragraphes unis ainsi que leur développement et leur cohérence est équivalente à la création d'algorithmes.
Mathématiques	Modéliser les objets et les relations mathématiques dans des expériences contextualisées.	Les mettre en pratique à l'aide d'un contenu concret (p. ex. : en manipulant) ; dessiner des images et des diagrammes, créer des tableaux et des graphiques pour représenter des données visuellement.
	La résolution en deux étapes d'équations dans lesquelles les coefficients, les constantes et les solutions sont des nombres entiers naturels.	Le processus de résolution d'équations en deux étapes est un algorithme simple.
Éducation physique et santé	Les façons de participer à différents types d'activités physiques, y compris les activités individuelles et à deux, les activités rythmiques et les jeux.	Les jeux comprennent habituellement des règles, qui sont des algorithmes simples.
Sciences	Planifier en collaboration une variété de types de recherches, y compris des travaux sur le terrain et des expériences, pour répondre à ses propres questions ou résoudre un problème	La planification d'une recherche nécessite la formulation d'une séquence d'étapes semblable à la création d'un algorithme simple.

	Élaborer et utiliser une variété de méthodes, notamment des tableaux, des graphiques et des technologies numériques, selon les besoins, pour représenter des régularités ou des relations dans les données.	Représenter visuellement des données pour démontrer des régularités ou des relations.
Sciences humaines	Utiliser les compétences et les processus d'investigation des sciences humaines pour poser des questions, recueillir, interpréter et analyser des idées et communiquer des conclusions et des décisions.	La démonstration d'une aptitude à interpréter des échelles et des légendes dans des graphiques, des tableaux et des cartes (p. ex. : diagramme climatique, carte topographique, graphique circulaire) et la sélection d'un type graphique approprié à la situation de communication nécessitent la représentation visuelle de problèmes et de données.

Normes d'apprentissage de la 8^e année

Matière	Norme d'apprentissage	Lien avec la norme d'apprentissage du programme CCPT	Exemple
Mathématiques	Appliquer des stratégies multiples pour résoudre des problèmes dans des situations abstraites et contextualisées; élaborer, prouver et appliquer des solutions mathématiques par le jeu, l'investigation et la résolution de problèmes.	La résolution de problèmes mathématiques nécessite souvent la division des problèmes en séries de sous-problèmes, ce qui est semblable au débogage.	Utiliser le raisonnement proportionnel afin de trouver la meilleure affaire en fonction du prix de deux ou de plusieurs produits.
Sciences	Relever les sources d'erreur possibles et proposer des améliorations à ses méthodes de recherche.	La recherche des sources d'erreur potentielles est l'équivalent d'un débogage et nécessite la division des méthodes de recherche en plus petites étapes.	Évaluer chaque étape d'une expérience et déterminer où pourrait se glisser une erreur.

Pourquoi enseigner la pensée informatique ?

Un des objectifs principaux du programme d'études repensé de la C.-B. est de développer les compétences essentielles des élèves en ce qui concerne la réflexion, la communication et les aptitudes personnelles et sociales afin de les encourager à approfondir leurs connaissances et à apprendre tout au long de leur vie. Plus particulièrement, la pensée créatrice et la pensée critique sont des compétences essentielles permettant aux élèves d'imaginer de nouvelles idées de grande valeur pour autrui et de formuler un jugement selon un raisonnement.

La pensée informatique est un complément à la pensée créatrice et critique. Elle permet aux élèves de résoudre des problèmes complexes. Elle nécessite d'examiner des problèmes sous différents angles pour élaborer des solutions potentielles. En faisant appel à la pensée informatique, les élèves peuvent mettre en œuvre au moins une des stratégies suivantes :

Stratégie	Définition
Décomposition	Diviser quelque chose en plus petites parties.
Reconnaissance des formes	Trouver des similitudes entre des éléments.
Abstraction	Enlever des détails inutiles.
Algorithmes	Former des séquences d'événements.

