



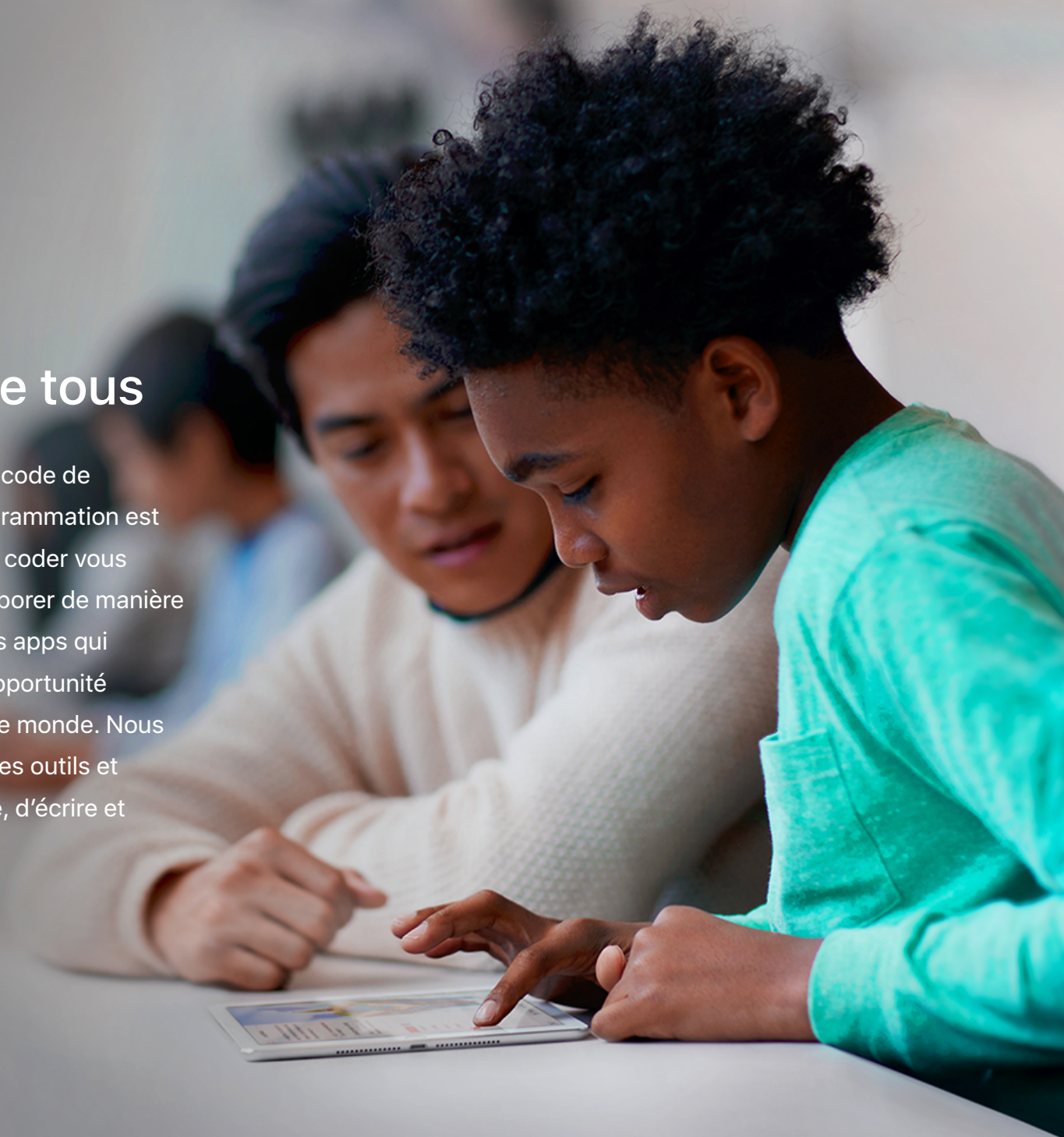
# Guide du programme App Development with Swift

Septembre 2017



# Le code à la portée de tous










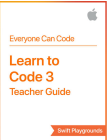





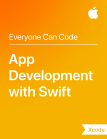


Les technologies ont leur propre langage : le code de programmation. Et nous pensons que la programmation est une compétence fondamentale. Apprendre à coder vous apprend à résoudre des problèmes et à collaborer de manière créative. Cela vous permet aussi de créer des apps qui donnent vie à vos idées. D'offrir à chacun l'opportunité d'imaginer quelque chose qui peut changer le monde. Nous avons élaboré un nouveau programme avec les outils et ressources permettant à chacun d'apprendre, d'écrire et d'enseigner la programmation.





