



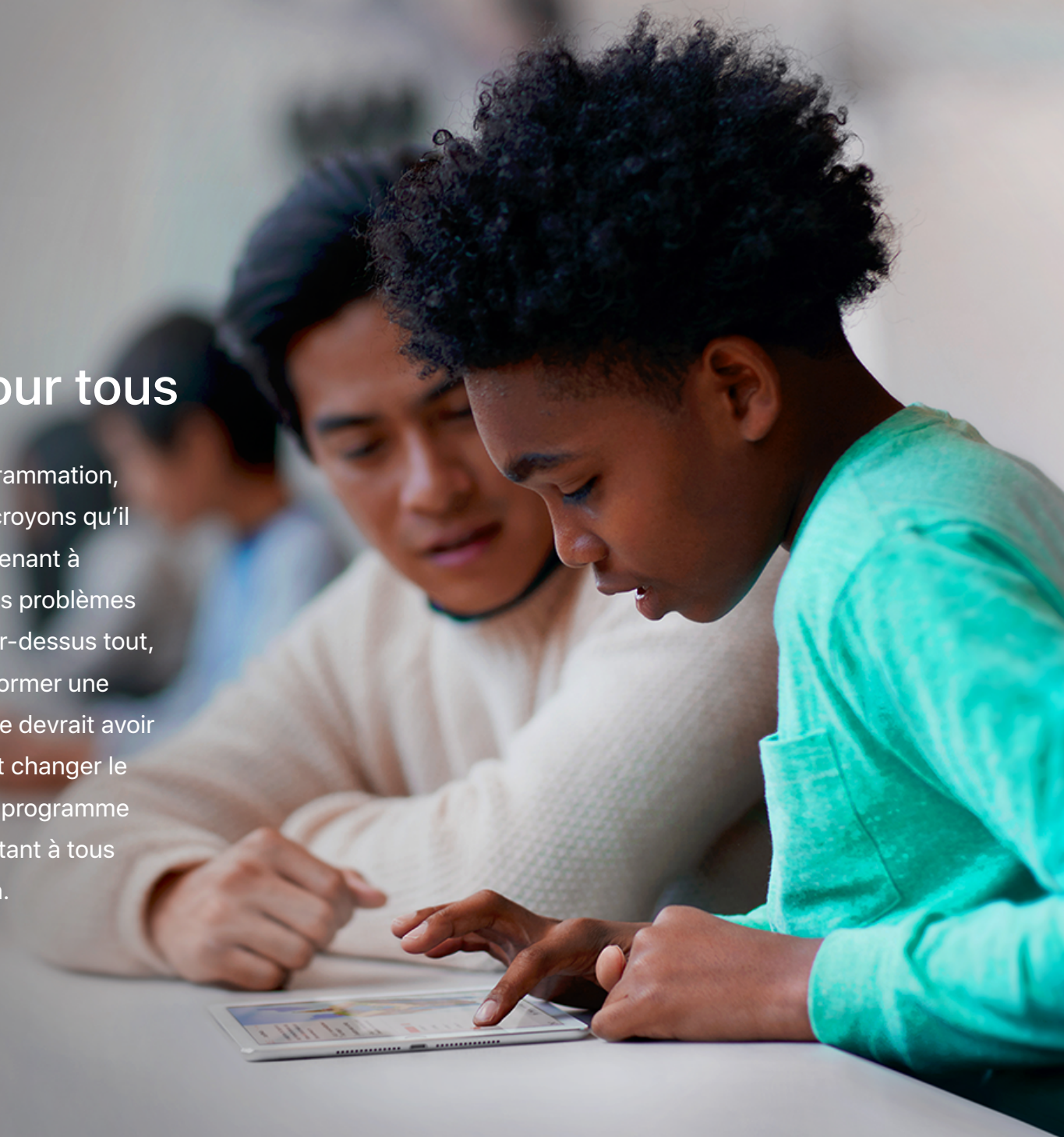
# Guide de ressources Développement d'apps avec Swift