Les stratégies de pensée informatique sont utiles dans toutes les disciplines d'apprentissage, mais celles-ci se manifestent différemment dans chacune. D'une part, les élèves peuvent faire appel à la stratégie de décomposition pour diviser un problème de mathématiques en plus petites parties et ainsi en faciliter la résolution. D'autre part, ils peuvent y faire appel pour diviser les éléments d'une histoire dans un texte et améliorer leur compréhension des différentes formes et des différents genres.

En programmation, la pensée informatique permet aux élèves de résoudre des problèmes afin que les ordinateurs puissent exécuter les solutions. Elle permet aux élèves tant de consommer la technologie que de faire appel à elle pour exprimer leurs idées et leurs solutions. De plus, la programmation est un processus très itératif. Inévitablement, les élèves feront des erreurs en rédigeant leurs codes. Toutefois, ils verront rapidement les résultats de leur code, y apporteront des changements et itéreront jusqu'à ce qu'ils obtiennent les résultats souhaités. La programmation offre un environnement sécuritaire où les erreurs sont permises, ce qui favorise la résilience.

Ateliers de formation

Présentation des stratégies

La plupart du temps, les participants aux ateliers sont divisés en deux groupes distincts. Le premier groupe adoptera la pensée informatique sans problème, car il est déjà très habitué à la technologie. Le deuxième groupe est plus enclin à être sceptique et ambivalent à son égard.

Il a été démontré constamment que la meilleure façon de bâtir un pont entre le deuxième groupe et le formateur est de présenter une carte conceptuelle simple aux enseignants qui ne sont pas parfaitement à l'aise avec les ordinateurs. La présentation de stratégies de pensée informatique dans des situations au quotidien favorise la confiance et l'ouverture envers les façons dont les humains adoptent naturellement des solutions à des problèmes de tous les jours. Les stratégies de pensée informatique sont renforcées par les compétences disciplinaires. Ces compétences sont souvent faciles à adopter parce qu'elles utilisent des termes que nous employons déjà la plupart du temps.

Pour beaucoup, les stratégies de pensée informatique résonnent comme un jargon incompréhensible. Il vaut mieux aborder ce sujet d'emblée pour anticiper sur le sentiment d'insécurité que pourraient ressentir les participants à l'utilisation de termes techniques ; un phénomène que les spécialistes en psychologie appellent le « syndrome de l'imposteur. » Celui-ci décrit le sentiment d'être submergé par de nouvelles informations, ce qui pousse les gens à être moins réceptifs aux nouvelles idées. Un participant atteint du syndrome de l'imposteur ne se sent pas assez qualifié ou compétent pour accomplir une tâche. En vérité, nous employons souvent les stratégies de pensée informatique, dans l'enseignement comme dans la vie ordinaire.

Algorithmes :

La séquence d'événements qui mène du problème à la solution. Nous les rencontrons dans notre vie quotidienne. Par exemple, ils sont présents dans les recettes, les listes de tâches à faire, les instructions pour assembler des meubles et les motifs de tricot. Ces exemples sont tous des algorithmes, une séquence d'événements menant du problème à la solution lorsqu'elle est bien suivie.

Reconnaissance des formes :

Il s'agit de la façon dont nous reconnaissons les visages, les points de repère et les indications sur une planche de jeu de société. L'habileté du cerveau à reconnaître les aspects récurrents dans les données nous permet de créer des algorithmes qui nous fournissent des informations à propos de celles-ci.

Décomposition :

La décomposition est la division d'un processus complexe en plus petites parties. Pour les élèves du secondaire, un bon exemple est le nettoyage d'une chambre en désordre, qui peut se faire une section à la fois ou à des intervalles de temps. Il s'agit souvent de la façon dont nous créons des algorithmes. Ce processus de division mène aussi bien souvent à la reconnaissance de formes.

Abstraction :

Nous faisons appel à l'abstraction quand nous choisissons un élément pour en représenter un autre. Nous savons tous que le « x » en algèbre est une valeur arbitraire. De plus, nous savons que la couleur rouge signifie « arrêt », que le vert signifie « marche » et qu'une vente dans un magasin d'électronique signifie que nous devons mettre de l'argent sur notre carte de crédit. Nous avons fait une abstraction du sens de ces événements et symboles afin de comprendre que ceux-ci représentent une autre série d'actions. Nous faisons une abstraction quand nous présentons un concept à l'aide de termes ou de symboles bien choisis.