# Programme Le code à la portée de tous

Le programme Le code à la portée de tous inclut toute une gamme de ressources qui guident les élèves, de leur initiation à la programmation à la conception de leurs premières apps. Le tableau ci-dessous offre un aperçu de toutes les ressources pédagogiques disponibles gratuitement.

| Programme   | Appareil  | Public                  | App   | Prérequis                | Présentation  | Supports pédagogiques  | Ressources d'assistance   | Nombre d'heures de cours incluses   |
|---|---|-------------------------|---|--------------------------|---|--|---|---|
|    |    | De la maternelle au CE1 |    | Aucun                    | Commencer à penser comme des programmeurs, avec découverte pratique des concepts de programmation à l'aide d'apps visuelles.  | <ul style="list-style-type: none"> <li>Leçons de l'app codeSpark Academy</li> <li>Cours Cadet de l'espace sur Tynker</li> </ul>          | <ul style="list-style-type: none"> <li>Débuter en programmation 1 : Guide d'enseignement</li> </ul>   | 30 heures, en comptant le Guide d'enseignement et les leçons de l'app                         |
|    |    | Du CE2 au CM2           |    | Aucun                    | Explorer les concepts de base de la programmation et s'entraîner à penser comme des programmeurs à l'aide d'apps visuelles.   | <ul style="list-style-type: none"> <li>Cours Sorts Dragon sur Tynker</li> </ul>  | <ul style="list-style-type: none"> <li>Débuter en programmation 2 : Guide d'enseignement</li> </ul>   | 36 heures, en comptant le Guide d'enseignement et les leçons de l'app                         |
|    |    | À partir du collège     |    | Aucun                    | Apprendre les bases de la programmation à l'aide du code Swift.   | <ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Leçons Apprendre à coder 1 et 2</li> <li>Cours iTunes U</li> </ul> | <ul style="list-style-type: none"> <li>Apprendre à coder 1 et 2 : Guide d'enseignement</li> <li>Badges Swift Playgrounds Apple Teacher Learning Center</li> </ul> | Jusqu'à 85 heures, en comptant le Guide d'enseignement et les leçons Apprendre à coder 1 et 2 |
|  |  | À partir du collège     |  | Apprendre à coder 1 et 2 | Approfondir ses compétences en programmation et apprendre à penser comme un développeur d'apps.   | <ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Leçons Learn to Code 3</li> </ul>                                  | <ul style="list-style-type: none"> <li>Learn to Code 3: Teacher Guide</li> </ul>  | Jusqu'à 45 heures, en comptant le Guide d'enseignement et les leçons Learn to Code 3          |
|  |  | Lycée et université     |  | Aucun                    | Acquérir une expérience pratique grâce aux outils, techniques et concepts nécessaires pour concevoir une app iOS simple de A à Z.   | Manuel Introduction au développement d'apps avec Swift et fichiers de projet   | <ul style="list-style-type: none"> <li>Introduction au développement d'apps avec Swift : Guide d'enseignement</li> </ul>  | 90 heures   |
|  |  | Lycée et université     |  | Aucun                    | Acquérir des connaissances solides sur Swift, UIKit et les technologies réseau grâce à des travaux pratiques et des projets guidés. À l'issue du cours, les élèves sont capables créer une app issue de leur imagination. | Livre App Development with Swift et fichiers de projet   | <ul style="list-style-type: none"> <li>App Development with Swift: Teacher Guide</li> </ul>   | 180 heures  |

# Présentation

Les cours Introduction au développement d'apps avec Swift et App Development with Swift sont destinés aux élèves du lycée et premières années d'université. Ils apprennent aux élèves ayant peu d'expérience en programmation à devenir des développeurs d'apps et à donner vie à leurs idées.