Mars 2018



# La programmation pour tous

















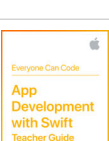


La technologie possède un langage. La programmation, ou le code, est la façon de le parler. Et nous croyons qu'il s'agit d'une compétence essentielle. En apprenant à programmer, on apprend aussi à résoudre des problèmes et à travailler en équipe de façon créative. Par-dessus tout, on maîtrise les outils 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
 <p>La programmation pour tous Premiers pas en programmation 1 Guide de l'enseignant</p>		De la maternelle à la 2 <sup>e</sup> 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"> <li>Leçons de l'app codeSpark Academy</li> <li>Cours Cadet de l'espace de l'app Tynker</li> </ul>	30 heures, y compris les leçons du guide de l'enseignant et des apps
 <p>La programmation pour tous Premiers pas en programmation 2 Guide de l'enseignant</p>		De la 3 <sup>e</sup> à la 5 <sup>e</sup> année		Aucun	Apprivoiser des concepts de programmation fondamentaux et s'entraîner à penser comme des programmeurs.	<ul style="list-style-type: none"> <li>Cours Sorts de dragon de l'app Tynker</li> </ul>	36 heures, y compris les leçons du guide de l'enseignant et de l'app
 <p>La programmation pour tous Apprendre à coder 1 et 2 Guide de l'enseignant</p>		Dès l'école intermédiaire ou secondaire		Aucun	Apprendre les bases de la programmation en utilisant le code Swift.	<ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Leçons d'Apprendre à coder 1 et 2</li> <li>Cours iTunes U</li> <li>Badges Swift Playgrounds de l'Apple Teacher Learning Center</li> </ul>	Jusqu'à 85 heures, y compris les leçons d'Apprendre à coder 1 et 2 (app et guide de l'enseignant)
 <p>La programmation pour tous Apprendre à coder 3 Guide de l'enseignant</p>		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"> <li>App Swift Playgrounds</li> <li>Leçons d'Apprendre à coder 3</li> </ul>	Jusqu'à 45 heures, y compris les leçons d'Apprendre à coder 3 (app et guide de l'enseignant)
 <p>La programmation pour tous Introduction au développement d'apps avec Swift Guide de l'enseignant</p>		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"> <li>Introduction au développement d'apps avec Swift – Guide de l'élève (et fichiers de projets)</li> <li>Introduction au développement d'apps avec Swift – Guide de l'enseignant</li> </ul>	90 heures
 <p>Everyone Can Code App Development with Swift Teacher Guide</p>		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"> <li>App Development with Swift – Student Guide (et fichiers de projets)*</li> <li>App Development with Swift – Teacher Guide*</li> </ul>	180 heures

\* En anglais.

# Aperçu

Les ressources Introduction au développement d’applications avec Swift et App Development with Swift\* sont conçues pour enseigner aux collégiens, cégepiens et universitaires ayant peu ou pas d’expérience en programmation à devenir des développeurs d’applications, capables de donner vie à leurs idées.

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

La suite, App Development with Swift, plonge les élèves encore plus dans le monde de la programmation, qu’ils en soient à leurs débuts ou qu’ils cherchent à se perfectionner. S’ils sont déjà à l’aise avec Swift, Xcode et le développement d’applications iOS, ils peuvent parcourir les leçons plus rapidement ou passer directement aux ateliers pratiques pour bâtir des miniprojets et tester leur code dans les playgrounds. À la fin du cours, ils seront en mesure de concevoir une application entièrement fonctionnelle.

## Première app

---



### 5.1 Nouveau projet

Maintenant que vous êtes de plus en plus à l’aise sur le « terrain de jeu », vous vous demandez bien comment construire une application pour votre appareil iOS ou votre Apple Watch. Vous savez déjà qu’un grand nombre d’éléments variables doivent fonctionner ensemble pour qu’une application puisse s’exécuter. Vous êtes sur le point de découvrir que Xcode est le meilleur outil pour coordonner ces éléments.

