



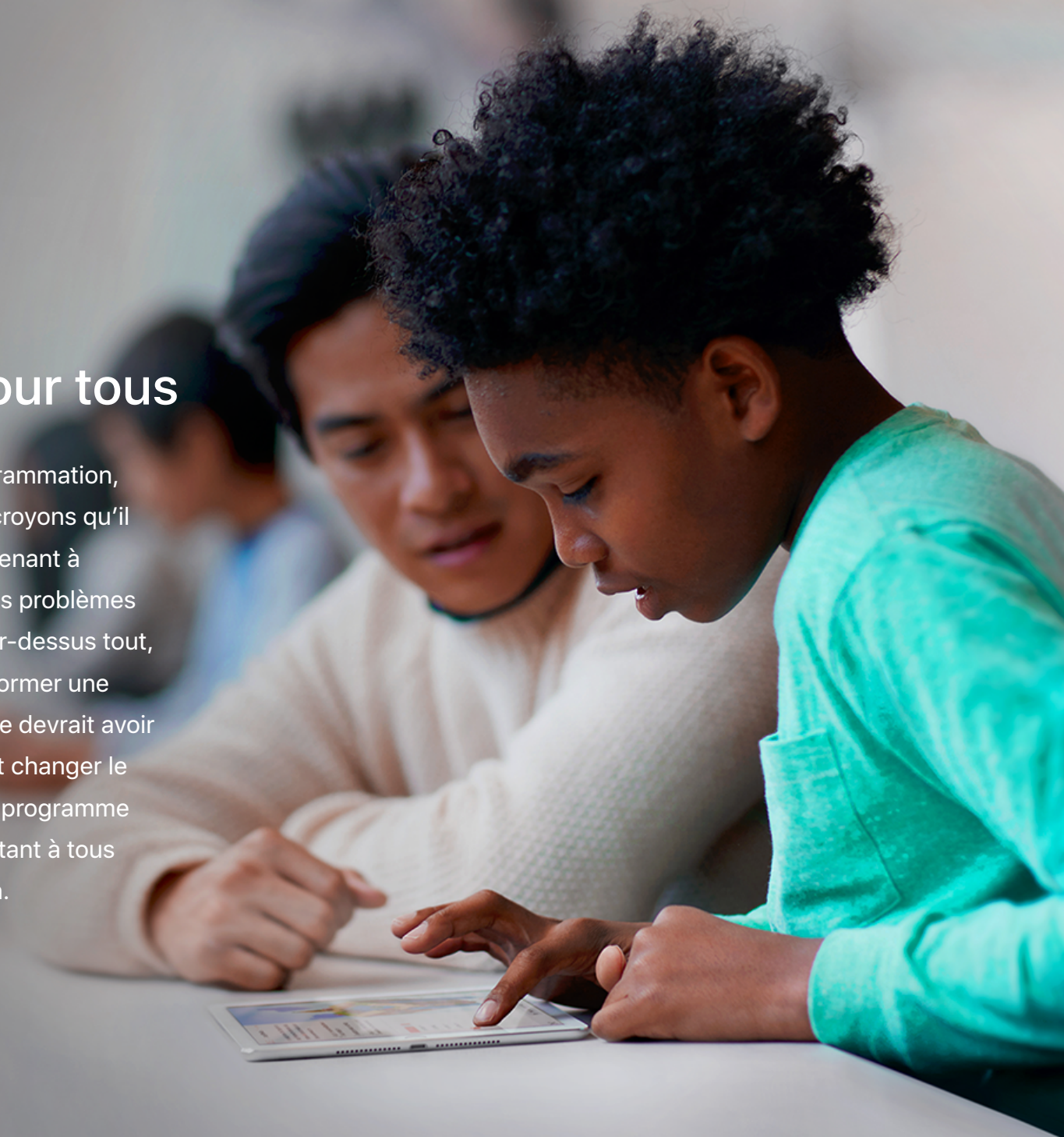
Guide de ressources Premiers pas en programmation

Mars 2018



















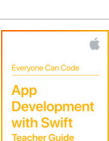


La programmation pour tous

La technologie possède un langage. La programmation, ou le code, est la façon de le parler. Et nous croyons qu'il s'agit d'une compétence essentielle. En apprenant à programmer, on apprend aussi à résoudre des problèmes et à travailler en équipe de façon créative. Par-dessus tout, on maîtrise les outils nécessaires pour transformer une simple idée en app. Selon nous, tout le monde devrait avoir la possibilité de créer quelque chose qui peut changer le monde. Nous avons donc conçu un nouveau programme qui offre des outils et des ressources permettant à tous d'apprendre et d'enseigner la programmation.



Ressources La programmation pour tous

Les ressources La programmation pour tous guident les élèves de leurs premiers pas en programmation jusqu'à la création de leur toute première app. Le tableau ci-dessous donne un aperçu des outils éducatifs qui vous sont offerts gratuitement.

