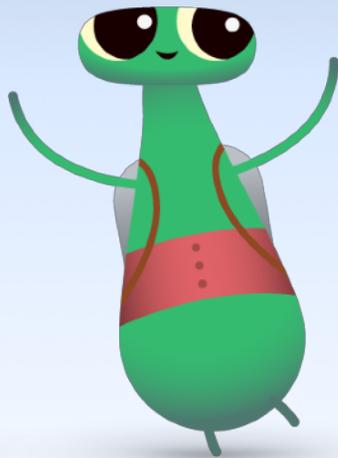




Swift Coding Club



Kit Swift Playgrounds

Bienvenue au Club de programmation Swift !

Apprendre à coder vous apprend à résoudre des problèmes et à collaborer de manière créative, et vous permet de créer des apps qui donnent vie à vos idées.

Les clubs de programmation Swift offrent une méthode ludique pour apprendre la programmation et la conception d'apps. Les activités utilisant Swift, le langage de programmation développé par Apple, favorisent l'esprit de collaboration et vous apprennent à programmer, à créer des prototypes d'apps et à réfléchir à l'impact que peut avoir la programmation sur votre environnement.

Vous n'avez pas besoin d'être enseignant ou expert en programmation pour diriger un club de programmation Swift. Les ressources étant adaptées au rythme de chacun, vous pouvez même apprendre en parallèle des membres du club. Et vous pourrez organiser un événement de présentation d'apps destiné à votre communauté pour mettre en valeur les idées et les réalisations de votre club.

Ce kit est organisé en trois sections :



Se lancer

Tout ce dont vous avez besoin pour créer un club de programmation Swift.



Apprendre et concevoir

Conseils et activités pour organiser les séances de votre club.



Célébrer

Ressources utiles pour planifier et organiser une présentation d'apps au sein de votre communauté.

Clubs de programmation Swift

Programmation par blocs |
8 à 11 ans

Apprenez les bases de la programmation grâce à des apps visuelles sur iPad.



Swift Playgrounds | 11 ans et +

Utilisez le code Swift pour apprendre les bases de la programmation avec Swift Playgrounds sur iPad.



Xcode | 14 ans et +

Apprenez à développer des apps avec Xcode sur Mac.



Se lancer



1. Téléchargez les ressources du club.

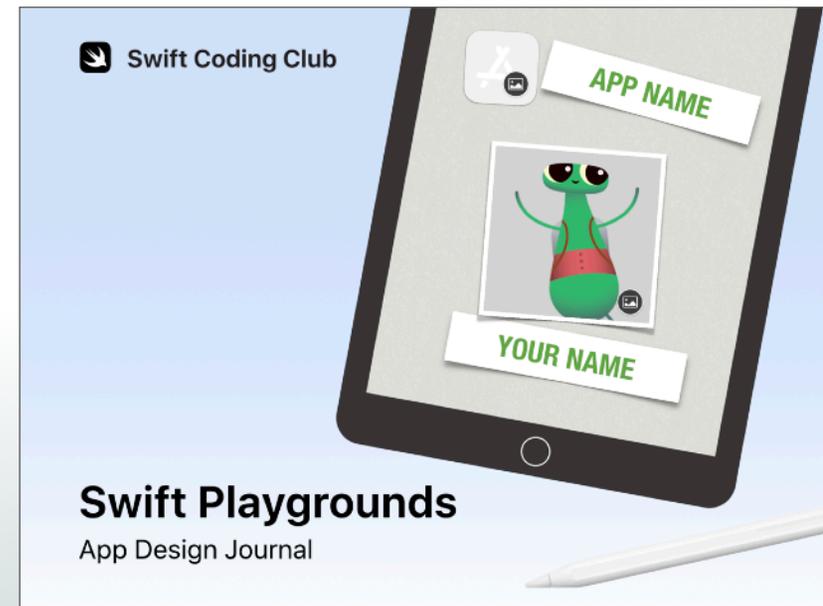
Utilisez AirDrop pour partager ces deux guides avec les membres du club lors de votre première séance. Ils sont également inclus dans ce document.



Activités de programmation

Apprenez les concepts de programmation à travers ces activités ludiques et collaboratives et résolvez des puzzles dans l'app Swift Playgrounds sur iPad.