Le cours Introduction au développement d'apps avec Swift présente aux élèves l'univers du développement d'applications ainsi que les bases de Swift et de Xcode. Le cours se termine par un projet final dans lequel les étudiants peuvent choisir une app iOS à concevoir parmi deux apps simples.

Le développement d'apps avec Swift permet aux élèves d'aller encore plus loin, qu'ils découvrent la programmation ou souhaitent approfondir leurs connaissances. S'ils maîtrisent déjà le développement avec Swift, Xcode et iOS, ils peuvent survoler les leçons ou passer directement aux travaux pratiques afin de créer de mini-projets et de tester leur code dans des playgrounds. À l'issue du programme, ils sont en mesure de créer une app fonctionnelle tout droit sortie de leur imagination.

## Première app



### 5.1 Nouveau projet

Maintenant que vous connaissez mieux les aires de jeux, vous vous demandez peut-être comment concevoir une app compatible avec votre appareil iOS, ou même votre Apple Watch. L'exécution d'une app nécessite que de nombreux éléments distincts fonctionnent ensemble. Xcode est le meilleur outil pour les combiner.

Dans cette leçon en trois parties, vous allez concevoir SinglePhoto, une app iOS simple qui affiche une photo. Dans le premier exercice, vous allez créer un projet d'app à partir de zéro. Vous utiliserez ensuite Xcode pour explorer votre projet et apprendre à naviguer au sein de votre environnement de programmation.

Il est possible de personnaliser une app dans ses moindres détails : de son icône à son écran d'accueil, en passant par le comportement des boutons. Xcode est composé de volets et de contrôles qui affichent toutes les options mises à votre disposition. Et l'apprentissage d'Interface Builder vous permettra de personnaliser votre première app.

Dans la dernière étape, vous ajouterez une image à votre projet et modifierez l'interface utilisateur. Vous découvrirez également Interface Builder, un composant puissant d'Xcode dans lequel vous créerez l'interface utilisateur de votre app. À la fin du deuxième exercice, votre app ressemblera à l'image ci-contre, avec la photo de votre choix.

Leçon 5.1 | Première app : Nouveau projet

25

## Lesson 1.8

### Interface Builder Basics

Xcode has a built-in tool called Interface Builder that makes it easy to create interfaces visually. In this lesson, you'll learn how to navigate through Interface Builder, add elements onto the canvas, and interact with those elements in code.

**What You'll Learn**

- How to use Interface Builder to build user interfaces
- How to preview user interfaces without compiling the app

**Vocabulary**

- action
- canvas
- Document Outline
- view controller
- initial view controller
- outlet
- scene
- XIB

**Related Resources**

- Xcode Help: Interface Builder workflow
- Build a Basic UI

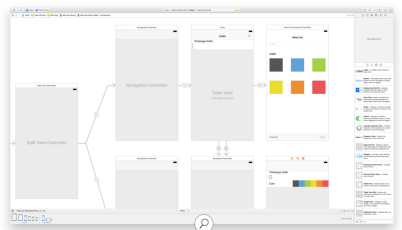
The best way to learn the basics of Interface Builder is to dive into Xcode and explore some of its features. Start by creating a new iOS project using the Single View Application template. Name the project "IBBasics".

