

Apple Programmer avec Swift

Guide de ressources

The image features a laptop screen displaying the Xcode IDE. The interface is split into three main areas: a project navigator on the left showing a file tree for a project named 'Travel', a central editor showing Swift code for a 'LoginViewController', and a preview window on the right showing a mobile app interface for a login screen. The code includes imports for UIKit, a custom notification class, and a class definition for 'LoginViewController' with various UI components and lifecycle methods. The preview window shows a mobile app with a suitcase and globe icon, the text 'Partons en voyage!', and input fields for email and password.

Overlaid in the foreground are four book covers:

- Programmer avec Swift** (orange cover)
- Introduction au développement d'applications avec Swift** (orange cover)
- Develop in Swift** (yellow cover)
- App Development with Swift** (yellow cover)

Novembre 2019

Enseigner la programmation avec Apple

En enseignant la programmation, vous ne faites pas que montrer aux élèves le langage de la technologie : vous leur donnez de nouvelles façons de penser et de concrétiser des idées. Et Swift, le langage de programmation simple, puissant et intuitif d'Apple, offre des moyens amusants et attrayants de se préparer pour l'avenir. Chaque élève devrait avoir la possibilité de créer quelque chose qui peut changer le monde. Apple fournit ce qu'il faut pour initier les élèves au code, qu'ils se familiarisent avec Swift Playgrounds sur iPad ou fassent leurs débuts dans Xcode sur Mac.



Ressources sur la programmation

Les ressources La programmation pour tous et Programmer avec Swift 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 principaux outils éducatifs qui vous sont offerts gratuitement.

Élève	Enseignant	Appareil	Public cible	App	Prérequis	Aperçu	Heures de cours	
La programmation pour tous								
La programmation pour tous – Puzzles				Dès la 4 ^e année	Swift Playgrounds	Aucun	Les élèves découvrent les concepts de base de la programmation, comme les boucles, les variables et les fonctions. Ils mettent à profit leurs compétences de diverses façons, notamment dans le cadre de puzzles et de playgrounds ouverts.	45 heures
La programmation pour tous – Aventures				Dès la 4 ^e année	Swift Playgrounds	La programmation pour tous – Puzzles	Les élèves se familiarisent avec des concepts plus poussés, comme la gestion d'événement, les tableaux avancés et la conception par composants, en planifiant et en réalisant leurs propres projets.	45 heures
Programmer avec Swift								
Introduction au développement d'apps avec Swift				Dès l'école secondaire	Xcode	Aucun	Les élèves acquièrent une expérience pratique avec les outils, les techniques et les concepts nécessaires pour concevoir de A à Z une app iOS de base.	90 heures
App Development with Swift (en anglais)				Dès l'école secondaire	Xcode	Aucun	Les élèves apprennent les bases de Swift, d'UIKit et de la réseautique grâce à des ateliers pratiques et à des projets dirigés. À la fin du cours, ils devraient être en mesure de concevoir leur propre app.	180 heures

Programmer avec Swift

En créant des apps avec Swift, les élèves réfléchissent à l'influence positive que la programmation peut avoir sur le monde. Les ressources Programmer avec Swift portent sur Swift, un langage de programmation en code source ouvert à la fois convivial et puissant compris dans l'environnement de programmation Xcode de Mac. C'est ce même langage qui est employé par les professionnels de l'industrie des apps, qui est en pleine expansion, pour concevoir des apps pour iOS, macOS, tvOS, watchOS et plus. Programmer avec Swift s'adresse tant aux élèves débutants qu'à ceux qui ont déjà de l'expérience en programmation. Les ressources peuvent même les aider à se préparer à une carrière en programmation grâce à une certification reconnue par l'industrie.