Dans cette leçon en trois parties, vous allez bâtir une application iOS simple qui affiche une seule photo (elle s’appelle d’ailleurs Photo unique). Dans le premier exercice, vous allez créer un projet d’application à partir de zéro. Vous utiliserez ensuite Xcode pour explorer ce projet et apprendre à naviguer dans l’environnement de programmation.

Vous pourrez personnaliser chaque partie de votre application, de son icône sur l’écran d’accueil au comportement de ses boutons. Les panneaux et les commandes de Xcode affichent les nombreuses options offertes. Tout en vous familiarisant avec Interface Builder, l’outil de création d’interfaces de Xcode, vous continuerez à personnaliser votre première application.

À la dernière étape, vous ajouterez une image à votre projet et améliorerez son interface utilisateur (IU). Vous découvrirez aussi les rudiments d’Interface Builder, le puissant outil intégré à Xcode qui vous permet de créer l’IU de votre application. À la fin du deuxième exercice, votre application ressemblera à ceci, sauf qu’elle affichera une photo de votre choix.

Leçon 5.1 | Première app : Nouveau projet
23

Lesson 1.8

## Interface Builder Basics

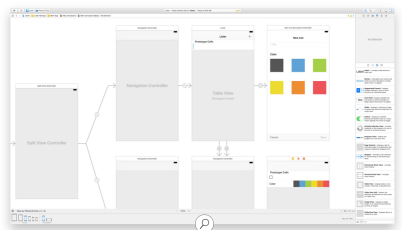
The best way to learn the basics of Interface Builder is to dive into Xcode and explore some of its features. Start by creating a new iOS project using the Single View Application template. Name the project "IBBasics".

**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.



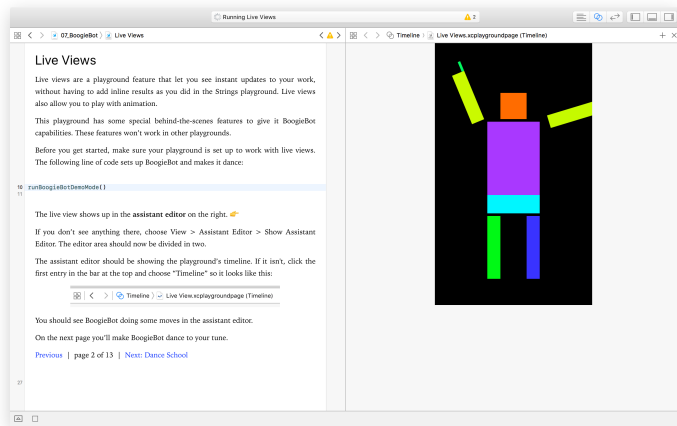
78
1.8 Interface Builder Basics | 79

\* En anglais.

