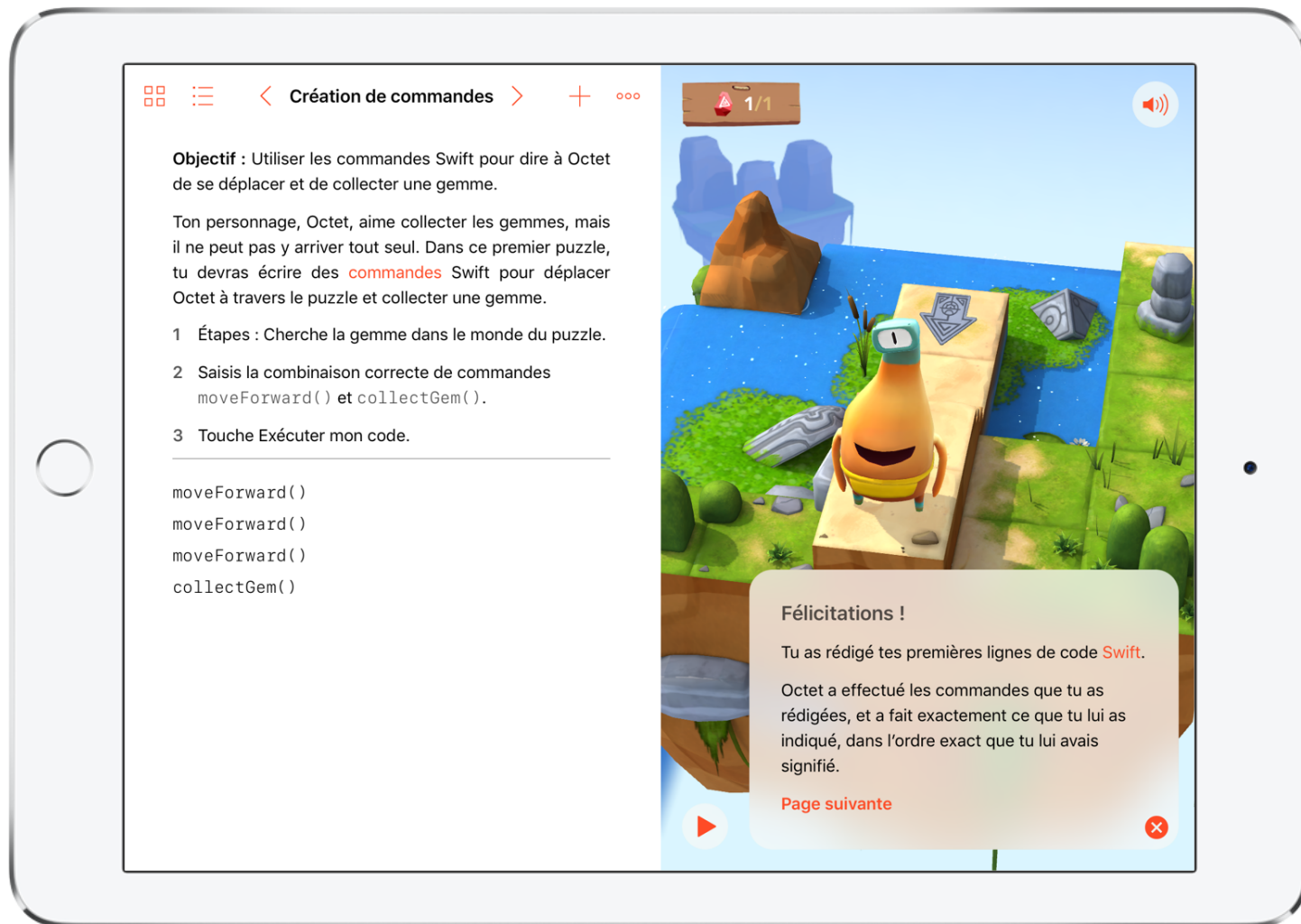




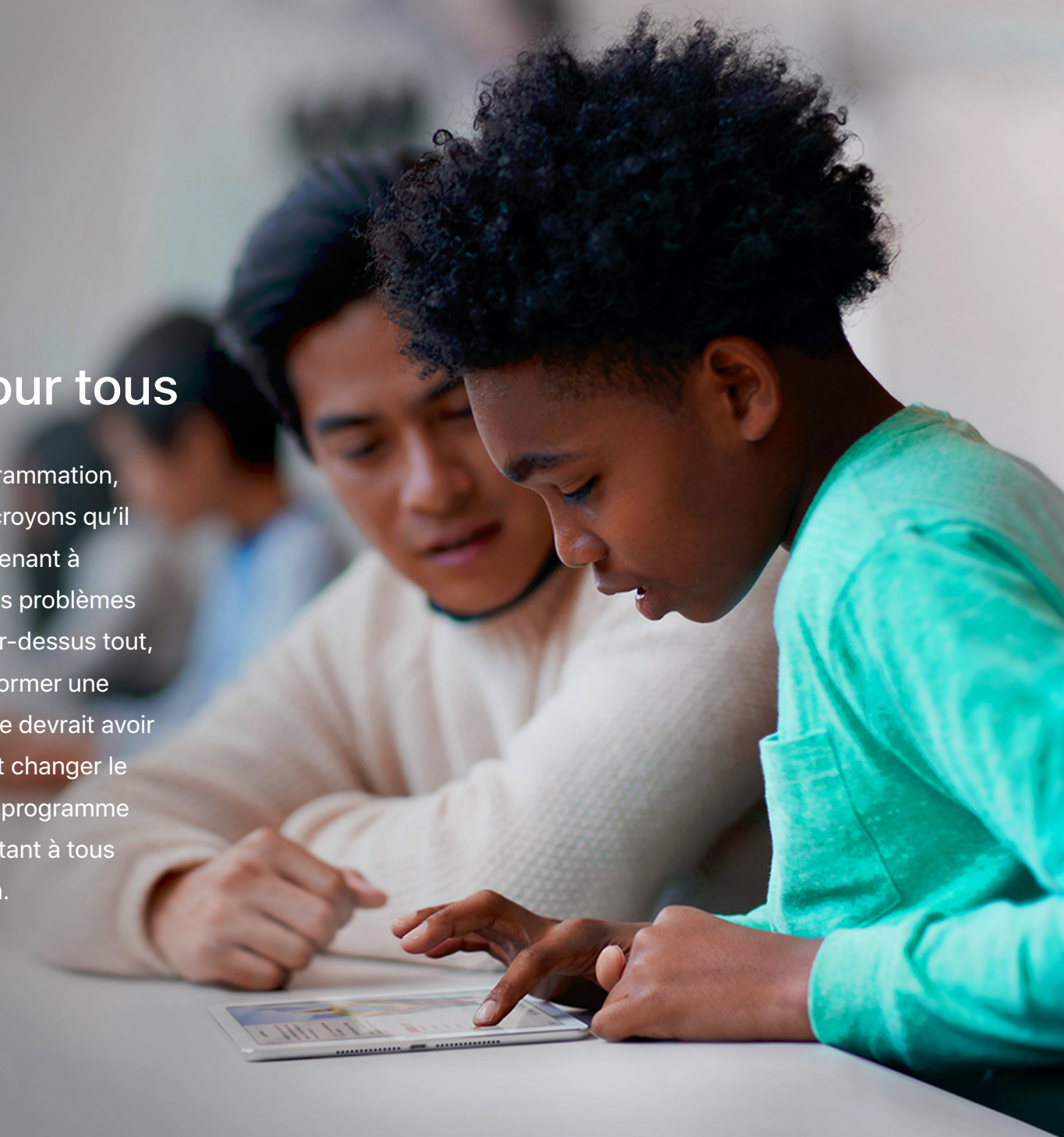
Guide de ressources Swift Playgrounds

Mai 2017



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.



La programmation pour tous

La programmation pour tous comprend une multitude de ressources qui guident les élèves, de leurs premiers pas en programmation à la création de leur toute première app. Le tableau ci-dessous donne un aperçu des ressources éducatives qui vous sont offertes gratuitement.