Aperçu des ressources

Programmer avec Swift enseigne aux élèves de niveau secondaire ou universitaire à concevoir et à bâtir eux-mêmes une app entièrement fonctionnelle. En plus de développer de nouvelles compétences, les élèves maîtrisent les concepts clés de la programmation et peuvent même obtenir une certification reconnue par l'industrie pour attester leur maîtrise de Swift et de Xcode. Les élèves certifiés reçoivent un badge numérique qu'ils peuvent afficher sur les réseaux professionnels pour témoigner de leurs aptitudes. Les guides de l'enseignant proposent des outils pour mobiliser les aspirants développeurs. Ils s'adressent à tous les éducateurs, qu'ils aient ou non déjà enseigné Swift ou d'autres langages de programmation. Les ressources complémentaires du Club de programmation Swift sont conçues pour les programmes d'enseignement parascolaires ou estivaux. Et les expositions d'apps donnent aux programmeurs l'occasion de célébrer leur ingéniosité avec la communauté, qu'ils apprennent en classe comme ailleurs.

Le cours Introduction au développement d'apps avec Swift apprend aux élèves à développer des apps en les initiant aux rudiments du langage Swift dans Xcode. Pour leur projet final, ils conçoivent l'une des deux apps iOS de base proposées.

Le cours App Development with Swift (en anglais seulement) fait suite au cours Introduction au développement d'apps avec Swift et permet aux élèves d'approfondir leurs connaissances. Les élèves qui sont à l'aise avec Swift, Xcode et le développement d'apps iOS peuvent parcourir les leçons rapidement ou passer directement aux ateliers pratiques pour bâtir des miniprojets et tester leur code dans les playgrounds. À la fin du cours, ils et elles seront en mesure de concevoir une app entièrement fonctionnelle. Et ils pourront obtenir une certification reconnue par l'industrie qui atteste leur maîtrise de Swift et de Xcode.

First App



Part 1 - New Project

Now that you're getting more comfortable with playgrounds, you might be wondering how to build an app you can use on your iOS device, or even your Apple Watch. A lot of moving parts need to work together to make an app run, and Xcode is the best tool for putting them all together.

In this three-part lesson, you'll build SinglePhoto—a simple iOS app that displays a single photo. In the first exercise, you'll create an app project from scratch. Then you'll use Xcode to explore your project and learn to navigate your coding environment.

You can customize every part of your app—from its icon on the Home screen to the way its buttons behave. There are panels and controls in Xcode that display the many options available to you. You'll practice using the Xcode Interface Builder to continue customizing your first app.

In the final step, you'll add an image to your project and edit the user interface. You'll also get an introduction to Interface Builder—a powerful component of Xcode where you create the user interface of your app. At the end of the second exercise, your app will look like this, but it will display a photo of your own choosing.

1.8 | First App 56

Lesson 1.8 Interface Builder Basics

The best way to learn the basics of Interface Builder is to dive into Xcode and explore some of its features. Start by creating a new iOS project using the Single View Application template. Name the project "IBBasics".

STORYBOARDS

Interface Builder opens whenever you select an XIB file (.xib) or a storyboard file (.storyboard) from the project navigator.

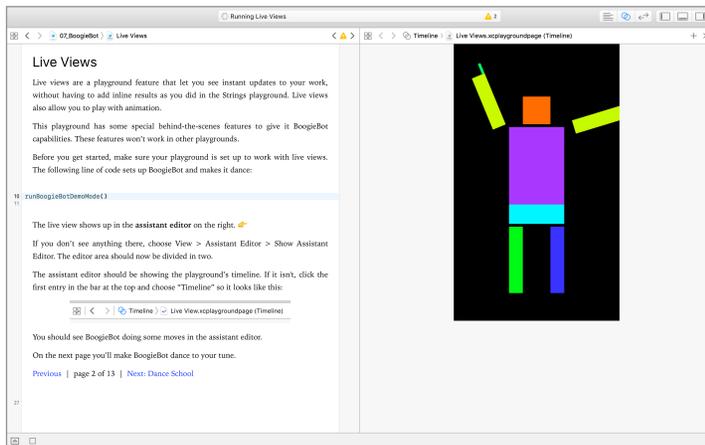
An XIB file contains the user interface for a single visual element, such as a full-screen view, a table view cell, or a custom UI control. XIBs were used more heavily before the introduction of storyboards. They're still a useful format in certain situations, but this lesson will focus on storyboards.

In contrast with an XIB, a storyboard file includes many pieces of the interface, defining the layout of one or many screens as well as the progression from one screen to another. As a developer, you'll find that the ability to see multiple screens at once will help you understand the flow within your app.