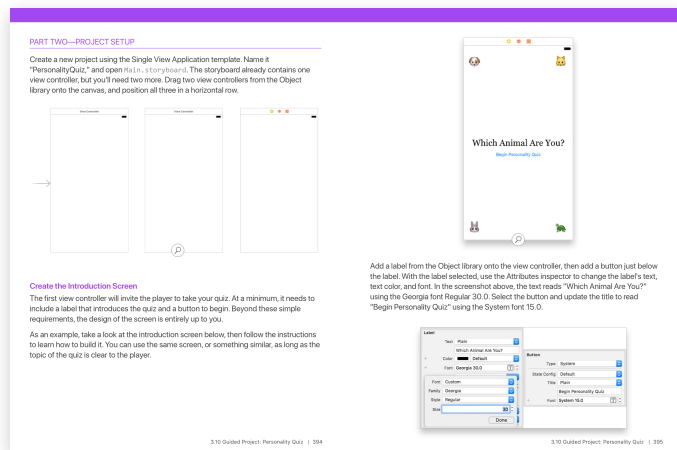


# Principales caractéristiques

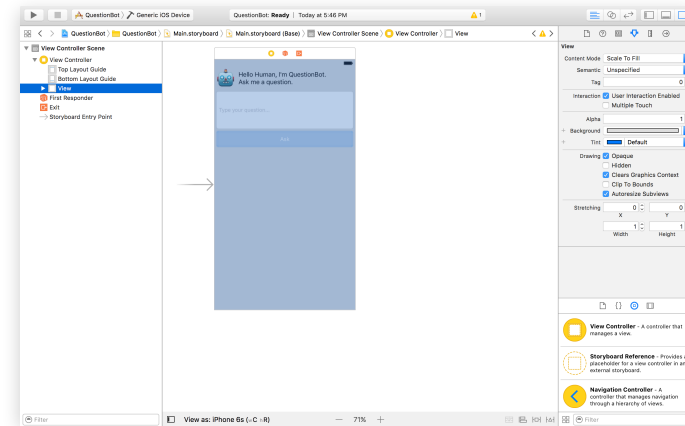
**Playgrounds.** Les élèves assimilent des notions de programmation en rédigeant du code dans des playgrounds, des terrains de jeu interactifs où ils font des essais de programmation et en voient les résultats immédiatement.



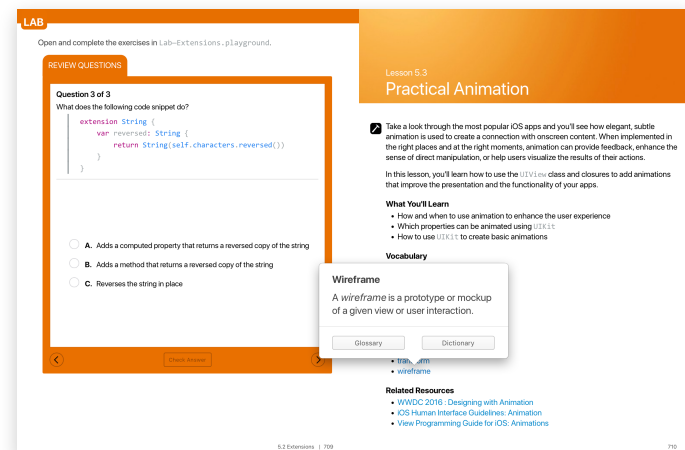
**Instructions pas à pas.** Des instructions détaillées avec images et vidéos guident les élèves à travers le processus de création d'une app dans Xcode.



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



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



# Structure des cours

## Introduction au développement d'apps avec Swift

Ce cours d'introduction d'un trimestre apprend aux élèves les fondements essentiels de la programmation au moyen du langage Swift. Ils acquièrent ainsi une expérience pratique des outils, techniques et concepts nécessaires pour concevoir une app iOS simple.

Dans la leçon de conception d'app, les élèves se familiarisent avec les étapes principales de la création de leur propre projet, dont le remue-méninges, la planification, le prototypage et l'évaluation. Et s'ils n'ont pas encore acquis toutes les habiletés nécessaires pour concevoir l'app en question, le travail qu'ils auront consacré au prototype les y aura préparés.

**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 de caractères.** 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 : 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 en assimilant les 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 des fonctions en découvrant comment les paramètres et les valeurs de retour peuvent les rendre plus efficaces et flexibles.

**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.

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"> <li><input checked="" type="checkbox"/> Paramètres, <input checked="" type="checkbox"/> Valeur de retour peindreToile() S'exécute seule et ne retourne aucune valeur.</li> <li><input checked="" type="checkbox"/> Paramètres, <input checked="" type="checkbox"/> Valeur de retour peindreToile(largeur: Int, hauteur: Int, couleurDominante: UIColor) Exécute une tâche qui change selon les arguments, mais ne retourne aucune valeur.</li> <li><input checked="" type="checkbox"/> Paramètres, <input checked="" type="checkbox"/> Valeur de retour peindreToile() -&gt; Toile Ne requiert aucune information, mais retourne une valeur.</li> <li><input checked="" type="checkbox"/> Paramètres, <input checked="" type="checkbox"/> Valeur de retour peindreToile(largeur: Int, hauteur: Int, couleurDominante: UIColor) -&gt; Toile Accepte de l'information et retourne une valeur.</li> </ol>	<p>Posez et explorez les questions de réflexion du guide de l'élève :</p> <ul style="list-style-type: none"> <li>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?</li> <li>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"> <li><code>func eteindreLeFour()</code> (Éteindre le four ne requiert aucun paramètre, et aucune valeur n'est retournée par cette action.)</li> <li><code>func prechaufferLeFour(temperature: Int)</code> (Préchauffer le four requiert un paramètre indiquant la température à atteindre.)</li> <li><code>func preparerDesBiscuits() -&gt; [Biscuit]</code> (Préparer des biscuits ne requiert aucun paramètre et produit un dessert.)</li> <li><code>func cuireAuFour(ingredients: [Ingredients]) -&gt; [Patisserie]</code> (Cuire au four requiert une liste d'ingrédients et retourne une délicieuse tournée de pâtisseries.)</li> </ul> </li> </ul>
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 bord que tiennent les élèves tout au long du cours.