**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.



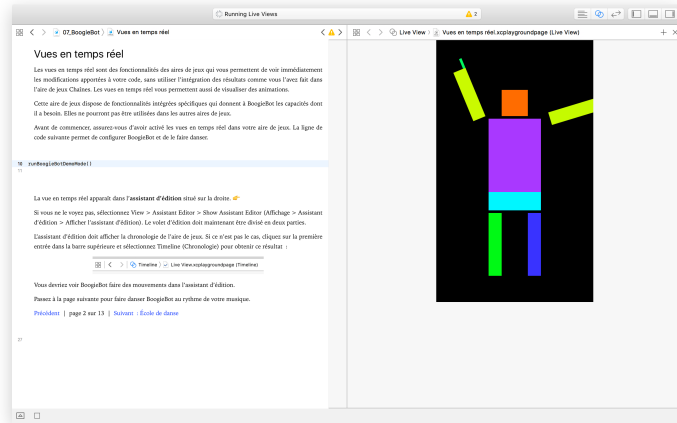
78

1.8 Interface Builder Basics | 79

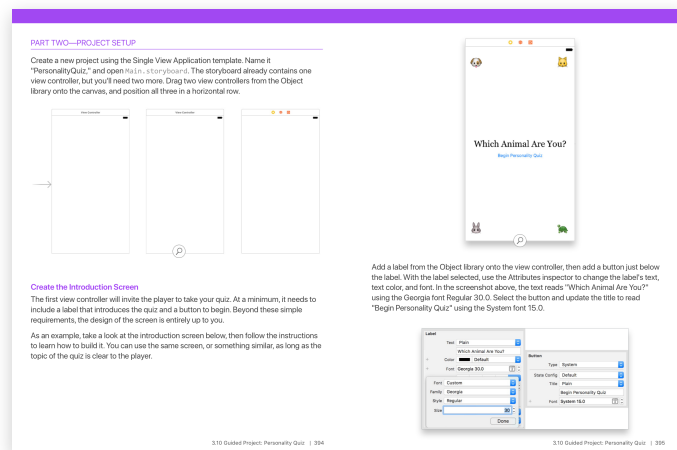


# Principales caractéristiques

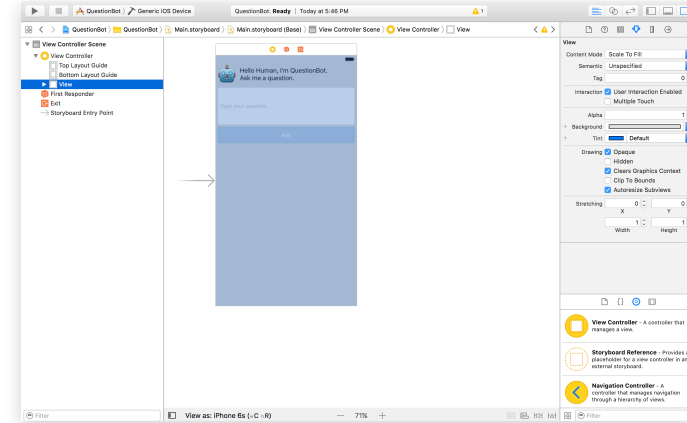
**Playgrounds.** Les élèves apprennent des notions de programmation en rédigeant du code dans des playgrounds, environnement de code interactif qui leur permet de faire des essais et d'en visualiser immédiatement les résultats.



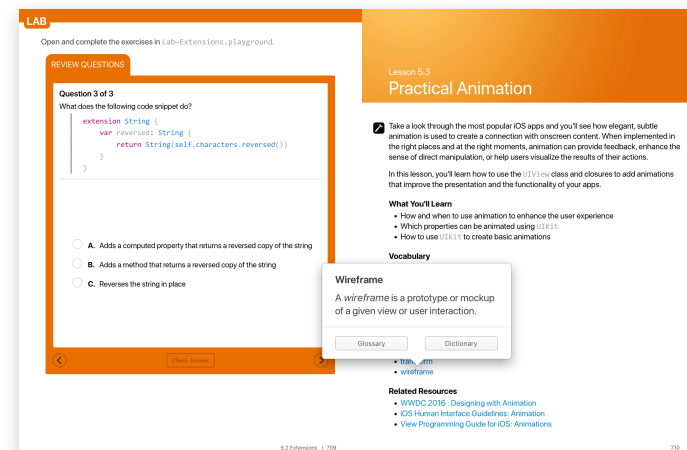
**Instructions détaillées.** Des instructions détaillées assorties d'images et de vidéos guident les élèves tout au long des étapes de création d'une app dans Xcode.