[Télécharger les activités de programmation Swift Playgrounds >](#)



Journal de conception d'app

Explorez le processus de conception d'applications avec ce journal Keynote. Réfléchissez, planifiez, créez des prototypes et évaluez les idées d'app de votre club.

[Télécharger le journal de conception d'app Swift Playgrounds >](#)



2. Vérifiez vos équipements.

Avant la première séance, assurez-vous de disposer des équipements suivants :

- **iPad.** iPad mini 2 (ou modèle ultérieur), iPad Air (ou modèle ultérieur) ou iPad Pro exécutant iOS 11 (ou version ultérieure). Il est préférable que chaque membre possède son propre appareil, mais il est aussi possible de les partager et de programmer à plusieurs.
- **App Swift Playgrounds.** [Télécharger Swift Playgrounds >](#)
- **Playgrounds Apprendre à coder 1 et 2.** Téléchargez ces playgrounds dans l'app Swift Playgrounds.
- **Keynote.** Vous utiliserez l'app Keynote sur iPad pour créer vos prototypes d'app.
- **Ressources Club de programmation Swift.**



3. Élaborez un plan.

Voici quelques points à envisager :

- Qui sont les membres de votre club ? Quels sont leurs centres d'intérêt ? Ont-ils de l'expérience en programmation ou sont-ils débutants ?
- À quelle fréquence votre club se réunit-il ? Si vous prévoyez d'organiser un camp d'été, combien d'heures d'activités de programmation comportera-t-il ?
- De quelles technologies le club dispose-t-il ?
- Quels sont les objectifs de votre club ?



4. Faites-vous connaître.

Parlez du club de programmation Swift autour de vous. Voici quelques idées et ressources qui vous aideront à attirer de nouveaux membres dans votre club :

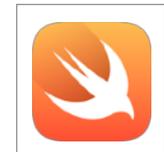
- **Annoncez l'ouverture de votre club.** Utilisez les e-mails, les médias sociaux, le Web, des prospectus ou le bouche à oreille.
- **Organisez une réunion informelle.** Demandez aux personnes intéressées par votre club quels sont leurs centres d'intérêt et quels types d'apps elles aimeraient concevoir. Partagez des idées sur l'organisation d'une présentation de conception d'app et sur la façon d'y faire participer les membres. Vous pouvez également partager une courte vidéo en ligne pour parler du club.

Les éléments suivants pourront vous aider à promouvoir et à personnaliser votre club de programmation Swift :

- **Affiches.** [Téléchargez ce modèle gratuit](#) et personnalisez-le pour créer votre propre affiche. Imprimez-la et affichez-la, ou créez une affiche numérique que vous partagerez en ligne. Pensez à préciser où les séances du club seront organisées, la date et l'heure, et les modalités d'inscription.
- **Autocollants et tee-shirts.** Utilisez ces [autocollants Club de programmation Swift](#) pour faire la promotion de votre club. Les tee-shirts sont un bon moyen d'identifier les membres participant à l'organisation des événements de présentation d'apps. Téléchargez le [modèle de tee-shirt Club de programmation Swift](#) pour faire des tee-shirts pour les membres de votre club.