# Structure des cours (suite)

**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 et à les utiliser en y ajoutant ou en y retirant des éléments; ils découvrent ce que sont les boucles for et comment les utiliser avec chaque élément 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 des interfaces utilisateur en se servant des outils d'Interface Builder et à relier les éléments d'interface au code au moyen d'actions et de sorties (outlets). Ils s'exercent à créer des sorties pour accéder aux propriétés d'une vue d'interface utilisateur, 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 interface utilisateur 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'interface utilisateur. 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 découvrent que les énumérations (enums) permettent de définir une liste d'options nommée, et ils explorent leur rôle, 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 présentés 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.

**Et ensuite?** Les élèves explorent une foule de ressources pour le développement d'apps, comme la page d'accueil du site des développeurs d'Apple, ou encore les vidéos de la conférence mondiale pour les développeurs (WWDC), qui porte sur les derniers cadres et outils conçus pour créer des apps sur chacune des plateformes d'Apple.



# Structures des cours (suite)

## App Development with Swift

Ce cours de deux trimestres comporte quarante-cinq leçons, chacune portant sur une compétence liée à Swift ou au développement d'apps. Chaque type de leçon repose sur une approche distincte :

- **Leçons Swift.** Ces leçons mettent l'accent sur des concepts précis. Les ateliers pratiques sont présentés dans des playgrounds pour permettre aux élèves de tester leur code et d'en voir immédiatement les résultats. Les fichiers des playgrounds sont inclus.
- **Leçons de développement d'apps.** Axées sur la création de fonctionnalités spécifiques aux apps iOS, ces leçons guident généralement les élèves dans la réalisation d'un miniprojet. Les ateliers les aident à mettre leurs apprentissages en pratique dans un nouveau contexte.

À la fin de chacune des cinq premières unités, les élèves 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 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'apps. Dans la dernière unité, ils se penchent sur la conception, le prototypage et l'architecture de leur propre app.

Le guide de l'enseignant présente des astuces pour approfondir ou adapter les leçons, favoriser la collaboration et offrir du soutien aux élèves qui en ont besoin. Il propose également des présentations Keynote téléchargeables pour chaque leçon, un corrigé pour les ateliers pratiques et une grille permettant d'évaluer les travaux des élèves.

**Unité 1 : Initiation au développement d'apps.** Les élèves 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, Xcode, la création et l'exécution d'une app, 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 élèves 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 élèves 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 persévérance.** Les élèves découvrent les vues déroulantes, les vues tabulaires et la création d'écrans d'entrée de données complexes. Ils découvrent également comment enregistrer des données, partager celles-ci avec d'autres apps et travailler avec des images tirées de la photothèque de l'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.

**Unité 5 : Utilisation du web.** Les élèves 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 élèves peuvent configurer le menu en utilisant les plats et les photos de leur choix.

**Unité 6 : Prototypage et planification de projets.** Les élèves 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.

# Renseignements supplémentaires

## Ressources Premiers pas en programmation pour les élèves de la maternelle à la 5<sup>e</sup> année

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

## Ressources Swift Playgrounds pour les élèves de la 6<sup>e</sup> à la 9<sup>e</sup> année (ou 3<sup>e</sup> 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](#).
- Prenez part à la conversation sur le [forum des développeurs d’Apple\\*](#).