Cours	Appareil	Public cible	App	Prérequis	Aperçu	Matériel pédagogique	Ressources d'aide	Nombre d'heures de cours
 La programmation pour tous Premiers pas en programmation 1 Guide de l'enseignant		De la maternelle à la 2 ^e année		Aucun	Commencer à penser comme les programmeurs grâce à des activités pratiques sur des concepts clés dans des apps de programmation visuelle.	- Leçons de l'app codeSpark Academy - Cours Cadet de l'espace de l'app Tynker	Premiers pas en programmation 1 – Guide de l'enseignant	20 heures, y compris les activités du guide de l'enseignant et les leçons de l'app
 La programmation pour tous Premiers pas en programmation 2 Guide de l'enseignant		De la 3 ^e à la 5 ^e année		Aucun	Apprivoiser des concepts de programmation fondamentaux et s'entraîner à penser comme les programmeurs.	Cours Sorts de dragon de l'app Tynker	Premiers pas en programmation 2 – Guide de l'enseignant	20 heures, y compris les activités du guide de l'enseignant et les leçons de l'app
 La programmation pour tous Apprendre à coder 1 et 2 Guide de l'enseignant		Dès l'école intermédiaire ou secondaire		Aucun	Apprendre les bases de la programmation en utilisant le code Swift.	- App Swift Playgrounds - Leçons Apprendre à coder 1 et 2 - Cours iTunes U*	- Apprendre à coder 1 et 2 – Guide de l'enseignant - Badges Apple Teacher Learning Center – Swift Playgrounds	45 heures, y compris les activités du guide de l'enseignant et les leçons Apprendre à coder 1 et 2
 La programmation pour tous Introduction au développement d'apps avec Swift		Dès l'école secondaire		Aucun	Acquérir une expérience pratique des outils, techniques et concepts nécessaires pour concevoir de A à Z une app iOS simple.	Livre Introduction au développement d'apps avec Swift et fichiers de projets	Introduction au développement d'apps avec Swift – Guide de l'enseignant	90 heures

* En anglais seulement.

Aperçu

Swift Playgrounds est une app gratuite pour iPad qui enseigne les rudiments de la programmation de façon amusante et interactive. Les élèves peuvent résoudre des puzzles pour maîtriser les bases du codage avec Swift, un puissant langage de programmation Apple qu'utilisent les professionnels pour créer bon nombre d'applications populaires.

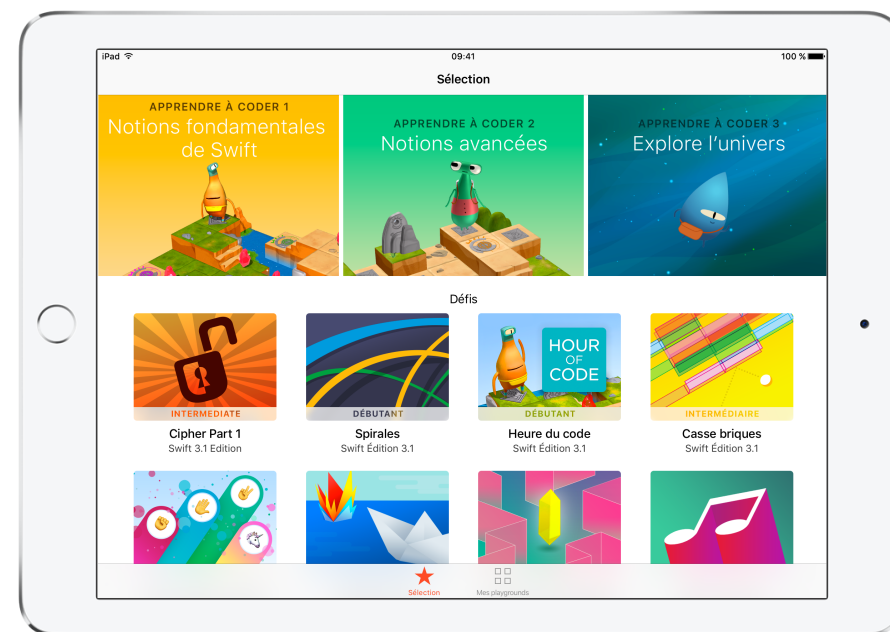
L'app propose également Apprendre à coder, une gamme complète de leçons spécialement conçues par Apple. À l'aide du vrai code Swift, les élèves sont amenés à résoudre des puzzles et à faire bouger des personnages d'un simple toucher. En explorant et en réglant une série de puzzles complexes, ils acquièrent des compétences qui servent de fondement à leur parcours en programmation.