Lien entre les stratégies et les compétences

En établissant un lien entre le programme CCPT et le quotidien, le formateur signe un contrat avec le participant pour la réussite de son apprentissage du nouveau contenu. Les stratégies de calcul sont renforcées et assimilées avec les compétences disciplinaires du programme CCPT. En voici des exemples :

Concevoir des idées et les mettre à l'essai

Nous essayons constamment de nouvelles choses. Qu'arriverait-il si je _____ ? Nous essayons un procédé et vérifions s'il fonctionne. De cette façon, nous mettons à l'essai des algorithmes.

Présenter et collaborer

La collaboration est véritablement au cœur de la pensée informatique. En collaborant, nous comparons notre perception d'un problème et notre solution avec les autres et obtenons des rétroactions. En acceptant d'avoir l'avis des autres, particulièrement en ce qui concerne la reconnaissance de formes, nous décomposons la solution selon leur perspective et faisons appel à leur expérience de débogage et d'expérimentation, ce qui nous permet de trouver une solution encore plus rapidement.

Assembler un prototype, le définir et le planifier

Nous avons tous déjà entendu que la planification prévenait l'échec. En planifiant, nous nous assurons que les choix que nous posons en créant un algorithme sont justifiés et que les techniques d'abstraction et d'expérimentation fonctionnent, ce qui nous permet de prédire les résultats selon nos décisions. Nous résolvons les problèmes de cette façon dans notre vie de tous les jours en concevant et en planifiant des solutions tant pour nous-mêmes que pour les autres.

Réaliser

En nous servant de la pensée informatique, nous sommes capables tant de résoudre des problèmes que de trouver de nouvelles manières de nous exprimer, que ce soit en utilisant des algorithmes pour raconter des histoires, en faisant des abstractions pour créer des animations ou en faisant appel à la logique pour concevoir des jeux.

Grandes idées

En mettant ces stratégies en place, les grandes idées viennent naturellement. L'un des buts principaux du développement de la pensée informatique est de faciliter l'apprentissage. Rappel :

Grandes idées	De la 6 ^e à la 8 ^e année	9 ^e année
Conception appliquée	La conception appliquée peut répondre à des besoins déterminés.	Des considérations sociales, éthiques et de durabilité influent sur la conception appliquée.
Compétences pratiques	L'exécution de tâches complexes nécessite l'acquisition de compétences supplémentaires.	L'exécution de tâches complexes exige un enchaînement de compétences pratiques.
Technologies appliquées	L'exécution de tâches complexes requiert parfois l'utilisation de plusieurs outils et technologies.	L'exécution de tâches complexes se fait à l'aide d'outils et de technologies variés, selon les étapes.

Mise en application

Il s'agit d'aborder le concept du code séparément de celui de la pensée informatique. La pensée informatique n'est qu'une méthode de résolution de problèmes tandis que le code est la façon dont nous indiquons les actions à entreprendre à un ordinateur. Il s'agit ont besoin d'être expliqués différemment, car il s'agit de concepts différents. Le code n'est qu'un aspect du programme CCPT et n'est pas enseigné avant la 6^e année, à l'aide de la programmation visuelle simple, alors que. simple, alors que l'intégration de la pensée informatique au programme est un atout tant pour les élèves que pour les enseignants, quelle que soit l'année ou la discipline.

Suggestion de feuille de route pour un atelier

Voici une suggestion de feuille de route pour organiser un atelier de pensée informatique (PI) au sein d'une école ou d'une commission scolaire. L'une des manières les plus efficaces d'enseigner la PI aux participants est de faire des expériences pratiques avec le contenu, puis de mener une discussion suivie d'une réflexion. Le plan ci-dessous est conçu pour un atelier de quatre heures. Toutefois, si vous avez plus de temps à votre disposition, vous y trouverez aussi des notes pour aller plus loin avec la matière.

Nous recommandons de commencer les ateliers en faisant connaissance et en présentant l'ordre du jour afin que les participants sachent bien à quoi s'attendre. Chaque séance doit se terminer avec une introduction à la ressource Discours.