Affiche Club de programmation Swift



Autocollant Club de programmation Swift

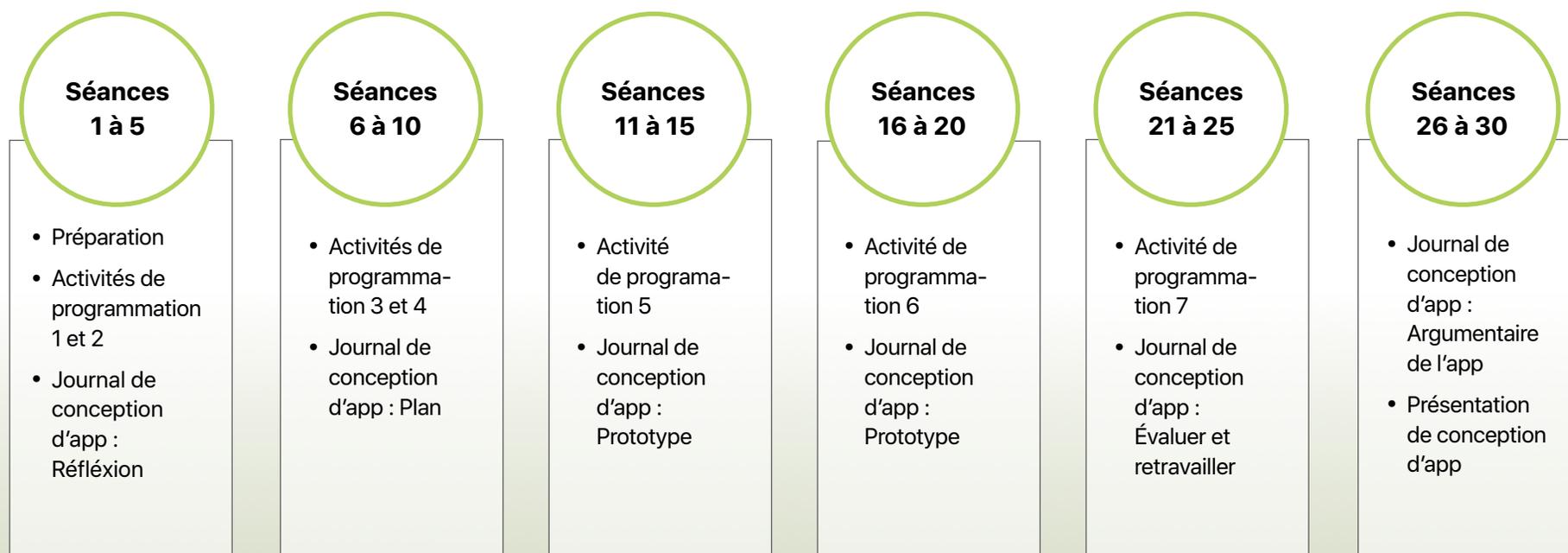


Tee-shirt Club de programmation Swift



Apprendre et concevoir

Les ressources du club sont conçues pour mélanger des activités de programmation et de conception d'apps. Vous pouvez également ajouter des séances en lien avec les centres d'intérêt de vos membres. Vous trouverez ci-dessous un exemple de planning pour 30 séances d'une heure.



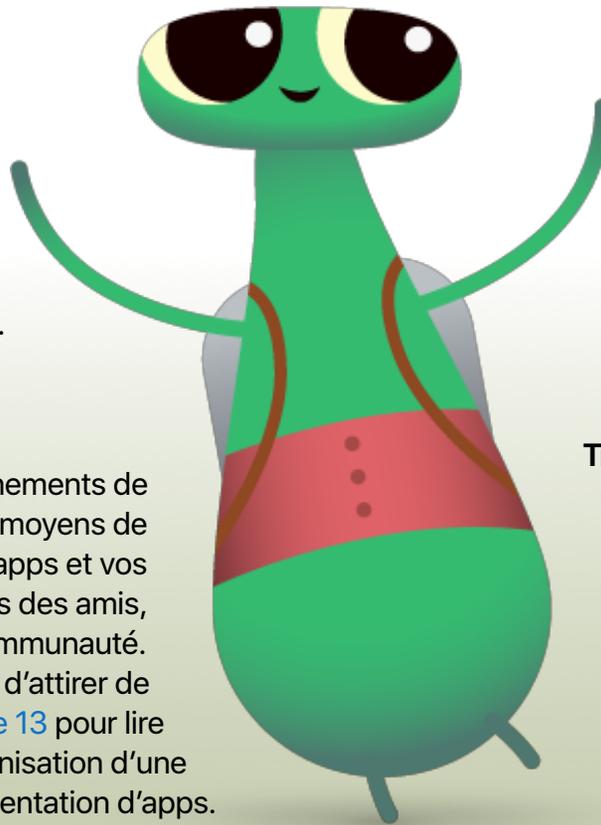
Pour aller plus loin dans la conception et la programmation d'apps, vous pouvez envisager d'ajouter des séances visant, par exemple, à créer un parcours d'obstacles pour un drone ou une mission de sauvetage pour un robot. Il est aussi possible d'inviter des intervenants ou d'organiser des excursions pour encourager la réflexion sur la conception d'apps.