Cours	Appareil	Public cible	App	Prérequis	Aperçu	Matériel pédagogique	Nombre d'heures de cours
		De la maternelle à la 2 ^e année	 	Aucun	Commencer à penser comme des programmeurs grâce à des activités pratiques sur des concepts clés dans des apps de programmation visuelle.	<ul style="list-style-type: none"> Leçons de l'app codeSpark Academy Cours Cadet de l'espace de l'app Tynker 	30 heures, y compris les leçons du guide de l'enseignant et des apps
		De la 3 ^e à la 5 ^e année		Aucun	Apprivoiser des concepts de programmation fondamentaux et s'entraîner à penser comme des programmeurs.	<ul style="list-style-type: none"> Cours Sorts de dragon de l'app Tynker 	36 heures, y compris les leçons du guide de l'enseignant et de l'app
		Dès l'école intermédiaire ou secondaire		Aucun	Apprendre les bases de la programmation en utilisant le code Swift.	<ul style="list-style-type: none"> App Swift Playgrounds Leçons d'Apprendre à coder 1 et 2 Cours iTunes U* Badges Swift Playgrounds de l'Apple Teacher Learning Center 	Jusqu'à 85 heures, y compris les leçons d'Apprendre à coder 1 et 2 (app et guide de l'enseignant)
		Dès l'école intermédiaire ou secondaire		Apprendre à coder 1 et 2	Renforcer les compétences de programmation et apprendre à penser comme des développeurs d'apps.	<ul style="list-style-type: none"> App Swift Playgrounds Leçons d'Apprendre à coder 3 	Jusqu'à 45 heures, y compris les leçons d'Apprendre à coder 3 (app et guide de l'enseignant)
		Secondaire, cégep et université		Aucun	Acquérir une expérience pratique des outils, techniques et concepts nécessaires pour concevoir de A à Z une app iOS simple.	<ul style="list-style-type: none"> Introduction au développement d'apps avec Swift – Guide de l'élève (et fichiers de projets) Introduction au développement d'apps avec Swift – Guide de l'enseignant 	90 heures
		Secondaire, cégep et université		Aucun	Apprendre les bases de Swift, d'UIKit et du réseautage grâce à des ateliers pratiques et des projets dirigés. Les élèves seront en mesure de concevoir leur propre app à la fin du cours.	<ul style="list-style-type: none"> App Development with Swift – Student Guide (et fichiers de projets)* App Development with Swift – Teacher Guide* 	180 heures

* En anglais.

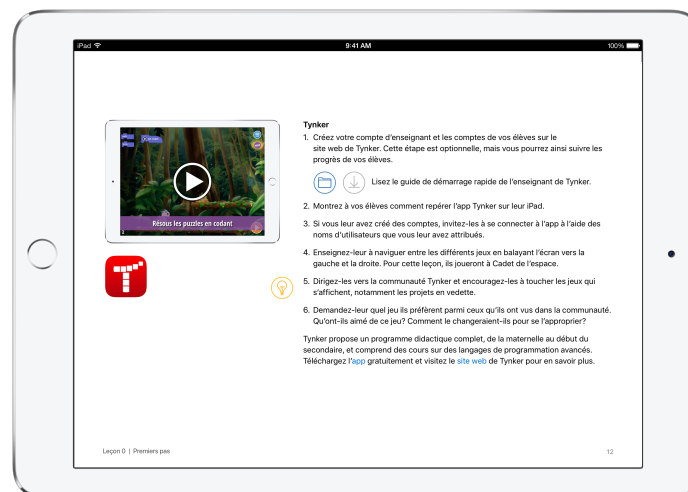
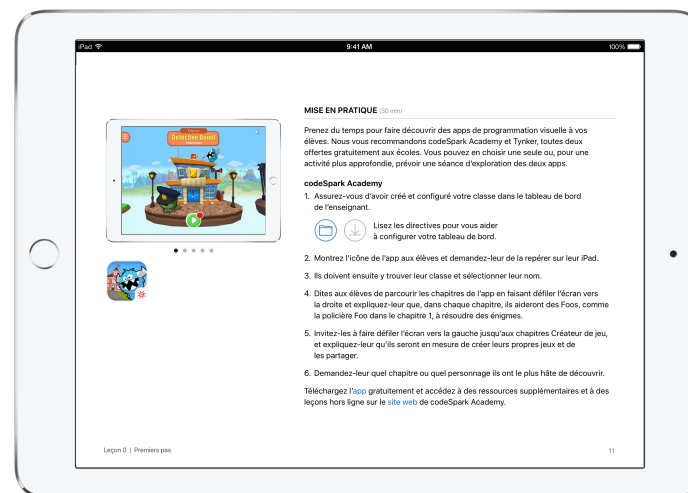
Aperçu

Les premières années d'école constituent un bon moment pour s'initier à la programmation. Ses concepts amènent les élèves à envisager le quotidien et le monde numérique sous un autre angle, et leur permettent d'acquérir les bases de la pensée informatique. Les apps de programmation spécialement conçues pour les jeunes élèves, comme codeSpark Academy et Tynker, proposent des puzzles visuels pour encourager la persévérance, la créativité et le développement d'habiletés en résolution de problèmes. Destinée aux élèves de cinq à sept ans, codeSpark Academy comporte une interface sans texte, ce qui permet aux enfants ne savant pas encore lire, aux apprenants de l'anglais et aux élèves ayant des difficultés en lecture de jouer aussi. Dans Tynker, les élèves de cinq à onze ans explorent d'abord des blocs de code visuels, puis progressent vers la programmation par texte alors qu'ils résolvent des puzzles et conçoivent des projets.