**Exemples de projets.** À l'aide des fichiers de projets fournis, les élèves peuvent tester certaines parties du code sans avoir à créer intégralement une app.



**Outils d'étude.** Les élèves peuvent vérifier leur compréhension et mettre en pratique ce qu'ils ont appris grâce, entre autres, à des questions de révision, à un vocabulaire essentiel et à des liens vers de la documentation.



# Plan du cours

## Introduction au développement d'apps avec Swift

Ce cours d'introduction, d'une durée d'un semestre, a été conçu pour aider les élèves à maîtriser les bases de la programmation en utilisant le langage Swift. Il permet aux élèves d'acquérir une expérience pratique des outils, des techniques et des concepts nécessaires pour concevoir de A à Z une app iOS basique.

Les leçons Conception d'apps guident les élèves tout au long du processus de conception d'une app, de la phase de brainstorming à celle d'évaluation de leur propre app, en passant par la planification et la création de prototypes. Bien qu'ils n'aient pas encore les compétences nécessaires pour réaliser concrètement l'app, le temps passé à mettre au point le prototype les mettra sur la bonne voie pour développer de futures apps.

**Leçon 1 : Les bases de l'aire de jeux.** Les élèves se familiarisent avec l'environnement interactif des aires de jeux.

**Leçon 2 : Nommage et identificateurs.** Les élèves apprennent les rudiments de la résolution de problèmes en utilisant des noms et des identificateurs appropriés.

**Leçon 3 : Chaînes.** Les élèves vont découvrir le concept des chaînes et d'interpolation de chaîne.

**Leçon 4 : Bonjour le monde !** Les élèves font leur entrée dans le monde de la programmation traditionnelle : ils apprennent 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 dans un simulateur iOS.

**Leçon 6 : Fonctions.** Pour comprendre l'efficacité des fonctions, les élèves combinent des étapes précises au sein d'une définition qu'ils pourront réutiliser à l'infini.

**Leçon 7 : BoogieBot.** Les élèves vont utiliser leurs connaissances sur les fonctions pour contrôler un robot animé dansant dans une aire de jeux.

**Leçon 8 : Constantes et variables.** Les élèves vont découvrir les concepts de constantes et de variables et améliorer ainsi leur maîtrise du nommage.

**Leçon 9 : Types.** Les élèves se familiarisent 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 approfondissent leur connaissance des fonctions en apprenant à utiliser des paramètres et des valeurs de retour, qui rendent les fonctions plus souples et plus puissantes.

**Leçon 11 : Prendre des décisions.** Les élèves apprennent à prendre des décisions dans leur code avec les instructions conditionnelles if/else, les valeurs booléennes true/false et les opérateurs de comparaison.

| MISE EN APPLICATION ET DÉVELOPPEMENT<br>DES COMPÉTENCES (25 minutes)   | SYNTHÈSE ET DISCUSSION (10 à 20 minutes)  |
|--|---|
| <p>Dans cette leçon, nous avons utilisé l'analogie d'un artiste peignant un tableau pour aider les élèves à différencier les quatre types de fonctions. Abordez les quatre types de fonctions :</p> <ol style="list-style-type: none"> <li>❌ Paramètres, ❌ Valeur de retour <code>paintPicture()</code><br/>La fonction réalise sa tâche et ne retourne pas de valeur.</li> <li>✅ Paramètres, ❌ Valeur de retour<br/><code>paintPicture(width: Int, height: Int, dominantColor: UIColor)</code> La fonction accomplit différentes tâches selon les arguments mais ne retourne pas de valeur.</li> <li>❌ Paramètres, ✅ Valeur de retour <code>paintPicture()</code><br/>→ <code>Painting</code>. La fonction n'a besoin d'aucune information mais retourne une valeur.</li> <li>✅ Paramètres, ✅ Valeur de retour<br/><code>paintPicture(width: Int, height: Int, dominantColor: UIColor)</code> → <code>Painting</code> La fonction accepte des informations et retourne une valeur.</li> </ol> | <p>Prenez le temps d'évoquer les questions de réflexion du guide des élèves.</p> <ul style="list-style-type: none"> <li>Vous savez désormais créer des fonctions capables d'intégrer des informations et de les exploiter. Mais la tâche réalisée reste la même. Imaginez que vous puissiez réaliser des tâches différentes selon les informations transmises.</li> <li>Quelles tâches et quels processus concrets seraient compatibles avec les différentes manières de définir une fonction (avec ou sans paramètres, avec ou sans type de retour) ? Voici quelques exemples :             <ul style="list-style-type: none"> <li><code>func. turnOffOven()</code> (Éteindre le four ne nécessite aucun paramètre, et aucune valeur n'est retournée à partir de cette action.)</li> <li><code>func. preheatOven(temperature: Int)</code> (Préchauffer un four nécessite d'indiquer la température comme paramètre pour commencer l'action.)</li> <li><code>func. bakeCookies()</code> → <code>[Cookie]</code> (Faire des cookies ne nécessite aucun paramètre mais vous donnera un résultat.)</li> <li><code>func. bake(ingredients: [Ingredient])</code> → <code>[BakedGood]</code> (Faire des pâtisseries nécessite une liste d'ingrédients et retourne une délicieuse fourée de pâtisseries.)</li> </ul> </li> </ul> |