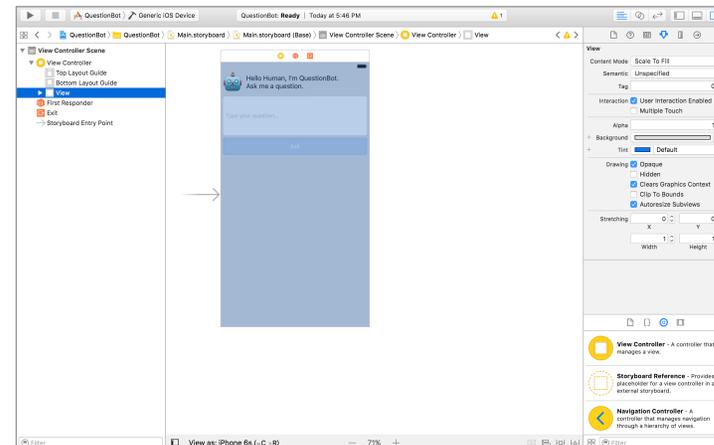


78 1.8 Interface Builder Basics | 79

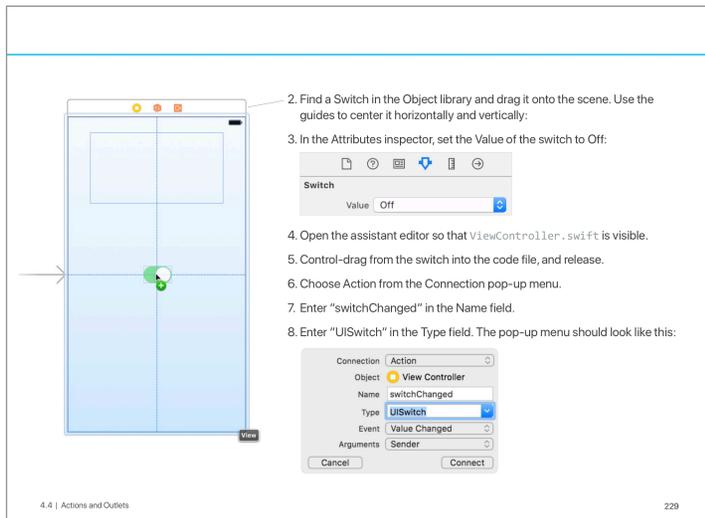
Principales caractéristiques



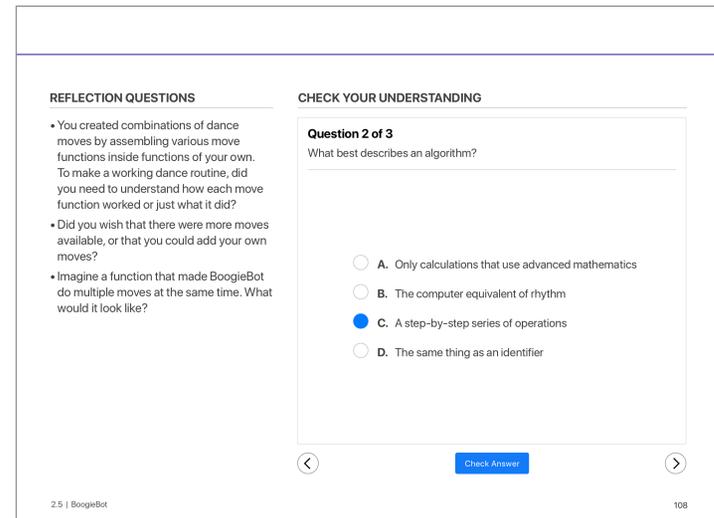
Playgrounds Xcode : Les élèves assimilent des notions de programmation en rédigeant du code dans des playgrounds Xcode, des terrains de jeu interactifs où ils font des essais de programmation et en voient les résultats immédiatement.



Exemples de projets : Les exemples de projets fournis permettent aux élèves de manipuler certaines parties du code sans avoir à créer l'app au complet.



Instructions pas à pas : Des instructions détaillées avec images et vidéos guident les élèves à travers le processus de création d'une app dans Xcode.