En classe

Les apps Tynker et codeSpark Academy, de même que les leçons des guides de l'enseignant Premiers pas en programmation, sont conçues pour vous aider à enseigner la programmation dès les premières années du primaire. Les leçons présentent des concepts clés de la programmation, tout en illustrant en quoi le mode de pensée qui les sous-tend s'applique à d'autres sphères de l'apprentissage et du quotidien.

Ces guides vous donnent tout le soutien nécessaire pour aider vos élèves à résoudre les puzzles, quelle que soit votre expérience en programmation. Exercices connexes, activités de conception d'app, questions de réflexion, idées de journal de bord, grille d'évaluation et plus encore vous aideront à favoriser chez vos élèves une compréhension approfondie de la matière. Vous pouvez enseigner les leçons dans leur intégralité ou par sections. En annexe, des tables de corrélation indiquent la conformité préliminaire des leçons avec les normes du programme Conception, compétences pratiques et technologies de la Colombie-Britannique (maternelle-5^e année) et les normes intérimaires d'enseignement informatique (maternelle-fin secondaire) établies par la Computer Science Teachers Association (CSTA) pour le niveau 1.



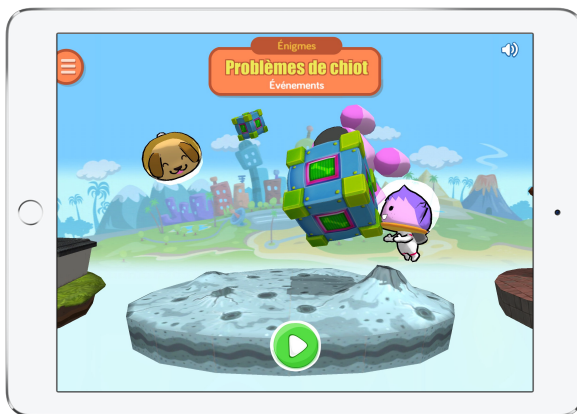
Principales caractéristiques

codeSpark Academy

Apprentissage. Dans ce jeu sans texte, les élèves résolvent des puzzles pour s’initier à des concepts de base de l’informatique, comme le séquençement, les boucles, les instructions conditionnelles et plus encore.

Création. Les élèves mettent leurs connaissances en pratique en créant leurs propres projets de programmation dans le Créateur de jeu.

Autonomie. Aucune expérience en programmation n’est requise pour enseigner, apprendre ou jouer. Les leçons et le tableau de bord de l’enseignant, qui comprend les rapports des progrès des élèves, sont offerts gratuitement aux éducateurs, dans 10 langues différentes.



Tynker

Environnement de programmation. À leur propre rythme, les élèves résolvent des puzzles de programmation structurés pour assimiler les concepts et les mettre en pratique de façon créative.



Évaluation automatique. Le tableau de bord de l’enseignant analyse les puzzles, les quiz et les exercices de programmation réalisés par les élèves afin d’évaluer leur maîtrise des compétences.

Fonctionnalité Swift. Les élèves peuvent passer des blocs de code visuels aux blocs de code Swift dans les puzzles; ils se familiarisent avec Swift et se préparent aux niveaux de programmation plus avancés.

Guides de l’enseignant

Fichiers téléchargeables. Des modèles de documents pour les activités des élèves et des présentations Keynote favorisent l’apprentissage en classe.

Corrigés. Les solutions aux puzzles de Tynker et codeSpark Academy vous permettent d’aider les élèves qui sont bloqués.

Exemples de travaux d’élèves. Voyez les résultats attendus des activités.

Réflexions. Les questions et sujets de réflexion fournis vous aident à renforcer le lien entre l’application du concept en programmation et dans la vie de tous les jours.

Astuces et exemples. Des idées pour approfondir et simplifier les leçons sont incluses.

Activités de conception d’app. Ces exercices guident les élèves dans un processus de conception où ils imaginent et prototypent une app qui résoudrait un problème dans leur classe ou leur école.

Structure des cours

Premiers pas en programmation 1

Grâce à des activités pratiques et interactives où ils appliquent des concepts de programmation à des situations du quotidien, les élèves commencent à penser comme des programmeurs. Ils découvrent les commandes, les séquences, les boucles, les événements et les algorithmes. Ensemble, ils s'exercent à prédire le résultat du code et à déboguer leur propre code et celui des autres. Ils utilisent les compétences acquises dans les apps de programmation visuelle, résolvent des puzzles et conçoivent des programmes. Et dans les exercices de conception facultatifs, ils suivent un processus où ils imaginent et prototypent une app qui résoudrait un problème dans leur classe ou leur école.

Leçon 0 – Premiers pas. Vous évaluez les connaissances des élèves en programmation, leur présentez les principales apps dont ils se serviront au fil des leçons, et organisez le tableau d'apprentissage de la classe. Vous leur présentez aussi les différents rôles qu'occupent les membres d'une équipe de conception d'app.