Avec Apprendre à coder 1 et 2, les élèves s'initient à des concepts tels que commandes, débogage, fonctions, boucles, algorithmes, et plus encore. Les leçons sont spécialement conçues pour les débutants et ne nécessitent aucune expérience préalable.

En classe

Tout comme les activités du guide de l'enseignant, les leçons Apprendre à coder 1 et 2 de l'app ont été conçues pour les élèves dès l'école intermédiaire ou secondaire. Leur contenu flexible s'adaptant à tous les environnements pédagogiques, elles peuvent servir de cours ou de complément de formation. Elles sont conçues pour des périodes de 45 à 60 minutes, et certaines s'étalent sur plusieurs séances. Le temps suggéré pour chaque section est indiqué dans le guide. Si votre cours possède une structure particulière (s'il s'agit d'une activité parascolaire, par exemple), vous pouvez subdiviser les leçons.

Le guide peut être utilisé par tous les enseignants, que leur expérience en programmation soit vaste ou limitée.

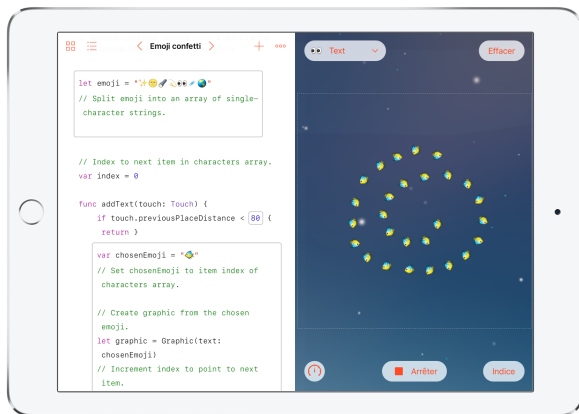


Swift Playgrounds comprend des leçons de programmation intégrées et des défis qui peuvent être relevés de façon autonome.

Principales caractéristiques

Vrai code Swift pour iOS. L'app Swift Playgrounds est basée sur Swift, le langage de programmation de nombreuses apps populaires offertes sur l'App Store. Les compétences qu'elle permet aux élèves d'acquérir touchent non seulement la programmation d'apps, mais une foule d'autres domaines.

Environnement interactif. Écrivez des lignes de code sur la partie gauche de votre écran et, d'un simple toucher, voyez instantanément le résultat du côté droit.



Accessibilité. Swift Playgrounds a été conçue pour tout le monde. Tirant parti des nombreuses fonctionnalités d'accessibilité d'iOS, dont le Contrôle de sélection et VoiceOver, elle permet aux élèves d'entendre une description des actions exécutées par les personnages qu'ils contrôlent en codant.

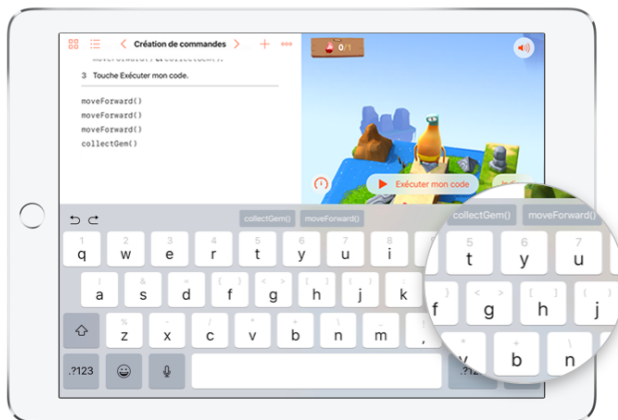
Animations captivantes. Chaque section commence par une animation captivante qui, en transposant des concepts de programmation dans la vie réelle, facilite la compréhension.