Outils d'apprentissage : Les élèves peuvent vérifier leur compréhension et réfléchir à leurs apprentissages à l'aide de questions de révision, de termes clés, de liens vers des ressources et plus encore.

Ressources Programmer avec Swift



Structure du cours Introduction au développement d'apps avec Swift

Ce cours d'introduction d'un trimestre (90 heures) apprend aux élèves les fondements essentiels de la programmation avec Swift. Les élèves acquièrent une expérience pratique des outils, techniques et concepts nécessaires pour concevoir une app iOS de base. Les leçons guident les élèves tout au long des étapes nécessaires à la conception d'une app : remue-méninge, planification, prototypage et évaluation. Même si les élèves n'ont pas encore les compétences requises pour bâtir une app véritable, les efforts qu'ils consacrent aux prototypes leur seront utiles à l'avenir.

Leçon 1 : Rudiments des playgrounds – Les élèves se familiarisent avec l'environnement interactif des playgrounds.

Leçon 2 : Désignation et identificateurs – Les élèves apprennent les bases de la résolution de problèmes en se servant des noms et des identificateurs appropriés.

Leçon 3 : Chaînes – Les élèves s'initient aux concepts de chaîne et d'interpolation de chaîne.

Leçon 4 : Hello, World! – Les élèves s'initient au monde de la programmation en apprenant à personnaliser leur environnement Xcode et à déboguer.

Leçon 5 : Première app – Les élèves créent leur toute première app avec Xcode et visualisent leur travail au moyen d'iOS Simulator.

Leçon 6 : Les fonctions – Les élèves découvrent toute la puissance des fonctions en formulant une définition universelle à partir d'étapes détaillées.

Leçon 7 : BoogieBot – Les élèves mettent en pratique leurs connaissances en programmation en contrôlant un robot-danseur animé dans le playground.

Leçon 8 : Constantes et variables – Les élèves approfondissent leur compréhension de la désignation grâce à une introduction aux concepts de constante et de variable.

Leçon 9 : Types – Les élèves se familiarisent davantage avec les fondements de Swift en explorant son système de types, de la bibliothèque standard de Swift aux types personnalisés.

Leçon 10 : Paramètres et résultats – Les élèves vont plus loin dans leur apprentissage en découvrant comment les paramètres et les valeurs de retour peuvent rendre les fonctions plus efficaces et flexibles.

APPLICATION ET EXPLORATION (25 min)	RÉVISION ET DISCUSSION (de 10 à 20 min)
<p>Au cours de cette leçon, les élèves apprendront à distinguer les quatre types de fonctions à l'aide de l'exemple d'un artiste qui crée une toile. Discutez des quatre types de fonctions :</p> <ol style="list-style-type: none">❌ Paramètres, ❌ Valeur de retour <code>peindreToile()</code> S'exécute seule et ne retourne aucune valeur.✅ Paramètres, ❌ Valeur de retour <code>peindreToile(largeur: Int, hauteur: Int, couleurDominante: UIColor)</code> Exécute une tâche qui change selon les arguments, mais ne retourne aucune valeur.❌ Paramètres, ✅ Valeur de retour <code>peindreToile() -> Toile</code> Ne requiert aucune information, mais retourne une valeur.✅ Paramètres, ✅ Valeur de retour <code>peindreToile(largeur: Int, hauteur: Int, couleurDominante: UIColor) -> Toile</code> Accepte de l'information et retourne une valeur.	<p>Posez et explorez les questions de réflexion du guide de l'élève :</p> <ul style="list-style-type: none">Vous savez maintenant programmer des fonctions capables d'accepter de l'information qu'elles utiliseront pour exécuter une tâche. Mais la tâche exécutée est toujours la même... Et si vous pouviez réaliser différentes tâches en fonction de l'information transmise?Pensez à des procédés ou à des tâches qui correspondent, dans la vie réelle, aux différentes définitions d'une fonction (avec ou sans paramètres, avec ou sans valeur de retour). Voici quelques exemples :<ul style="list-style-type: none"><code>func eteindreLeFour()</code> (Éteindre le four ne requiert aucun paramètre, et aucune valeur n'est retournée par cette action.)<code>func prechaufferLeFour(temperature: Int)</code> (Préchauffer le four requiert un paramètre indiquant la température à atteindre.)<code>func preparerDesBiscuits()</code> (Préparer des biscuits ne requiert aucun paramètre et produit un dessert.)<code>func cuireAuFour(ingredients: [Ingredients]) -> [Patisserie]</code> (Cuire au four requiert une liste d'ingrédients et retourne une délicieuse fournée de pâtisseries.)
Leçon 10 Paramètres et résultats	45