Leçon 1 – Mettre de l'ordre : Introduction au séquençement. Les élèves découvrent les séquences à l'aide d'exemples du quotidien, créent une séquence en s'inspirant d'une histoire connue, et se servent de séquences simples pour résoudre des puzzles dans des apps de programmation visuelle. Ils explorent aussi les diverses utilités des apps.

Leçon 2 – Procéder par étapes : La création de séquences. En comprenant l'importance de l'ordre des étapes dans une séquence, les élèves apprennent que les mêmes actions peuvent être placées dans un autre ordre pour former une séquence différente, puis ils créent une drôle de danse. Ils découvrent de nouvelles commandes et résolvent des problèmes plus complexes dans les apps de programmation. Ils comparent aussi différentes apps qui aident les utilisateurs à comprendre de nouveaux concepts.

Leçons 3 – Faire des choix : Le séquençement modulable. Les élèves apprennent que l'ordre de certaines étapes d'une séquence peut être modifié, puis ils créent leurs propres séquences modulables. Ils découvrent qu'ils peuvent résoudre un puzzle de plusieurs façons et partagent leurs solutions avec leurs camarades.

Ils explorent également des apps qui facilitent la communication entre les membres d'une communauté.

Leçon 4 – Répéter à l'infini : Les boucles. Les élèves trouvent des boucles dans des situations du quotidien, puis les appliquent de façon créative au cours d'une activité de percussions corporelles. Ils voient comment les boucles sont représentées en programmation et s'en servent pour simplifier l'écriture de leur code. Ils en apprennent plus sur la conception d'interface, qui permet de rendre une app amusante et facile à utiliser.

Leçon 5 – Corriger les erreurs : Le débogage. En saisissant l'importance de la persévérance et du débogage dans divers contextes, les élèves évaluent et utilisent leurs nouvelles compétences de programmation pour résoudre un problème et déboguer les solutions de leurs camarades. Ils s'exercent à prédire le résultat du code qu'ils écrivent et à y repérer les bogues quand il ne fonctionne pas comme prévu. Et ils se penchent sur la conception d'apps qui permettent de résoudre un problème.

Leçon 6 – Déclencher le processus : Les événements et les actions. Les élèves découvrent comment les événements rendent leur expérience de jeu et leur code plus dynamiques et réactifs, et apprennent à s'en servir en programmation. Ils réfléchissent aux différentes façons de déclencher des événements au quotidien, puis créent une télécommande de robot pour s'exercer à appeler des événements. Ils commencent aussi à concevoir leurs propres apps.

Leçon 7 – Respecter les conditions : Les instructions *if*. Les élèves sont initiés aux instructions conditionnelles et réfléchissent à la place des instructions *if* dans leur quotidien. Ils repèrent celles-ci dans des règles de jeux de société connus. Ils programment en utilisant des instructions *if* et créent du code qui réagit davantage aux conditions imposées par l'environnement. Ils illustrent ensuite le fonctionnement de leurs apps à l'aide d'organigrammes.

Leçon 8 – Résoudre les problèmes : Les algorithmes. Les élèves consolident leurs acquis et conçoivent des algorithmes composés de séries d'étapes leur permettant de résoudre des problèmes. Ils commencent par comprendre le processus d'une recette simple, puis créent et programment leur propre labyrinthe interactif. Ils conçoivent des prototypes de leurs apps et réalisent leur projet final, qui présente leur processus de conception.

Structures des cours (suite)

Premiers pas en programmation 2

Au fil des leçons, les élèves apprivoisent des concepts de programmation fondamentaux et s'entraînent à penser comme des programmeurs. En plus de découvrir les algorithmes, les fonctions, les boucles, les instructions conditionnelles et les variables, ils se familiarisent avec les bases de la conception d'interface. Seuls et en équipe, ils renforcent leurs compétences de programmation en résolvant de véritables problèmes, en testant le code de leurs camarades et en concevant des programmes pour différents robots. Ils développent aussi leurs habiletés dans Tynker en trouvant des solutions à des problèmes variés et en appliquant les concepts qu'ils ont appris au cours des activités. Et dans les exercices de conception facultatifs, ils suivent un processus où ils imaginent et prototypent une app qui résoudrait un problème dans leur classe ou leur école.

Leçon 0 – Premiers pas. Vous évaluez les connaissances des élèves en programmation, leur présentez une app de programmation visuelle comme Tynker, et configurez leurs portfolios numériques au moyen d'une app comme Seesaw. Vous leur proposez également un défi de conception d'une app.

Leçon 1 – Penser en étapes : La résolution de problèmes avec des algorithmes. Les élèves apprennent qu'un algorithme est une suite de commandes permettant de résoudre un problème ou d'effectuer une tâche. Dans l'app Tynker, ils choisissent un dragon, puis créent des algorithmes pour résoudre des puzzles et ainsi développer leurs capacités de séquençement. Ils conçoivent et testent des algorithmes au cours d'activités en classe, et ils commencent à trouver des idées d'apps pouvant résoudre un problème.