Glossaire intégré. Les définitions du glossaire aident les élèves à comprendre des termes techniques.

Conseils pratiques. À tout moment, les élèves peuvent obtenir de l'aide sous forme d'indices. Dans de nombreux cas, celles-ci changent de façon dynamique selon le code entré.

Barre de raccourcis. La barre de raccourcis QuickType affiche des suggestions au bas de l'écran, permettant ainsi aux élèves de coder en quelques touchers.

Clavier virtuel. Un clavier spécialement conçu pour Swift donne un accès rapide aux chiffres et symboles les plus utilisés.



Sauvegarde et partage. Les élèves peuvent sauvegarder leurs écrans pour montrer leur démarche.

Révision du code. En exécutant le code plus vite ou plus lentement, ou en parcourant les lignes de code pour les surligner en cours d'exécution, les élèves peuvent repérer plus aisément les erreurs potentielles.

Fonction Glisser-déposer. Glissez-déposez autour d'un code existant des structures complexes entourant d'autres codes, par exemple des boucles ou des fonctions. Il suffit de toucher un mot-clé (comme « for ») pour que la fonction glisser-déposer apparaisse à l'écran.



Modifications instantanées. Changez les valeurs numériques, les couleurs et les opérateurs de façon simple et rapide avec le pavé numérique contextuel.

Ressources d'aide

Apprendre à coder 1 et 2 – Guide de l'enseignant

Ce guide est conçu pour aider les enseignants à intégrer les leçons Apprendre à coder 1 et 2 en classe dès l'école intermédiaire ou secondaire. Il présente les concepts clés de la programmation, tout en montrant comment son mode de pensée peut s'appliquer à d'autres sujets et situations du quotidien. Il comprend des activités à faire en classe, des exercices de révision et de réflexion, des rubriques sur l'évaluation et des présentations Keynote.

Les leçons représentent environ 45 heures de cours, ce qui équivaut à un trimestre. En annexe, une table de corrélation indique la conformité préliminaire des leçons Apprendre à coder 1 et 2 avec les normes intérimaires d'enseignement informatique établies par la Computer Science Teachers Association (CSTA) pour le niveau 2.

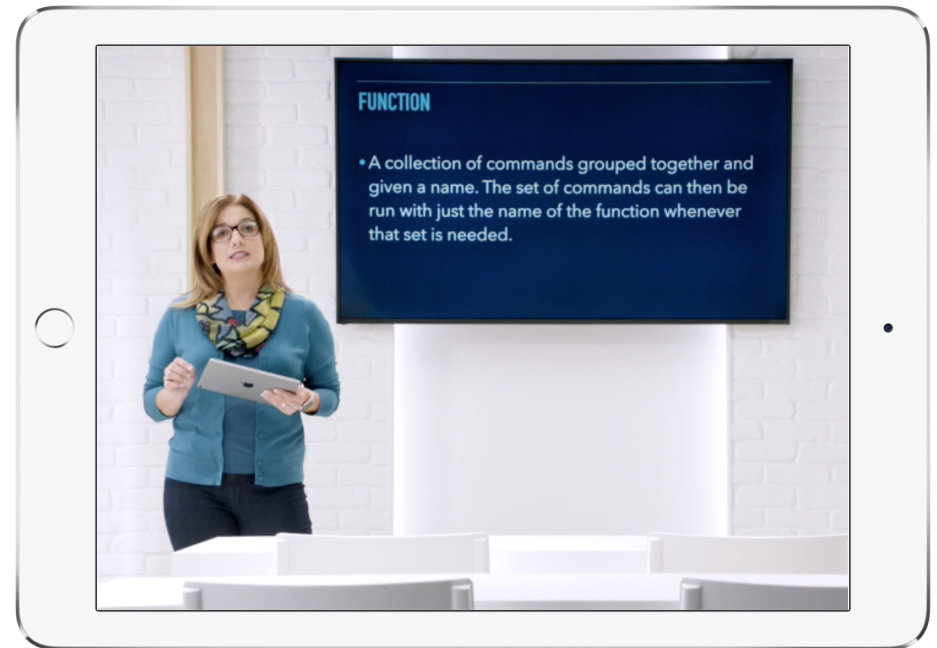
Learn to Code 1&2 – Cours iTunes U*

À l'aide de vidéos et d'autres ressources pertinentes, ce cours iTunes U bonifie le guide de l'enseignant Apprendre à coder 1 et 2. Grâce aux vidéos, les éducateurs disposent d'un modèle pour mettre en pratique les leçons du guide de l'enseignant.