Le guide de l'enseignant comprend des activités connexes supplémentaires, des points de discussion et des activités pour le journal de conception d'apps que tiennent les élèves tout au long du cours.

Structure du cours Introduction au développement d'apps avec Swift (suite)

Leçon 11 : Prise de décisions – Les élèves apprennent à prendre des décisions en codant, au moyen d'instructions conditionnelles *if/else*, des valeurs *true* et *false*, et des opérateurs de comparaison.

Leçon 12 : Instances, méthodes et propriétés – Les élèves enrichissent leurs connaissances sur les types en apprenant comment les méthodes et les propriétés composent l'instance d'un type.

Leçon 13 : QuestionBot – Les élèves apprennent à modifier un projet Xcode existant en écrivant une nouvelle logique pour un robot qui répond à des questions.

Leçon 14 : Tableaux et boucles – Les élèves apprennent à créer des tableaux, puis à les utiliser en y ajoutant ou en y retirant des objets. Ils découvrent aussi comment les boucles interagissent avec chacun des objets d'un tableau.

Leçon 15 : Définition de structures – Les élèves prennent conscience de l'aspect pratique des types personnalisés pour regrouper des informations et fonctionnalités liées.

Leçon 16 : QuestionBot 2 – Les élèves poursuivent leur exploration de l'app QuestionBot en créant ChatBot, une app qui affiche l'historique des conversations. Ils apprennent ce qu'est un modèle de source de données et conçoivent un objet source de données simple qui fournit de l'information sur les objets Message devant être affichés dans la vue en liste. Et ils s'exercent à effectuer des ajouts à un tableau pour stocker des messages dans l'objet source de données afin de tenir un historique des conversations.

Leçon 17 : Actions et sorties – Les élèves apprennent à concevoir une interface utilisateur (IU) en se servant des outils d'Interface Builder et à relier les éléments d'interface au code au moyen d'actions et de sorties. Ils s'exercent à créer des sorties pour accéder aux propriétés d'une vue d'IU et des actions pour répondre aux interactions de l'utilisateur avec les boutons et les autres commandes.

Leçon 18 : Interfaces utilisateur dynamiques – Les élèves apprennent un processus reproductible permettant de créer une IU pour le plus petit format d'iPhone, qu'ils pourront ensuite adapter à tous les formats d'iPhone et à toutes les orientations d'écran. Ils découvrent Auto Layout, une fonctionnalité servant à définir des contraintes pour établir la position et la taille des éléments de l'IU. Ils utilisent aussi la vue en pile (Stack View), un objet conçu spécifiquement pour configurer automatiquement les contraintes d'Auto Layout grâce à des réglages plus simples et un système utilisant une grille. En suivant ce processus, ils créent les apps Centrage facile, Questionnaire sur les éléments et Cris d'animaux.

Leçon 19 : Énumérations et instruction switch – Les élèves s'initient aux énumérations (enums), qui permettent de définir une liste d'options nommée. Ils découvrent le rôle des énumérations, la façon de les définir et leurs utilisations courantes. Ils apprennent aussi à se servir de l'instruction switch pour exécuter conditionnellement du code basé sur la valeur d'une instance d'énumération.

Leçon 20 : Projet final – Les élèves montent un des projets finaux, ou les deux, en partant de zéro. Le premier est un jeu de roche-papier-ciseaux, et le deuxième est une machine à mèmes. Les élèves revoient divers concepts abordés dans le cours et bâtissent l'interface utilisateur, les données du modèle et les objets du contrôleur qui composent l'app.