### À propos de Swift

Swift est le langage de programmation puissant et intuitif qu’Apple a créé pour développer des applications. Il rend la programmation plus facile, plus flexible et plus amusante. 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](#).

\* En anglais.

# Corrélation avec le curriculum (11<sup>e</sup> année) : Introduction au développement d'apps avec Swift










Voici un aperçu de la conformité du guide de l'enseignant Introduction au développement d'apps avec Swift avec les normes du programme Conception, compétences pratiques et technologies de la Colombie-Britannique (11<sup>e</sup> année) en lien avec la programmation.

British Columbia Ministry of Education Intro to App Development with Swift CORRELATION MAP	Applied Design, Skills, and Technologies – Computer Programming (Grade 11)								
	Understand: Big Ideas			Do: Curricular Competencies			Know: Content Learning Standards		
	Understand context	Defining	Ideating	Prototyping	Testing	Making	Sharing	Applied Skills	Applied Technology
Students are expected to be able to do the following:									
Overall Alignment									
Playground Basics ⓘ									
Naming and Identifiers ⓘ									
Strings ⓘ									
Hello world! ⓘ									
First App ⓘ									
Functions ⓘ									
Boogie Bot ⓘ									
Constants and Variables ⓘ									
Types ⓘ									
Parameters and Results ⓘ									
Making Decisions ⓘ									
Instances, Methods and Properties ⓘ									
Question Bot ⓘ									

**KEY** ⓘ Information 100% coverage 75% coverage 50% coverage 25% coverage Further Notes



# Corrélation avec le curriculum (11<sup>e</sup> année) : Introduction au développement d'apps avec Swift (suite)

British Columbia Ministry of Education Intro to App Development with Swift CORRELATION MAP	Applied Design, Skills, and Technologies – Computer Programming (Grade 11)		
	Understand: Big Ideas	Do: Curricular Competencies	Know: Content Learning Standards
The big ideas consist of generalizations and principles and the key concepts important in an area of learning.	<b>Big Ideas</b>		
	Products can be designed for life cycle	Personal design interests require the evaluation and refinement of skills	Tools and technologies can be adapted for specific purposes
			
<p><b>KEY</b>     Information     100% coverage     75% coverage     50% coverage     25% coverage     Further Notes</p>			










# Corrélation avec le curriculum (12<sup>e</sup> année) : Introduction au développement d'applications avec Swift

Voici un aperçu de la conformité du guide de l'enseignant Introduction au développement d'applications avec Swift avec les normes du programme Conception, compétences pratiques et technologies de la Colombie-Britannique (12<sup>e</sup> année) en lien avec la programmation.

British Columbia Ministry of Education Intro to App Development with Swift CORRELATION MAP	Applied Design, Skills, and Technologies – Computer Programming (Grade 12)								
	Understand: Big Ideas			Do: Curricular Competencies			Know: Content Learning Standards		
	Understand context	Defining	Ideating	Prototyping	Testing	Making	Sharing	Applied Skills	Applied Technology
Students are expected to be able to do the following:									
Overall Alignment									
Playground Basics									
Naming and Identifiers									
Strings									
Hello world!									
First App									
Functions									
Boogie Bot									
Constants and Variables									
Types									
Parameters and Results									
Making Decisions									
Instances, Methods and Properties									
Question Bot									

**KEY** Information 100% coverage 75% coverage 50% coverage 25% coverage Further Notes

# Corrélation avec le curriculum (12<sup>e</sup> année) : Introduction au développement d'apps avec Swift (suite)

British Columbia Ministry of Education Intro to App Development with Swift CORRELATION MAP	Applied Design, Skills, and Technologies – Computer Programming (Grade 12)		
	Understand: Big Ideas	Do: Curricular Competencies	Know: Content Learning Standards
The big ideas consist of generalizations and principles and the key concepts important in an area of learning.	<b>Big Ideas</b>		
	Products can be designed for life cycle	Personal design interests require the evaluation and refinement of skills	Tools and technologies can be adapted for specific purposes
			