Mettez en place une équipe de responsables. Il sera plus simple et plus agréable de gérer le club si vous êtes entouré d'un groupe de membres pour vous épauler. Parmi les membres de votre club, lesquels ont des compétences de leadership ? Envisagez de trouver des responsables pour gérer les événements, la programmation, la conception d'app, etc.

Apprenez ensemble. Les responsables de club n'ont pas besoin de tout connaître. Encouragez les membres à développer leurs compétences de recherche de solutions et de résolution de problèmes, puis invitez-les à aider leurs camarades.

Présentez les projets. Les événements de présentation d'apps sont d'excellents moyens de promouvoir votre club, vos idées d'apps et vos compétences de programmation auprès des amis, de la famille, des enseignants et de la communauté. Cela peut aussi vous permettre d'attirer de nouveaux membres. Consultez la [page 13](#) pour lire des conseils portant sur l'organisation d'une présentation d'apps.

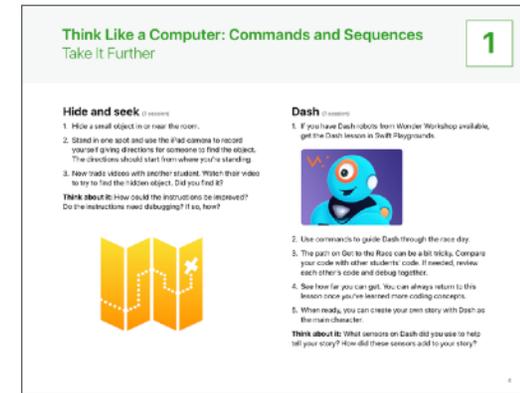
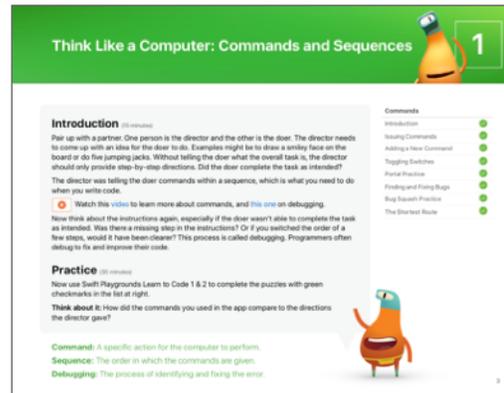


Partagez vos idées. Certains membres auront envie de concevoir des jeux. D'autres pourront vouloir créer des apps pour aider les gens, apprendre le langage Swift ou contrôler des robots. Arrangez-vous pour que les membres travaillent ensemble sur des projets qui leur tiennent à cœur.

Travaillez ensemble. Parfois, les membres plus expérimentés prennent de l'avance sur les autres. Demandez à ces membres de faire de la programmation en binôme avec des débutants. L'enseignement est une excellente méthode d'apprentissage !



Activités de programmation Swift Playgrounds



Activités de programmation :
ces activités de groupe se basent sur Swift Playgrounds et présentent les concepts et compétences essentiels en matière de programmation.

Concepts de programmation :
dans chaque activité, les membres du club apprennent un concept de programmation fondamental et le découvrent dans un contexte de tous les jours, puis ils doivent le mettre en application pour résoudre des puzzles dans Swift Playgrounds.

Pour aller plus loin : chaque concept de programmation est associé à une activité Pour aller plus loin. La première activité renforce la compréhension du concept de programmation en question et encourage la communication et le travail d'équipe. Les membres doivent utiliser leurs connaissances dans le cadre d'un projet créatif conçu sur iPad.

La deuxième activité facultative invite les membres à mettre en application le concept dans un playground issu des sections Défis, Abonnements et Points de départ de Swift Playgrounds. Certaines activités nécessitent l'utilisation d'appareils connectés spécifiques.



Everyone Can Code

Learn to Code 1 & 2
Teacher Guide

Swift Playgrounds

Besoin de plus d'informations ou d'approfondir le sujet ?