Leçon 21 : Conception d'app – Les élèves suivent un cycle de conception axé sur le prototypage, tout comme le feraient de vrais développeurs.



Structure du cours App Development with Swift*

Ce cours de deux trimestres (180 heures) comporte 51 leçons, chacune portant sur une compétence liée à Swift ou au développement d'app. À la fin de chacune des cinq premières unités, les étudiants réalisent des projets dirigés, notamment la description de fonctionnalités axées sur l'utilisateur, l'élaboration d'un plan de projet et la définition d'instructions pour la conception et la création d'une app entièrement fonctionnelle. Dans le cadre de ces projets, ils conçoivent des fonctionnalités qui les intéressent, tout en effectuant le travail typique d'un environnement professionnel de développement d'app.

Unité 1 : Initiation au développement d'app – Les étudiants se familiarisent avec les notions de base liées aux données, aux opérateurs et aux flux de commandes dans Swift, ainsi qu'avec la documentation, le débogage, la création et l'exécution d'une app, Xcode et Interface Builder. Ils mettent ensuite leurs connaissances en pratique dans le projet dirigé Light, qui les amène à créer une app de lampe de poche toute simple.

Unité 2 : Introduction à UIKit – Les étudiants explorent les chaînes, les fonctions, les structures, les collections et les boucles Swift. Ils se familiarisent également avec UIKit – les affichages et les réglages système qui composent une interface utilisateur – et apprennent à afficher des données à l'aide d'Auto Layout et de la vue en pile. Ils mettent leurs compétences à profit dans le projet dirigé Apple Pie, où ils créent une app de devinettes de mots.

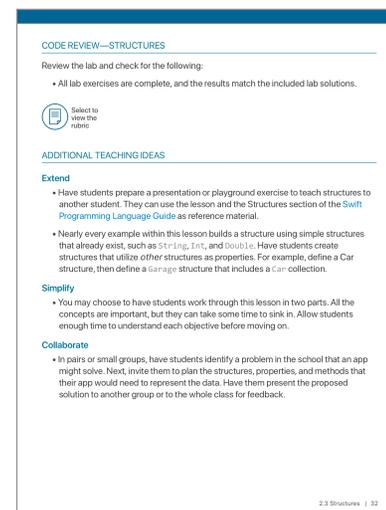
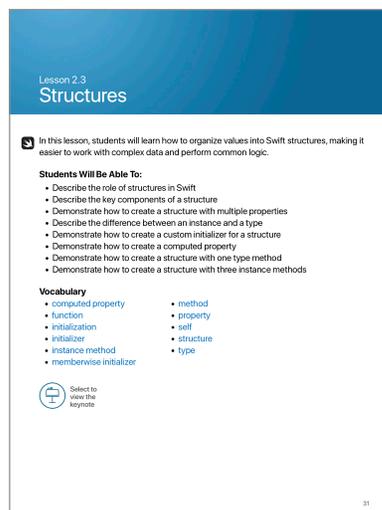
Unité 3 : Navigation et flux de travail – Les étudiants découvrent comment créer des flux de travail simples et des hiérarchies de navigation à l'aide de contrôleurs de navigation, de contrôleurs de la barre d'onglets et d'enchaînements. Ils examinent également deux puissants outils dans Swift : les optionnels et les énumérations. Ils mettent leurs connaissances en pratique dans le projet dirigé Personality Quiz, un sondage personnalisé qui révèle des réponses amusantes à l'utilisateur.

Unité 4 : Tableaux et persistance – Les étudiants découvrent les vues déroulantes, les vues tabulaires et la création d'écrans d'entrée de données complexes. Ils apprennent également à enregistrer des données, à partager celles-ci avec d'autres apps et à travailler avec des images tirées de la photothèque d'un utilisateur. Ils font ensuite appel à leurs habiletés dans le projet dirigé List, une app de suivi des tâches où l'utilisateur ajoute, modifie et supprime des éléments dans une interface familière sous forme de tableau. Ils personnalisent l'app de façon à garder une trace de tout type d'information, comme des collections, des tâches ou des listes de lecture.

* En anglais seulement.