<p><b>KEY</b>     Information     100% coverage     75% coverage     50% coverage     25% coverage     Further Notes</p>			



# Corrélation avec le curriculum (CSTA) : Introduction au développement d'apps avec Swift

Voici un aperçu de la conformité préliminaire du guide de l'enseignant Introduction au développement d'apps avec Swift avec les normes de la Computer Science Teachers Association (CSTA) pour le niveau 3A. 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.

Normes intérimaires d'enseignement informatique de la CSTA (maternelle-fin secondaire) – Niveau 3A (9 <sup>e</sup> -10 <sup>e</sup> année/3 <sup>e</sup> -4 <sup>e</sup> secondaire)												
Norme de la CSTA	3A-A-2-1 Création d'artéfacts computationnels	3A-A-2-2 Collaboration	3A-A-7-3 Licences	3A-A-5-4 Réaction à un événement	3A-A-5-5 Recherche	3A-A-5-6 Concepts mathématiques	3A-A-4-7 Hiérarchie et abstraction	3A-A-4-8 Décomposition d'un problème	3A-A-4-9 Abstraction	3A-A-3-10 Conception d'algorithmes	3A-A-3-11 Modélisation et simulation	3A-A-6-12 Débogage
Rudiments des playgrounds												●
Désignation et identificateurs						●		●	●	●		●
Chaînes de caractères						●		●	●	●	●	
Hello, world!								●		●		●
Première app				●				●				
Fonctions						●		●	●	●	●	
BoogieBot						●		●	●	●	●	
Constantes et variables						●		●	●	●	●	●
Types						●		●	●	●	●	●
Paramètres et résultats						●		●	●	●	●	
Prise de décisions				●		●		●	●	●	●	
Instances, méthodes et propriétés						●		●	●	●	●	
QuestionBot				●		●		●	●	●	●	●
Tableaux et boucles						●		●	●	●	●	
Définition de structures						●		●	●	●	●	
QuestionBot 2				●		●		●	●	●	●	
Actions et sorties				●		●		●		●		●
Interfaces utilisateur dynamiques				●		●		●		●		●
Énumérations et instruction switch						●		●	●	●	●	
Projet final				●		●		●	●	●	●	
Conception d'app	●	●		●	●							

# Corrélation avec le curriculum (CSTA) : Introduction au développement d'applications avec Swift

Voici un aperçu de la conformité préliminaire du cours App Development with Swift avec les normes de la CSTA pour le niveau 3A. 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.

Conformité du cours App Development with Swift – Normes d'enseignement de la CSTA (maternelle-fin secondaire) – Niveau 3A (9 <sup>e</sup> -10 <sup>e</sup> année/3 <sup>e</sup> -4 <sup>e</sup> secondaire)												
CSTA Standard	3A-A-2-1 Création d'artefacts computationnels	3A-A-2-2 Collaboration	3A-A-7-3 Licences	3A-A-5-4 Réaction à un événement	3A-A-5-5 Recherche	3A-A-5-6 Concepts mathématiques	3A-A-4-7 Hiérarchie et abstraction	3A-A-4-8 Décomposition d'un problème	3A-A-4-9 Abstraction	3A-A-3-10 Conception d'algorithmes	3A-A-3-11 Modélisation et simulation	3A-A-6-12 Débogage
Unité 1 : Initiation au développement d'applications	●	●	●	●	●	●		●	●	●		●
Unité 2 : Introduction à UIKit	●	●		●		●		●	●	●		●
Unité 3 : Navigation et flux de travail	●	●		●	●	●		●	●	●		●
Unité 4 : Tableaux et persévérance	●	●		●		●		●	●	●		●
Unité 5 : Utilisation du web	●	●		●	●	●		●	●	●		●
Unité 6 : Prototypage et planification de projets	●	●		●	●	●		●	●	●	●	

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