[Télécharger Swift Playgrounds : Guide d'enseignement Apprendre à coder 1 et 2 >](#)

Quelques conseils pour apprendre avec Swift Playgrounds



Commencez par étudier les puzzles. Encouragez les membres du club à étudier le monde d'Octet en zoomant et en pivotant dans la vue en direct pour mieux visualiser les étapes à suivre. Ils peuvent aussi l'afficher en plein écran en touchant de façon prolongée la partition entre les deux fenêtres, puis en faisant glisser vers la gauche.

Décomposez les puzzles. Les puzzles deviennent de plus en plus complexes. Les membres du club peuvent décomposer les puzzles pour mieux réfléchir aux étapes nécessaires à leur résolution. Ils peuvent utiliser Pages ou Notes pour planifier et noter les étapes avant de saisir le code.

Mettez en place un centre d'assistance. Réservez un espace où les experts du club pourront apporter de l'aide aux autres membres.



Trouvez plusieurs solutions. Chaque puzzle peut être résolu de plusieurs manières. Si certains membres terminent en avance, invitez-les à trouver d'autres moyens de les résoudre. En ouvrant leur réflexion et en comparant différentes solutions, les membres développent leur esprit critique.

Programmez en binôme. Demandez aux membres du club de travailler à plusieurs sur le même iPad. Ils chercheront ensemble des solutions aux puzzles et écriront le code à tour de rôle.

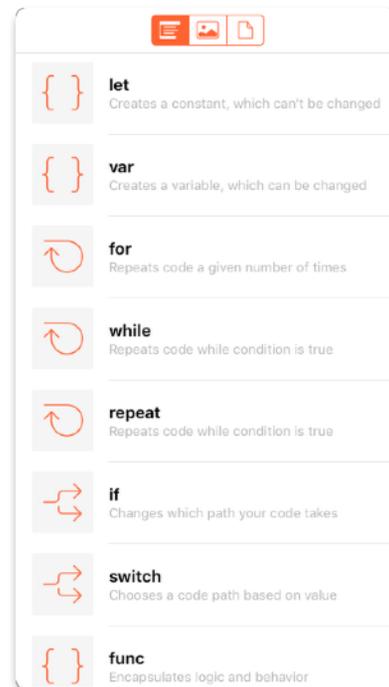
Utilisez les fonctionnalités d'accessibilité. Swift Playgrounds prend en charge les fonctionnalités d'assistance intégrées à iOS pour que l'apprentissage de la programmation soit accessible à tous. Les développeurs peuvent, par exemple, inverser les couleurs, activer les nuances de gris et zoomer pour ajuster la visibilité.

Explorer Swift Playgrounds



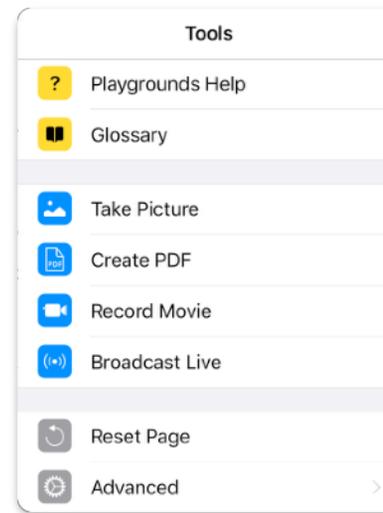
Bibliothèque d'extraits.

Pour limiter la saisie, touchez  dans la barre d'outils et allez dans la Bibliothèque d'extraits pour faire glisser des blocs de code courants dans votre code.



A vertical list of Swift code snippets with icons and descriptions:

- let**: Creates a constant, which can't be changed
- var**: Creates a variable, which can be changed
- for**: Repeats code a given number of times
- while**: Repeats code while condition is true
- repeat**: Repeats code while condition is true
- if**: Changes which path your code takes
- switch**: Chooses a code path based on value
- func**: Encapsulates logic and behavior