Unité 5 : Utilisation du web – Les étudiants se familiarisent avec les animations, l'accès simultané et le travail sur le web. Ils mettent leurs acquis en pratique dans le projet Restaurant, une app personnalisable qui affiche le menu d'un restaurant et permet à l'utilisateur de passer une commande. Grâce à un service web, les étudiants peuvent configurer le menu en utilisant les plats et les photos de leur choix.

Unité 6 : Prototypage et planification de projets – Les étudiants apprennent à concevoir leur propre projet et à en développer le prototype et l'architecture. S'ils ont suffisamment de temps, ils devraient pouvoir le bâtir par eux-mêmes.



Le guide de l'enseignant présente des astuces pour approfondir ou adapter les leçons et offrir du soutien aux étudiants qui en ont besoin.

Ressources d'aide

Évaluations

Les guides de l'enseignant Programmer avec Swift proposent des évaluations par projet qui permettent d'observer les aptitudes de collaboration et de communication et l'esprit critique des étudiants, ainsi que d'évaluer leur code et leur documentation dans le cadre des tâches demandées.

Certification

Les éducateurs qui donnent le cours App Development with Swift peuvent aider les élèves à faire attester leur maîtrise de Swift et de Xcode. La réussite d'un examen administré par Certiport* permet d'obtenir la certification de niveau 1 pour le cours App Development with Swift et de continuer à progresser vers une carrière de développeur d'apps.

[En savoir plus sur la certification >](#)

Journal de bord de conception d'app

Avant même de commencer à programmer avec Swift, les élèves se penchent sur les types d'apps qu'ils aimeraient concevoir et créer. Le journal de bord les guide aux étapes du prototypage de leurs idées, de la mise à l'essai auprès de leurs pairs et de l'amélioration de l'expérience utilisateur.

Guide pour l'exposition d'apps

Célébrez l'ingéniosité des élèves. Encouragez-les à présenter leurs créations aux membres de leur communauté dans le cadre d'une exposition. Le Guide pour l'exposition d'apps fournit des conseils pratiques pour vous aider à planifier et à organiser un tel événement.

Club de programmation Swift

Invitez les élèves à poursuivre leur cheminement au-delà de la salle de classe au sein d'un club de programmation Swift. Ils relèveront ensemble des défis amusants allant de la programmation d'appareils connectés à la conception de leurs propres apps. Les guides de l'animateur fournissent les outils nécessaires pour aider chaque élève à faire preuve de créativité.

* D'autres modalités pourraient s'appliquer; consultez le site web de [Certiport](#) (en anglais) pour en savoir plus.



[Télécharger le Journal de bord de conception d'app >](#)



[Télécharger le Guide pour l'exposition d'apps >](#)



[Découvrir le Club de programmation Swift >](#)

Renseignements supplémentaires pour l'enseignement de la programmation

Ressources Swift Playgrounds

- [La programmation pour tous – Puzzles](#)
- [Guide de l'enseignant La programmation pour tous – Puzzles](#)
- [App Swift Playgrounds](#)

Guides sur le développement d'applications avec Swift

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

Ressources complémentaires

- [Apprenez-en plus sur les ressources d'Apple pour enseigner la programmation.](#)
- Prenez part à la conversation sur le [forum des développeurs d'Apple*](#).
- Obtenez la [certification de niveau 1 pour le cours App Development with Swift*](#).
- Importez App Development with Swift dans votre canevas.

À propos de Swift

Swift est le langage de programmation puissant et intuitif qu'Apple a créé pour développer des applications. En plus d'être idéal pour effectuer ses premiers pas en programmation, Swift est très polyvalent. Il a été pensé pour écrire le plus simple des programmes, comme « Hello, world! », aussi bien que le plus évolué des logiciels. Apprenez-en plus sur [Swift](#).

À propos de Xcode

Xcode est l'app Mac qui sert à créer toutes les apps pour Mac et iOS. Elle comprend tous les outils dont vous avez besoin pour créer des expériences logicielles hors du commun. Et elle est [téléchargeable gratuitement](#) sur le Mac App Store. Apprenez-en plus sur [Xcode](#).