Première heure

Activité À la rescousse de la première unité du module d'apprentissage. (De 60 à 75 minutes)

- . Divisez les participants en groupes et invitez-les à réaliser l'activité. Celle-ci permettra de briser la glace et d'instaurer un bon climat pour toute la séance. (45 minutes)
- . Ensuite, faites un lien entre les apprentissages et les compétences de l'activité À la rescousse et le programme CCPT. (De 15 à 30 minutes)

Deuxième heure

Discussion (60 minutes)

1. Comment la PI est-elle déjà présente dans la vie de tous les jours et en classe? (De 15 à 20 minutes)
2. Pour chaque stratégie de la PI (décomposition, abstraction, reconnaissance des formes, et formation des séquences), invitez les enseignants à travailler en groupes (un à un) pour établir des liens avec leur enseignement actuel. (De 30 à 45 minutes)

Troisième heure

Aperçu du module (60 minutes)

- . Module 1 (25 minutes)
- . Module 2 (25 minutes)
- . Accent sur l'application et l'implantation multidisciplinaire et transversale (10 minutes)

Quatrième heure

Réflexion (30 minutes)

1. Réfléchir à propos de l'intégration de la pensée informatique au programme et dans l'enseignement.
2. Si le temps le permet, vous aurez l'occasion de faire une réflexion après chaque aperçu de module et une réflexion supplémentaire sur l'utilisation transversale.

Séance de questions et de réponses propres à la commission scolaire ou à l'école (30 minutes)

1. Évaluation
2. Préparation au nouveau programme
3. Répondre aux inquiétudes concernant la perturbation des cours précédents par le nouveau programme.

Séances prolongées

Voici quelques options pour prolonger la séance si le temps le permet :

Utilisez le jeu Jeopardy présenté aux ateliers provinciaux.

- . Faites découvrir aux participants l'activité Kahoot ! ou Twine du module d'apprentissage.
- . Demandez aux participants de réfléchir à propos de critères pour évaluer la pensée informatique.
- . Explorez des concepts d'implantation multidisciplinaire et transversale du contenu sur la PI.

Ressources supplémentaires

Learn to Code, Code to Learn, Mitchel Resnick, EdSurge (anglais)

<https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>

Let's Teach Kids to Code, Mitchel Resnick, TED (anglais, sous-titres français)

http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=en

Education Reform, Brian Aspinall, TEDxChathamKent (anglais)

<https://youtu.be/ngeZPU35zm4>

Hacking the Classroom, Brian Aspinall, TEDxKitchenerED (anglais)

<https://youtu.be/UyxfPnO5lgk>

New Frameworks for Studying and Assessing the Development of Computational Thinking, Karen Brennan and Mitchel Resnick, MIT Media Lab (anglais)

http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AFERA2012_CT.pdf

Scratch est fantastique pour créer des projets. Il s'agit d'un outil convivial conçu pour permettre aux élèves de la maternelle à la 12^e année d'explorer la pensée informatique en faisant appel à leur créativité et à leur curiosité.

(en anglais et en français)

<http://scratch.mit.edu>

Le guide *Programmation créative en Scratch* a été créé par Harvard pour vous orienter pendant l'enseignement de Scratch en classe et comprend des plans de cours :

<http://scratched.gse.harvard.edu/resources/informatique-cr%C3%A9ative> (version française)

[Http://scratched.gse.harvard.edu/guide](http://scratched.gse.harvard.edu/guide) (version anglaise)

La mission de *Code Club* est de donner à chaque enfant la chance d'apprendre à programmer. Il s'agit d'une banque de projets par étapes en langage Scratch, en HTML et en Python facilitant la création de clubs parascolaires. Il vous permet d'agrandir votre réseau de soutien et d'apprenants.

<http://codeclub.ca>

L'initiative *l'Heure de code* a généré beaucoup d'intérêt envers la programmation à l'échelle mondiale. Cette section du site est destinée aux enseignants de tous les niveaux souhaitant découvrir des techniques et des ressources pour enseigner la pensée informatique et l'informatique elle-même en classe.

<http://code.org/educate>