Leçon 10 | Paramètres et résultats 52

Le Guide d'enseignement contient des activités connexes supplémentaires, des questions de discussion ainsi que des activités liées au journal de développement d'apps que les élèves enrichissent tout au long du semestre.

# Plan du cours (suite)

**Leçon 12 : Instances, méthodes et propriétés.** Les élèves complètent leurs connaissances sur les types en explorant les méthodes et les propriétés qui constituent une instance d'un type spécifique.

**Leçon 13 : QuestionBot.** Les élèves modifient un projet Xcode existant en écrivant une nouvelle logique pour un robot d'app qui répond à différentes questions.

**Leçon 14 : Tableaux et boucles.** Les élèves apprennent à créer et à exploiter des tableaux en ajoutant et en supprimant des objets. Ils découvrent également les boucles « for », qui permettent de traiter tous les objets d'un tableau.

**Leçon 15 : Définir des structures.** Les élèves réalisent qu'il est souvent utile de regrouper les informations et les fonctionnalités similaires au sein d'un type personnalisé.

**Leçon 16 : QuestionBot 2.** Les élèves poursuivent leur exploration de l'app QuestionBot en créant ChatBot, une app qui affiche l'historique des conversations. Les élèves explorent le modèle de source de données et créer un objet de source de données simple pour fournir des informations sur les objets Messages à afficher dans la vue en liste des messages. Ils vont s'entraîner à ajouter de nouveaux éléments à un tableau pour que ce tableau stocke les messages de l'objet de source de données et conserve l'historique de la conversation.

**Leçon 17 : Actions et outlets.** Les élèves apprennent à concevoir des interfaces utilisateur à l'aide d'Interface Builder et à intégrer les éléments d'interface au code par le biais d'outlets et d'actions. Ils vont s'entraîner à créer des outlets pour accéder aux propriétés d'une vue d'interface utilisateur, et des actions pour répondre à l'interaction de l'utilisateur avec les boutons ou autres commandes.

**Leçon 18 : Interfaces utilisateur adaptatives.** Les élèves découvrent un processus reproductible permettant de créer une interface utilisateur sur l'iPhone doté du plus petit écran, et de l'adapter à toutes les tailles et

orientations d'appareils. Ils vont explorer Mise en page auto, le système permettant de définir des contraintes d'emplacement et de taille pour les éléments de l'interface utilisateur. Puis ils utiliseront les vues en pile, un objet spécialement conçu pour appliquer des contraintes de mise en page automatique d'après des paramètres simples et un système de quadrillage. Grâce à ce processus, ils créent les apps SimpleCenter, ElementQuiz et AnimalSounds.