Leçon 2 – Penser aux solutions : Le débogage. Les élèves apprennent à détecter et à corriger des erreurs dans leurs algorithmes et leur code. Dans Tynker, ils modifient les algorithmes qui comportent des bogues pour créer un programme permettant de résoudre les puzzles. Ils se familiarisent aussi avec le rôle du clavier dans une app et pensent aux façons de l'utiliser dans leurs propres concepts.

Leçon 3 – Penser en rond : Les boucles. Les élèves apprennent à considérer les boucles comme des patrons, puis conçoivent et testent un algorithme pour créer un serpent bouclé. Dans Tynker, ils utilisent les boucles *for* pour résoudre des puzzles en repérant des patrons. Ils réfléchissent ensuite aux utilisations possibles de la caméra et du microphone dans leur app.

Leçon 4 – Penser en segments : La composition et la décomposition. Pour concevoir un algorithme qui permettra de jouer la mélodie du gobelet, les élèves décomposent les mouvements en plusieurs segments. Dans Tynker, ils résolvent des problèmes en les décomposant en sous-problèmes. Ils explorent également l'écran tactile pour rendre leur app plus interactive.

Leçon 5 – Penser aux ensembles : L'abstraction. Les élèves découvrent les notions de similarité et de généralisation tandis qu'ils classent des éléments dans des ensembles, puis ils expliquent leur raisonnement. Dans Tynker, ils font appel à l'abstraction pour repérer les similitudes entre les problèmes, et résolvent des puzzles de plus en plus complexes grâce à leurs nouveaux outils de programmation. Ils pensent aussi à l'utilisation de fonctionnalités comme Bluetooth pour se connecter à d'autres appareils à proximité.

Leçon 6 – Penser aux patrons : Les fonctions. Les élèves créent une série de mouvements pour un robot-assistant et la décomposent en fonctions; ils échangent ensuite leurs algorithmes pour comparer les résultats prévus et réels. Dans Tynker, ils utilisent des noms de fonctions et appellent celles-ci en réutilisant des séries d'instructions pour optimiser leur code. Puis, ils se familiarisent avec les données auxquelles leur app pourrait se connecter par GPS.

Structures des cours (suite)

Leçon 7 – Penser en détail : Les instructions conditionnelles. Les élèves définissent les conditions de leur voyage de rêve virtuel en utilisant un ensemble de critères sous forme d'instructions *if*. Dans Tynker, ils utilisent les instructions *if* pour gérer les décisions à prendre et les solutions de rechange dans les puzzles. Ils trouvent également des idées novatrices pour rendre leur app unique.

Leçon 8 – Penser en cycles : Les boucles *while* et emboîtées. Responsables d'une boutique de beignes virtuelle, les élèves utilisent les boucles *while* et emboîtées pour créer des algorithmes afin que le robot glace des beignes pour chaque client. Dans Tynker, ils utilisent les boucles pour raccourcir leur code. Ils forment ensuite des équipes de conception et commencent à créer le prototype de leur propre app.

Leçon 9 – Penser aux changements : Les variables, les entrées et les sorties. Dans l'activité Jam poétique, les élèves utilisent des variables pour concevoir l'algorithme d'une chanson ou d'un rap, qu'ils interprètent en intégrant des données fournies par la classe (des entrées). Dans Tynker, ils utilisent des variables pour s'attaquer à des puzzles plus complexes, mettant à profit les compétences de programmation qu'ils ont acquises. Ils réalisent aussi des entrevues qui les aident à définir le public cible de leur app.

Leçon 10 – Penser en pratique : La conception d'une interface utilisateur. Les élèves analysent les caractéristiques d'un bon concept, puis créent un pictogramme pour leur école. Dans l'activité de Tynker, ils utilisent toutes leurs nouvelles habiletés et terminent ainsi les leçons de Premiers pas en programmation 2. Puis, ils en apprennent plus sur l'interface et l'expérience utilisateur, et créent un tableau d'ambiance pour leur app. Pour leur projet final, ils présentent leur concept.

Renseignements supplémentaires

Ressources Premiers pas en programmation pour les élèves de la maternelle à la 5^e année

- [Tynker](#)
- [codeSpark Academy](#)
- [Premiers pas en programmation 1](#)
- [Premiers pas en programmation 2](#)

Ressources Swift Playgrounds pour les élèves de la 6^e à la 9^e année (ou 3^e secondaire)

- [Learn to Code 1&2 – Cours iTunes U*](#)
- [Apprendre à coder 1 et 2 – Guide de l’enseignant](#)
- [Apprendre à coder 3 – Guide de l’enseignant](#)
- [App Swift Playgrounds](#)

Guides sur le développement d’applications avec Swift pour les élèves du secondaire et du postsecondaire

- [Introduction au développement d’applications avec Swift](#)
- [Introduction au développement d’applications avec Swift – Guide de l’enseignant](#)
- [App Development with Swift*](#)
- [App Development with Swift – Teacher Guide*](#)

Ressources complémentaires

- Apprenez-en plus sur les ressources [La programmation pour tous](#).
- Apprenez-en plus sur [Swift](#).
- Apprenez-en plus sur [Xcode](#).
- Prenez part à la conversation sur le [forum des développeurs d’Apple*](#).
- Apprenez-en plus sur [codeSpark Academy*](#).
- Apprenez-en plus sur [Tynker*](#).