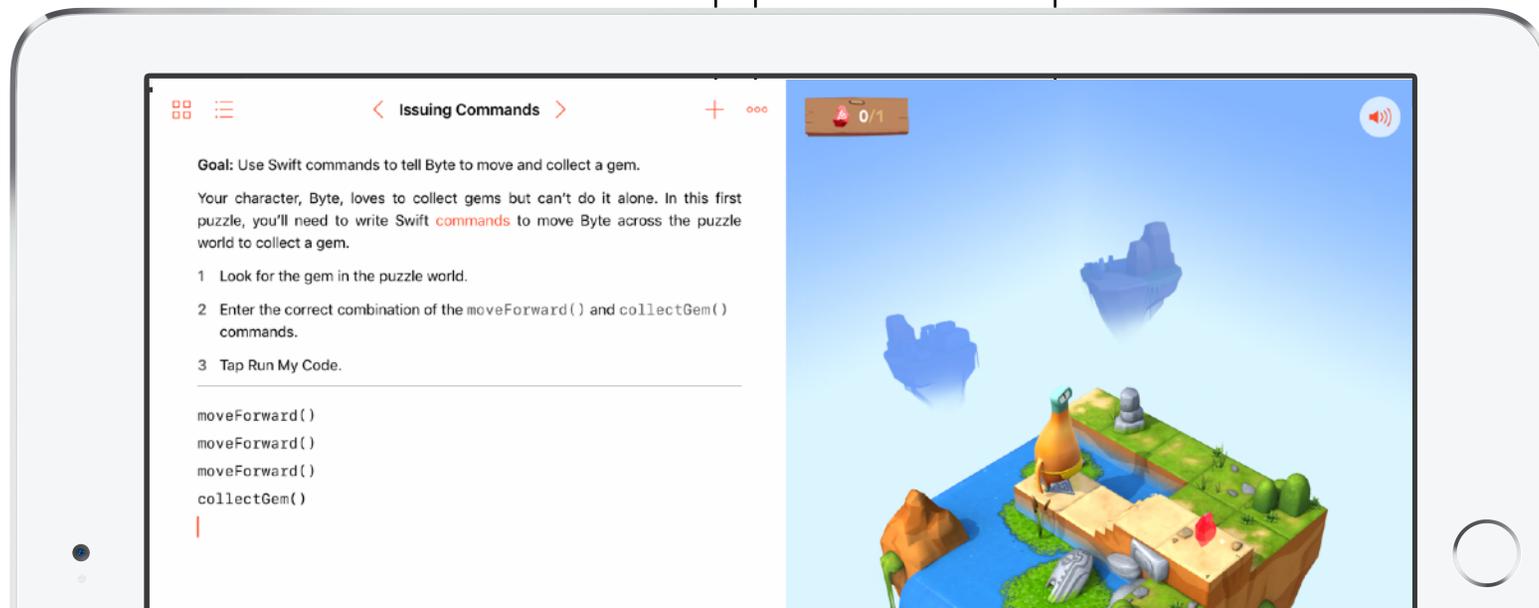


A vertical list of tools with icons:

- Playgrounds Help
- Glossary
- Take Picture
- Create PDF
- Record Movie
- Broadcast Live
- Reset Page
- Advanced



Choisissez un personnage. Pour une expérience plus personnelle, touchez le personnage et choisissez-en un autre.



The main interface of the Swift Playground. On the left, a text area contains instructions for a puzzle:

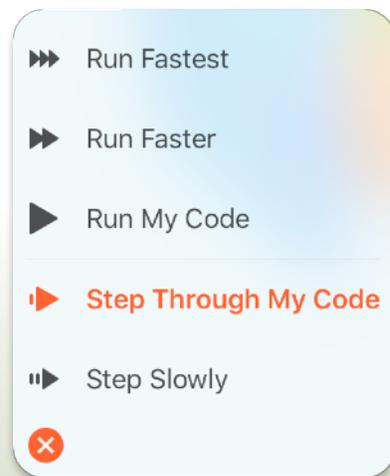
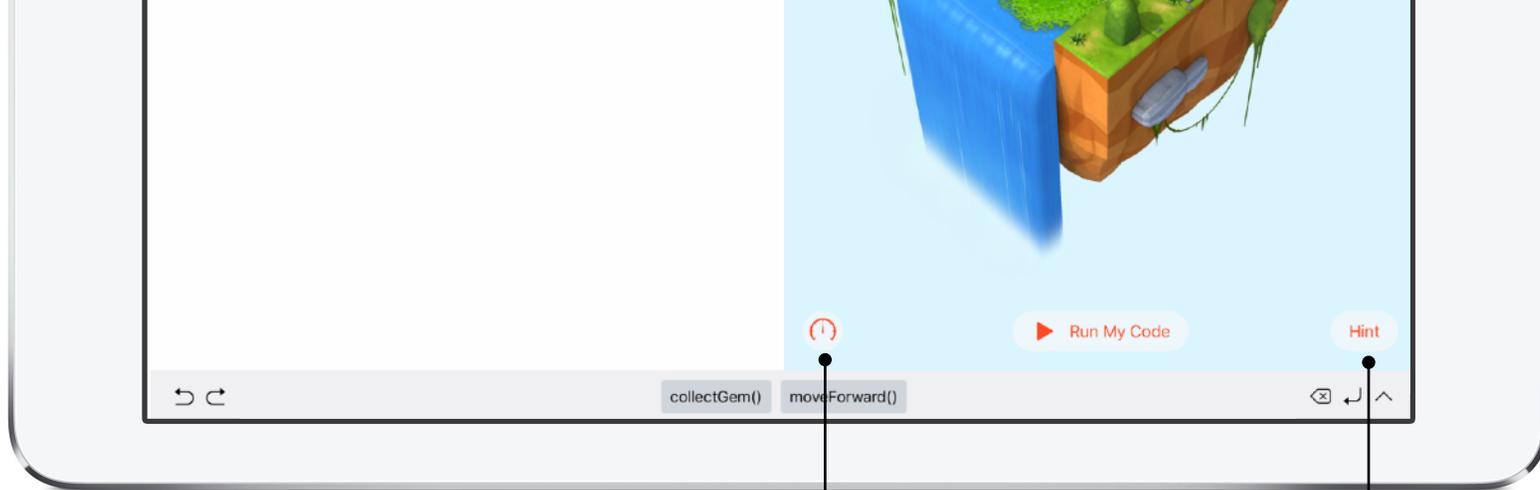
Goal: Use Swift commands to tell Byte to move and collect a gem.

Your character, Byte, loves to collect gems but can't do it alone. In this first puzzle, you'll need to write Swift **commands** to move Byte across the puzzle world to collect a gem.

- Look for the gem in the puzzle world.
- Enter the correct combination of the `moveForward()` and `collectGem()` commands.
- Tap Run My Code.

```
moveForward()  
moveForward()  
moveForward()  
collectGem()  
|
```

On the right, a 3D puzzle world is shown with a character (Byte) on a platform, a gem, and a floating island in the background.



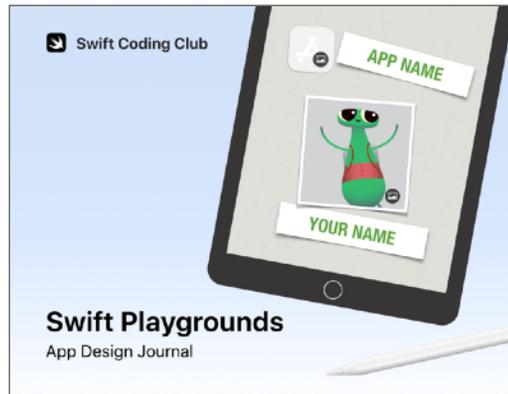
Indice. Cette fonctionnalité aide les apprenants en leur faisant des suggestions. Elle peut aussi donner la solution, mais un simple copier-coller ne suffira pas à résoudre le puzzle. Les développeurs devront tout de même suivre les étapes nécessaires et écrire eux-mêmes le code.

Contrôlez la vitesse. Accélérez ou ralentissez la vitesse de lecture du code.

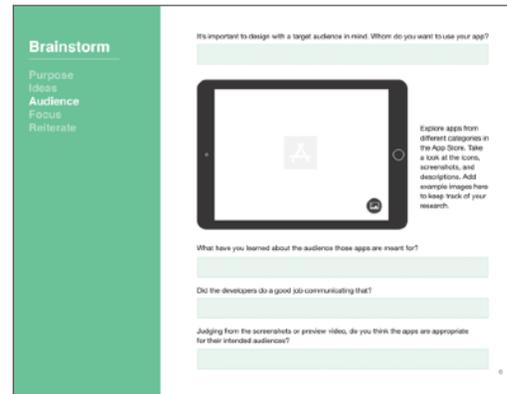
Surlignez le code exécuté. Utilisez la fonction Avancer dans mon code pour surligner les lignes de code en cours d'exécution et mieux comprendre son fonctionnement.