**Leçon 19 : Énumérations et switch.** Les élèves découvrent que les énumérations sont un moyen de définir une liste d'options nommée. Ils vont comprendre à quoi elles servent et comment les définir, puis ils identifieront diverses façons de les exploiter. Ils apprendront également à utiliser l'instruction switch pour exécuter conditionnellement un code spécifique basé sur une valeur définie par une énumération.

**Leçon 20 : Projet final.** Les élèves vont devoir réaliser de A à Z l'un des deux projets finaux proposés — ou les deux. Le premier projet proposé est un jeu Pierre, feuille, ciseaux ; le second, un générateur de mèmes. Les élèves revoient divers concepts abordés dans le cours pour concevoir l'interface utilisateur, les données du modèle et les objets du contrôleur composant l'application.

**Leçon 21 : Conception d'apps.** Les élèves vont appliquer un cycle de conception axé sur la réalisation de prototypes, très proche du processus suivi par les développeurs professionnels.

**Et après ?** Les élèves explorent un large éventail de ressources de développement d'apps, de la page d'accueil du site des développeurs Apple aux vidéos de la conférence Apple Worldwide Developers Conference (WWDC) présentant les derniers frameworks et outils conçus pour créer des apps destinées à toutes les plates-formes Apple.



# Plan du cours (suite)

## App Development with Swift

Ce cours, d'une durée de deux semestres, propose 45 leçons, chacune étant conçue pour transmettre une compétence spécifique, liée à Swift ou au développement d'apps. Chaque type de leçon adopte une approche particulière :

- **Leçons sur Swift.** Ces leçons s'axent sur des concepts spécifiques. Les travaux pratiques correspondants sont présentés dans des playgrounds, afin que les élèves puissent expérimenter avec le code et voir immédiatement le résultat. Les fichiers Playgrounds sont fournis.
- **Leçons sur le développement d'apps.** Axées sur l'élaboration de fonctionnalités spécifiques pour des apps iOS, ces leçons guident les élèves pas à pas tout au long d'un mini-projet. Les travaux pratiques permettent aux élèves de mettre en application ce qu'ils ont appris dans le cadre d'un nouveau scénario.

À l'issue de chacune des cinq premières unités, les élèves réalisent des projets guidés comprenant une description des fonctionnalités centrées sur l'utilisateur, un plan de projet et des instructions pour la création d'une app pleinement opérationnelle. Ces projets permettent aux élèves de créer les fonctionnalités qui les intéressent tout en effectuant le type de travail qu'on attendra d'eux dans un contexte professionnel de développement d'apps. Dans la dernière unité, ils réfléchiront à la façon de concevoir, de prototyper et de construire une app issue de leur propre imagination.

Le Guide d'enseignement propose des conseils pour prolonger ou adapter les leçons, améliorer la collaboration et soutenir les élèves nécessitant une aide renforcée. Il comprend également des présentations Keynote téléchargeables pour chaque leçon, le code de solutions pour les travaux pratiques et une grille d'évaluation du travail des élèves.

**Unité 1 - Prototyping and Project Planning.** Les élèves découvrent les notions de base sur les données, les opérateurs et le flux de contrôle dans Swift ainsi que la documentation, le débogage, Xcode, la création et l'exécution d'une app, et Interface Builder. Ils mettent ensuite ces connaissances en application dans le projet guidé Light, dans lequel ils créent une app de lampe torche simple.

**Unité 2 - Introduction to UIKit.** Les élèves explorent les notions Swift de chaînes, fonctions, structures, collections et boucles. Ils découvrent également UIKit, c'est-à-dire les vues et commandes système constituant une interface utilisateur, et les moyens d'afficher les données à l'aide du système Mise en page auto et des vues en pile. Ils mettent ces connaissances en pratique dans le projet guidé Apple Pie, dans lequel ils créent un jeu consistant à faire deviner des mots.

**Unité 3 - Navigation and Workflows.** Les élèves découvrent comment créer des workflows (flux de travaux) simples et des hiérarchies de navigation à l'aide de contrôleurs de navigation, de contrôleurs de barre d'onglets et de « segues ». Ils examinent également deux puissants outils de Swift : les types facultatifs (Optional) et les énumérations. Ils mettent ces connaissances en pratique dans le projet guidé Personality Quiz, une enquête personnalisée qui leur permet de révéler à l'utilisateur une réponse amusante.