* En anglais.

Corrélation avec le curriculum : Premiers pas en programmation 1

Voici un aperçu de la conformité du guide Premiers pas en programmation 1 avec les normes du programme Conception, compétences pratiques et technologies de la Colombie-Britannique (maternelle-3^e année).

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 1 TABLE DE CORRÉLATION	Premiers pas en programmation 1 – C.-B. – Maternelle-2 ^e année		
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Grandes idées		
	Les concepts découlent de la curiosité naturelle.	Les compétences peuvent être développées par le jeu.	Les technologies sont des outils qui permettent d'augmenter les capacités humaines.
	●	●	●

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 1 TABLE DE CORRÉLATION	Premiers pas en programmation 1 – C.-B. – Maternelle-2 ^e année		
	Comprendre : Grandes idées	Faire : Compétences essentielles	
	Compétences essentielles		Compétences disciplinaires
	Compétence de communication	Compétence de réflexion	Compétence personnelle et sociale
	La compétence de communication englobe l'ensemble des aptitudes que les élèves utilisent pour transmettre et échanger de l'information, des expériences et des idées, pour explorer le monde qui les entoure et pour comprendre et utiliser de manière efficace les médias numériques.	La compétence de réflexion englobe les connaissances, aptitudes et processus associés au développement intellectuel. Elle permet aux élèves de prendre un contenu propre à une matière et de le transformer pour parvenir à une nouvelle compréhension. Elle englobe des aptitudes de réflexion précises, des habitudes intellectuelles et une conscience métacognitive.	La compétence personnelle et sociale est l'ensemble des habiletés associées à l'identité de l'élève dans le monde, tant sur le plan individuel que sur le plan collectif. Elle englobe les aptitudes que les élèves doivent maîtriser pour se réaliser personnellement, pour se comprendre eux-mêmes et comprendre les autres, pour prendre soin d'eux-mêmes et d'autrui, et pour définir et exercer leur rôle dans la société.
	●	●	●

Corrélation avec le curriculum (maternelle-3^e année) : Premiers pas en programmation 1 (suite)

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 1 TABLE DE CORRÉLATION	Premiers pas en programmation 1 – C.-B. – Maternelle-2 ^e année		
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Compétences essentielles		Compétences disciplinaires
Autres compétences disciplinaires	Concevoir des idées	Réaliser	Présenter
<p>Ce que l'élève doit pouvoir accomplir :</p> <ul style="list-style-type: none"> • Utiliser les matériaux, les outils et les technologies de manière sécuritaire, tant dans des environnements physiques que numériques. • Développer ses compétences pratiques et en acquérir de nouvelles par le jeu et le travail coopératif. • Étudier la possibilité d'utiliser des outils et des technologies simples afin d'accroître ses capacités. 	<ul style="list-style-type: none"> • Par l'exploration, déceler des besoins et des occasions de conception appliquée. • Formuler des idées à partir de ses expériences vécues et de ses intérêts. • Développer les idées des autres. • Choisir une idée à développer. 	<ul style="list-style-type: none"> • Choisir des outils et des matériaux. • Réaliser un produit en se servant de méthodes connues, ou en prenant exemple sur d'autres. • Employer l'approche essais-erreurs pour apporter des modifications, résoudre des problèmes, ou incorporer ses nouvelles idées ou celles d'autres personnes. 	<ul style="list-style-type: none"> • Déterminer comment et à qui présenter son produit. • Faire une démonstration de son produit, en relater la conception et la réalisation, et expliquer en quoi il est utile à l'individu, à la famille, à la communauté ou à l'environnement. • S'appuyer sur ses préférences personnelles pour évaluer le succès de son concept. • Réfléchir sur sa capacité à travailler efficacement, seul comme en équipe.
○	●	◐	●

Corrélation avec le curriculum : Premiers pas en programmation 2

Voici un aperçu de la conformité du guide Premiers pas en programmation 2 avec les normes du programme Conception, compétences pratiques et technologies de la Colombie-Britannique (4^e-5^e année).

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année		
	3 ^e année		4 ^e -5 ^e année
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Grandes idées		
	Les concepts découlent de la curiosité naturelle.	Les compétences peuvent être développées par le jeu.	Les technologies sont des outils qui permettent d'augmenter les capacités humaines.
	●	●	●

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année		
	3 ^e année		4 ^e -5 ^e année
	Comprendre : Grandes idées	Faire : Compétences essentielles	
	Compétences essentielles		Compétences disciplinaires
	Compétence de communication	Compétence de réflexion	Compétence personnelle et sociale
	La compétence de communication englobe l'ensemble des aptitudes que les élèves utilisent pour transmettre et échanger de l'information, des expériences et des idées, pour explorer le monde qui les entoure et pour comprendre et utiliser de manière efficace les médias numériques.	La compétence de réflexion englobe les connaissances, aptitudes et processus associés au développement intellectuel. Elle permet aux élèves de prendre un contenu propre à une matière et de le transformer pour parvenir à une nouvelle compréhension. Elle englobe des aptitudes de réflexion précises, des habitudes intellectuelles et une conscience métacognitive.	La compétence personnelle et sociale est l'ensemble des habiletés associées à l'identité de l'élève dans le monde, tant sur le plan individuel que sur le plan collectif. Elle englobe les aptitudes que les élèves doivent maîtriser pour se réaliser personnellement, pour se comprendre eux-mêmes et comprendre les autres, pour prendre soin d'eux-mêmes et d'autrui, et pour définir et exercer leur rôle dans la société.
	●	●	●