Programme Apple Teacher – Badges Swift Playgrounds

Apple Teacher est un programme de développement professionnel gratuit conçu pour soutenir et valoriser les enseignants. Dans une perspective d'autoformation, il propose matériel didactique, conseils, idées et nouvelles. Les membres Apple Teacher ont accès à l'Apple Teacher Learning Center, où ils peuvent répondre à des quiz sur l'enseignement et l'apprentissage de Swift Playgrounds et obtenir quatre badges. Ceux qui réussissent reçoivent également le logo Apple Teacher – Swift Playgrounds pour afficher leur réalisation.

* En anglais seulement.



Structure des cours

Apprendre à coder 1

En résolvant des puzzles dans un monde 3D dynamique, les élèves peuvent acquérir des compétences de codage et bâtir leur vocabulaire de programmation. Dans un premier temps, ils explorent des commandes, des fonctions et des boucles simples. Comme de véritables programmeurs, ils sont appelés à utiliser le code Swift.

Leçon 0 – Premiers pas. Les élèves sont initiés à l'informatique et prennent connaissance des objectifs du cours.

Leçon 1 – Penser comme un ordinateur : Les commandes et les séquences. Les élèves apprennent à utiliser des commandes et des séquences dans une situation du quotidien, puis à les mettre en pratique pour programmer.

Leçon 2 – Penser comme un détective : Le débogage. Les élèves découvrent le débogage dans une situation du quotidien, puis l'appliquent en contexte de programmation.

Leçon 3 – Penser efficacement : Les fonctions et les boucles. Les élèves s'initient aux fonctions et aux boucles dans une situation du quotidien, puis les utilisent dans la programmation.

Révision et réflexion. Les élèves révisent les notions abordées au cours des trois premières leçons, revoient leurs portfolios et créent une communauté d'évaluation par les pairs.

Leçon 4 – Penser logiquement : Le code conditionnel. Les élèves apprennent à utiliser le code conditionnel, les booléens et les opérateurs logiques, puis les mettent en pratique dans la programmation.

Leçon 5 – Penser encore et encore : Les boucles *while*. Les élèves découvrent les boucles *while* dans une situation du quotidien, puis les utilisent en programmation.

Leçon 6 – Repenser une idée : Les algorithmes. Les élèves s'initient aux algorithmes dans une situation du quotidien, puis les appliquent en contexte de programmation.

Révision et réflexion. Les élèves révisent les concepts de programmation abordés de la leçon 3 à la leçon 6, poussent la réflexion sur leurs portfolios et poursuivent leur apprentissage collectif.

Apprendre à coder 2

Avec ce cours, les élèves approfondissent leur connaissance de Swift. Ils vont au-delà de la simple résolution de puzzles pour créer des mondes bien à eux. Ils découvrent les variables et les types – des éléments de programmation permettant de stocker l'information et d'y accéder – ainsi que le procédé d'initialisation et les paramètres. Avec les compétences acquises, les élèves peuvent mieux exploiter le code pour interagir avec le personnage et les puzzles, et ainsi modifier les règles qui régissent leur monde.

Leçon 7 – Penser comme un bot-journaliste : Les variables. Les élèves explorent les variables dans une situation du quotidien, puis s'en servent pour programmer.

Leçon 8 – Penser comme un architecte : Les types. Les élèves découvrent les types dans une situation du quotidien, puis les appliquent en programmation.

Leçon 9 – Penser avec précision : Les paramètres. Les élèves s'initient aux paramètres dans une situation du quotidien, puis les utilisent pour programmer.