**Unité 4 - Tables and Persistence.** Les élèves découvrent les vues par défilement, les vues par tableaux et la réalisation d'écrans de saisie complexes. Ils apprennent également à sauvegarder des données, partager des données avec d'autres apps et exploiter les images de la photothèque de l'utilisateur. Ils mettent en œuvre leurs nouvelles compétences dans le projet guidé List, une app de suivi des tâches permettant à l'utilisateur d'ajouter, de modifier et de supprimer des éléments au sein d'une interface familière, basée sur des tableaux. Les élèves peuvent personnaliser l'app pour permettre le suivi de tout type d'information, comme une collection, des tâches ou des playlists.

**Unité 5 - Working with the Web.** Les élèves découvrent les animations, la concurrence et l'exploitation du Web. Ils mettent en œuvre ce qu'ils ont appris dans le projet guidé Restaurant, une app de menu personnalisable qui affiche les plats d'un restaurant et permet à l'utilisateur de passer commande. Cette app utilise un service web permettant aux élèves de composer un menu à partir de leurs propres éléments et photos.

**Unité 6 - Prototyping and Project Planning.** Les élèves apprennent à concevoir, prototyper et construire un projet issu de leur propre imagination. Donnez-leur suffisamment de temps pour qu'ils puissent créer ce projet de façon indépendante.

# Informations complémentaires

## Télécharger les ressources Débuter en programmation

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

## Télécharger les ressources Swift Playgrounds

- [Learn to Code 1 & 2: iTunes U Course](#)
- [Apprendre à coder 1 et 2 : Guide d'enseignement](#)
- [Learn to Code 3: Teacher Guide](#)
- [App Swift Playgrounds](#)

## Télécharger les guides App Development with Swift

- [Introduction au développement d'applications avec Swift](#)
- [Introduction au développement d'applications avec Swift : Guide d'enseignement](#)
- [App Development with Swift](#)
- [App Development with Swift: Teacher Guide](#)

## Ressources complémentaires

- En savoir plus sur le programme [Le code à la portée de tous](#).
- Communiquez avec d'autres enseignants sur les [forums de développeurs Apple](#).

## À propos de Swift

Swift est un langage de programmation à la fois puissant et intuitif, créé par Apple pour l'élaboration d'applications. Il rend la programmation plus simple, plus souple et plus amusante. Swift n'est pas réservé aux débutants en programmation, c'est aussi un langage très puissant. Il a été conçu pour développer les programmes les plus simples tels que « Bonjour le monde ! », comme les logiciels les plus sophistiqués au monde.

En savoir plus sur [Swift](#).

## À propos de Xcode

Xcode, c'est l'app Mac utilisée pour créer une app Mac sur deux, ainsi que toutes les apps iOS. Elle comprend tous les outils dont vous avez besoin pour créer des expériences logicielles hors du commun. Et elle est disponible sous forme de téléchargement gratuit sur le Mac App Store.

En savoir plus sur [Xcode](#).

Les fonctionnalités sont sujettes à modification. La disponibilité des fonctionnalités peut varier en fonction des zones géographiques et des langues.

© 2017 Apple Inc. Tous droits réservés. Apple, le logo Apple, iPhone, iTunes U, Keynote, Mac, macOS et Xcode sont des marques d'Apple Inc., déposées aux États-Unis et dans d'autres pays. Swift et Swift Playgrounds sont des marques d'Apple Inc. App Store est une marque de service d'Apple Inc., déposée aux États-Unis et dans d'autres pays. Les autres noms de produits et de sociétés mentionnés dans ce document appartiennent à leurs propriétaires respectifs. Les caractéristiques des produits sont susceptibles d'être modifiées sans préavis. Les informations contenues dans ce document sont fournies à titre indicatif uniquement ; Apple n'assume aucune responsabilité quant à leur utilisation. Septembre 2017