Corrélation avec le curriculum (4^e-5^e année) : Premiers pas en programmation 2 (suite)

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année		
	3 ^e année		4 ^e -5 ^e année
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Compétences essentielles		Compétences disciplinaires
Autres compétences disciplinaires	Concevoir des idées	Réaliser	Présenter
<p>Ce que l'élève doit pouvoir accomplir :</p> <ul style="list-style-type: none"> Utiliser les matériaux, les outils et les technologies de manière sécuritaire dans des environnements physiques et numériques. Développer ses compétences pratiques et en acquérir de nouvelles par le jeu et le travail coopératif. Étudier la possibilité d'utiliser des outils et des technologies simples afin d'accroître ses capacités. 	<ul style="list-style-type: none"> Par l'exploration, déceler des besoins et des occasions de conception appliquée. Tirer des idées de ses expériences vécues et de ses champs d'intérêt. Développer les idées des autres. Choisir une idée à développer. 	<ul style="list-style-type: none"> Choisir des outils et des matériaux. Réaliser un produit en se servant de méthodes connues, ou en prenant exemple sur d'autres. Employer l'approche essais-erreurs pour apporter des modifications, résoudre des problèmes, ou incorporer ses nouvelles idées ou celles d'autres personnes. 	<ul style="list-style-type: none"> Déterminer comment et à qui présenter son produit. Évaluer son produit en fonction de critères précis et expliquer en quoi il est utile sur les plans individuel, familial, collectif ou écologique. S'appuyer sur ses préférences personnelles pour évaluer le succès de son concept. Réfléchir sur son aptitude à travailler individuellement et en équipe.
	○	●	◐

Corrélation avec le curriculum (4^e-5^e année) : Premiers pas en programmation 2 (suite)

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année		
	3 ^e année		4 ^e -5 ^e année
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Grandes idées		
	Un concept peut être amélioré à l'aide d'un prototype et d'essais.	Le développement des compétences repose sur la pratique, l'effort et l'action.	Le choix de la technologie et des outils à employer dépend de la nature de la tâche.
	●	●	●

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année		
	3 ^e année		4 ^e -5 ^e année
	Comprendre : Grandes idées		Faire : Compétences essentielles
	Compétences essentielles		Compétences disciplinaires
	Compétence de communication	Compétence de réflexion	Compétence personnelle et sociale
	La compétence de communication englobe l'ensemble des aptitudes que les élèves utilisent pour transmettre et échanger de l'information, des expériences et des idées, pour explorer le monde qui les entoure et pour comprendre et utiliser de manière efficace les médias numériques.	La compétence de réflexion englobe les connaissances, les compétences et les processus associés au développement intellectuel. C'est grâce à leur compétence de réflexion que les élèves peuvent prendre un contenu propre à une matière et le transformer pour parvenir à une nouvelle compréhension. La compétence de réflexion comprend des habiletés de réflexion précises, des habitudes intellectuelles et la métacognition.	La compétence personnelle et sociale est l'ensemble des habiletés associées à l'identité de l'élève dans le monde, tant sur le plan individuel que sur le plan collectif. Elle englobe les aptitudes que les élèves doivent maîtriser pour se réaliser personnellement, pour se comprendre eux-mêmes et comprendre les autres, pour prendre soin d'eux-mêmes et d'autrui, et pour définir et exercer leur rôle dans la société.
	●	●	●

Corrélation avec le curriculum (4^e-5^e année) : Premiers pas en programmation 2 (suite)