Développement professionnel d'Apple

Le programme de développement professionnel d'Apple prend la forme de missions de plusieurs jours, sur une période donnée, qui visent à fournir :

- Des conseils en planification et en orientation du leadership pour aider les administrateurs à hiérarchiser les objectifs d'apprentissage.
- De l'encadrement de la part d'un spécialiste en développement professionnel d'Apple tout au long de votre projet.
- Des plans de développement professionnel personnalisés, fondés sur la recherche et adaptés à vos objectifs pédagogiques.
- Des expériences de développement immersives et concrètes pour permettre aux enseignants de repenser leurs pratiques pédagogiques et de motiver les élèves.
- Une occasion de collaborer avec des responsables du développement professionnel afin d'assurer une certaine stabilité.

Avec les spécialistes en développement professionnel d'Apple, vos enseignants profiteront d'un soutien sur mesure qui les aidera à repenser leurs pratiques pédagogiques de manière à susciter l'intérêt de leurs élèves en plus de leur offrir un apprentissage personnalisé.

Pour en savoir plus, communiquez avec votre responsable de compte Apple ou écrivez-nous à apls@apple.com.

* En anglais seulement.

Conformité avec les normes*

Le cours App Development with Swift de Programmer avec Swift s’inscrit dans les normes intérimaires d’enseignement informatique établies par la Computer Science Teachers Association (CSTA) pour le niveau 3A. La table de corrélation vise le concept Algorithmes et programmation, qui fait partie des normes établies par la CSTA en 2016 pour les élèves de la maternelle jusqu’à la fin du secondaire. La CSTA fera une révision officielle du cours App Development with Swift avant l’année scolaire 2020-2021.

CSTA K–12 Computer Science Standards Level 3A for Grades 9–10												
CSTA Standard	3A-A-2-1 Design Artifact	3A-A-2-2 Collaborating	3A-A-7-3 Licensing	3A-A-5-4 Respond to Event	3A-A-5-5 Research	3A-A-5-6 Mathematical Concepts	3A-A-4-7 Hierarchy & Abstraction	3A-A-4-8 Deconstruct Problem	3A-A-4-9 Abstraction	3A-A-3-10 Design Algorithms	3A-A-3-11 Modelling & Simulation	3A-A-6-12 Debugging
Playground Basics	●				●							●
Naming and Identifiers	●				●	●		●	●	●		●
Strings	●	●			●	●		●	●	●	●	
Hello, world!	●				●			●		●		●
First App	●			●	●			●				
Functions	●					●		●	●	●	●	
BoogieBot	●			●		●		●	●	●	●	
Constants and Variables	●					●		●	●	●	●	●
Types	●				●	●		●	●	●	●	●
Parameters and Results	●				●	●		●	●	●	●	
Making Decisions	●			●	●	●		●	●	●	●	
Instances, Methods, and Properties	●				●	●		●	●	●	●	
QuestionBot	●			●	●	●		●	●	●	●	●
Arrays and Loops	●				●	●		●	●	●	●	
Defining Structures	●				●	●		●	●	●	●	
QuestionBot 2	●			●	●	●		●	●	●	●	
Actions and Outlets	●			●	●	●		●		●		●
Adaptive User Interfaces	●			●	●	●		●		●		●
Enumerations and Switch	●	●			●	●		●	●	●	●	
Final Project	●	●		●	●	●		●	●	●	●	

* La table de corrélation est disponible en anglais seulement.



© 2019 Apple Inc. Tous droits réservés. Apple, le logo Apple, iPad, Mac et Xcode sont des marques de commerce d'Apple Inc., déposées aux États-Unis et dans d'autres pays. Swift, le logo Swift et Swift Playgrounds sont des marques de commerce d'Apple Inc. App Store est une marque de service d'Apple Inc., déposée aux États-Unis et dans d'autres pays. IOS est une marque de commerce ou une marque de commerce déposée de Cisco aux États-Unis et dans d'autres pays; elle est utilisée sous licence. Les autres produits et dénominations sociales mentionnés ici peuvent être des marques de commerce de leurs sociétés respectives. Les caractéristiques des produits peuvent changer sans préavis. Le présent document n'est fourni qu'à titre d'information; Apple se dégage de toute responsabilité quant à son utilisation. Novembre 2019