Leçon 10 – Penser de façon ordonnée : Les tableaux. Les élèves apprennent à se servir de tableaux dans une situation du quotidien, puis à les employer en programmation.

Projet final. À l'aide des concepts assimilés au cours du programme, les élèves bâtissent un univers bien à eux et une histoire pour l'accompagner. Ils réfléchissent sur leurs acquis en révisant leurs portfolios et en faisant appel au système d'évaluation par les pairs.

Renseignements supplémentaires

Swift Playgrounds nécessite iOS 10 et fonctionne sur les modèles d'iPad suivants :

- iPad Pro (9,7 po)
- iPad Pro (12,9 po)
- iPad Air 2
- iPad Air
- iPad mini 4
- iPad mini 3
- iPad mini 2

Ressources pour l'app Swift Playgrounds

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

Guides Introduction au développement d'applications avec Swift

- [Introduction au développement d'applications avec Swift](#)
- [Introduction au développement d'applications avec Swift – Guide de l'enseignant](#)

Ressources pour les cours Premiers pas en programmation

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

Ressources complémentaires

- Apprenez-en plus sur [Swift Playgrounds](#).
- Apprenez-en plus sur le programme [La programmation pour tous](#).
- Apprenez-en plus sur [Swift](#).
- Prenez part à la conversation sur le [forum des développeurs d'Apple*](#) en cliquant sur l'icône « For Educators » .

* En anglais seulement.

Conformité avec le curriculum : Apprendre à coder 1 et 2

Voici un aperçu de la conformité préliminaire du guide de l'enseignant Apprendre à coder 1 et 2 avec les normes intérimaires d'enseignement informatique établies par la Computer Science Teachers Association (CSTA) pour le niveau 2. La CSTA fera une révision officielle du guide une fois que les nouvelles normes auront été fixées. La table de corrélation vise le concept Algorithmes et programmation, qui fait partie des normes intérimaires d'enseignement informatique établies par la CSTA en 2016 pour les élèves de la maternelle jusqu'à la fin du secondaire.

Grâce aux puzzles d'Apprendre à coder 1 et 2 et aux leçons du guide de l'enseignant, les élèves découvrent des concepts de programmation comme les fonctions, les algorithmes, les variables, les types, le code conditionnel, les boucles et les tableaux. Ils apprennent à repérer et à définir les problèmes computationnels en formulant des activités courantes de manière abstraite. Pour créer leurs propres artefacts, ils se servent d'abstractions pour décomposer un problème en solutions qu'ils expriment sous forme de fonctions et processus. Ils utilisent des algorithmes et prédisent leurs résultats, proposent des solutions, sollicitent les commentaires de leurs pairs et mettent à l'essai leur travail pour repérer les bogues et améliorer le résultat.

Normes d'enseignement de la CSTA – Niveau 2 (intermédiaire, fin primaire et début secondaire)										
Norme de la CSTA	2-A-2-1 Commentaires des pairs	2-A-7-2 Algorithmes : Comparaison	2-A-7-3 Attribution	2-A-7-4 Algorithmes : Interprétation	2-A-5-5 Artefacts computationnels	2-A-5-6 Développement de programmes	2-A-5-7 Variables	2-A-4-8 Paramètres	2-A-3-9 Décomposition	2-A-6-10 Conception itérative
Chapitres Swift Playgrounds	Commandes	●	●						●	●
	Fonctions	●	●		●	●		●	●	●
	Boucles <i>for</i>	●	●		●	●		●	●	●
	Code conditionnel	●	●		●	●		●	●	●
	Opérateurs logiques	●	●		●	●		●	●	
	Boucles <i>while</i>	●	●		●	●		●	●	●
	Algorithmes	●	●		●	●		●	●	●
	Variables	●	●		●	●	●	●	●	●
	Types	●	●		●	●	●	●	●	●
	Initialisation	●	●					●	●	
	Paramètres	●	●		●	●		●	●	
	Création du monde	●	●		●	●		●	●	
	Tableaux	●	●		●	●	●	●	●	●

Légende : ● Conforme à la norme

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