Ministère de l'Éducation de la Colombie-Britannique Premiers pas en programmation 2 TABLE DE CORRÉLATION	Premiers pas en programmation 2 – C.-B. – 3 ^e -5 ^e année						
	3 ^e année			4 ^e -5 ^e année			
	Comprendre : Grandes idées			Faire : Compétences essentielles			
	Compétences essentielles			Compétences disciplinaires			
Autres compétences disciplinaires	Comprendre le contexte	Définir	Concevoir des idées	Assembler un prototype	Mettre à l'essai	Réaliser	Présenter
<p>Ce que l'élève doit pouvoir accomplir :</p> <ul style="list-style-type: none"> • Démontrer une connaissance des mesures de précaution à prendre et des consignes de sécurité à respecter en cas d'urgence, tant dans des environnements physiques que numériques. • Décrire et évaluer la nature et le niveau des compétences pratiques requises, individuellement ou en groupe, par rapport à une tâche précise; les développer au besoin. • Sélectionner, en se documentant au besoin, les outils et les technologies susceptibles d'accroître ses capacités à accomplir une tâche. • Décrire les conséquences sur le plan personnel, social et environnemental, y compris les conséquences négatives involontaires, de ses choix technologiques. • Expliquer comment des 	<ul style="list-style-type: none"> • Recueillir des renseignements sur les utilisateurs potentiels ou auprès de ceux-ci. 	<ul style="list-style-type: none"> • Choisir une possibilité de conception. • Identifier les caractéristiques principales ou les utilisateurs et leurs besoins. • Fixer l'objectif principal du travail de conception et indiquer toute contrainte existante. 	<ul style="list-style-type: none"> • Formuler des idées et développer les idées des autres. • Sélectionner les idées en fonction de l'objectif et des contraintes. • Choisir une idée à développer. 	<ul style="list-style-type: none"> • Tracer les grandes lignes d'un plan général indiquant les outils et les matériaux à utiliser. • Réaliser une première version du produit et, au besoin, changer les outils, les matériaux et la méthode employés. • Consigner la réalisation des versions successives du prototype. 	<ul style="list-style-type: none"> • Faire l'essai du produit. • Recueillir auprès de ses pairs une rétroaction et des idées. • Apporter des modifications et faire des essais jusqu'à ce qu'il soit satisfait du produit. 	<ul style="list-style-type: none"> • Réaliser la version finale du produit, avec les modifications prévues dans le plan. 	<ul style="list-style-type: none"> • Déterminer comment et à qui présenter son produit. • Faire une démonstration de son produit et en décrire le processus suivi. • Déterminer si son produit satisfait à l'objectif fixé et est utile à l'individu, à la famille, à la communauté ou à l'environnement • Réfléchir sur sa pensée et son processus de conception, et sur sa
○	●	●	●	●	●	◐	●

Corrélation avec le curriculum : Premiers pas en programmation 1

Les leçons de Premiers pas en programmation 1 sont conformes au concept Algorithmes et programmation des normes intérimaires d'enseignement informatique (maternelle-fin secondaire) établies en 2016 par la CSTA pour le niveau 1, de la maternelle à la 2^e année.

Conformité du cours Premiers pas en programmation 1 – Normes d'enseignement de la CSTA (maternelle-fin secondaire) – Niveau 1 (maternelle-2 ^e année)								
Norme de la CSTA	1A-A-7-1 Indication de sources	1A-A-5-2 Création de programmes	1A-A-5-3 Production de documents	1A-A-4-4 Représentation de données	1A-A-3-5 Décomposition	1A-A-3-6 Catégorisation d'éléments	1A-A-3-7 Algorithmes	1A-A-6-8 Analyse et débogage
Conformité globale		●	●	●	●	●	●	●
Mettre de l'ordre : Introduction au séquençement		●	●	●	●	●	●	●
Procéder par étapes : La création de séquences		●	●	●	●	●	●	●
Faire des choix : Le séquençement modulable		●	●	●	●	●	●	●
Répéter à l'infini : Les boucles		●	●	●	●	●	●	●
Corriger les erreurs : Le débogage		●	●	●	●		●	●
Déclencher le code : Les événements et les actions		●	●	●	●		●	●
Respecter les conditions : Les instructions <i>if</i>		●	●	●	●		●	●
Résoudre les problèmes : Les algorithmes		●	●	●	●		●	●

Légende : ● Conformité globale ● Conforme à la norme

Corrélation avec le curriculum : Premiers pas en programmation 2

Les leçons de Premiers pas en programmation 2 sont conformes au concept Algorithmes et programmation des normes intérimaires d'enseignement informatique (maternelle-fin secondaire) établies en 2016 par la CSTA pour le niveau 1, de la 3^e à la 5^e année.

Conformité du cours Premiers pas en programmation 2 – Normes d'enseignement de la CSTA (maternelle-fin secondaire) – Niveau 1 (3 ^e -5 ^e année)								
Norme de la CSTA	1B-A-2-1 Stratégies de collaboration	1B-A-7-2 Indication de sources	1B-A-5-3 Planification	1B-A-5-4 Création de programmes	1B-A-5-5 Opérations mathématiques	1B-A-3-6 Décomposition	1B-A-3-7 Algorithmes	1B-A-6-8 Analyse et débogage
Conformité globale	●	●	●	●	●	●	●	●
Penser en étapes : La résolution de problèmes avec des algorithmes	●		●	●		●	●	●
Penser aux solutions : Le débogage	●	●	●	●		●	●	●
Penser en rond : Les boucles	●		●	●		●	●	●
Penser en segments : La composition et la décomposition	●		●	●		●	●	●
Penser aux ensembles : L'abstraction	●		●	●			●	●
Penser aux patrons : Les fonctions	●	●	●	●		●	●	●
Penser en détail : Les instructions conditionnelles	●	●	●	●		●	●	●
Penser en cycles : Les boucles <i>while</i> et emboîtées	●		●	●	●	●	●	●
Penser aux changements : Les variables, les entrées et les sorties	●		●	●		●	●	●
Penser en pratique : La conception d'une interface utilisateur	●	●	●	●		●		●

Légende : ● Conformité globale ● Conforme à la norme

Les fonctionnalités peuvent changer. Certaines fonctionnalités peuvent ne pas être offertes dans certaines régions ou certaines langues.