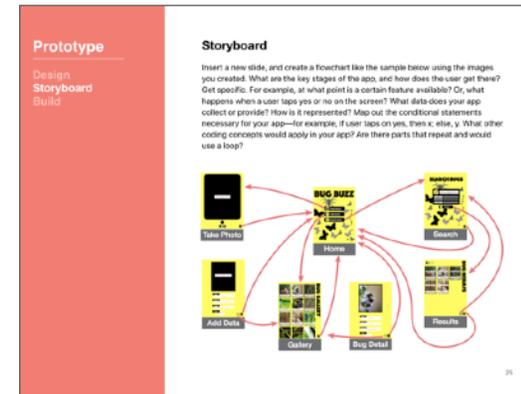
Journal de conception d'app



Les développeurs utilisent ce journal Keynote pour connaître les fonctionnalités des apps et concevoir des apps permettant de résoudre des problèmes rencontrés par la communauté.



Les membres du club se séparent en plusieurs équipes pour réfléchir à la solution de l'app, la planifier, puis créer un prototype fonctionnel de l'app dans Keynote.



Le journal guide les développeurs dans les processus d'évaluation de leur conception et d'ajustement de leurs prototypes, des étapes incontournables pour tout concepteur d'apps professionnel.

Les membres mettent au point une présentation ou une vidéo argumentaire de trois minutes et exposent leur travail au cours d'une présentation de conception d'app.



Célébrer



Présentation de conception d'app

Le processus de conception d'app et la présentation sont des moments que vous pouvez utiliser pour impliquer la communauté dans son ensemble et réfléchir à l'impact que peuvent avoir les apps sur la résolution de problèmes contemporains. La présentation est aussi idéale pour mettre en avant le talent des membres de votre club !

1. Préparez le grand jour. Choisissez une date pour la présentation et invitez les élèves, les enseignants, les parents et les membres de la communauté à y assister.

Donnez le temps à chaque équipe de présenter l'argumentaire de son app et de participer à une courte session de questions-réponses. Si votre club compte beaucoup de membres, vous pouvez le diviser en deux groupes qui regarderont à tour de rôle les argumentaires de l'autre.

Si vous le pouvez, clôturez l'événement avec un diaporama amusant de photos prises pendant les séances du club.



2. Distribuez des prix. La compétition amicale peut être très motivante. Incitez les membres du club à s'impliquer en leur remettant des prix qui récompensent leurs réussites dans la conception de leur app. Vous pouvez par exemple créer les prix suivants :

- Meilleure ingénierie
- Meilleure innovation
- Meilleure conception
- Meilleur argumentaire

Vous pouvez aussi encourager le public à participer avec un grand prix du public.



Vous pouvez télécharger et modifier [ce certificat](#) pour l'adapter aux prix que vous remettrez.

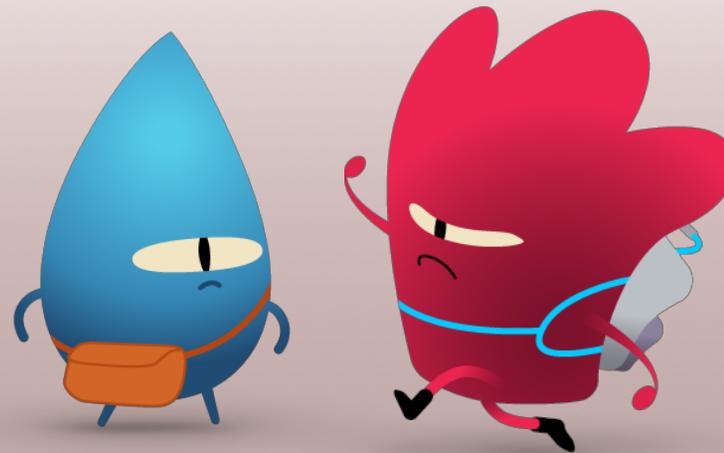


3. Impliquez des juges et des mentors. Les juges et les mentors peuvent être des enseignants ou des membres de l'établissement, des élèves ayant des connaissances avancées en programmation, des experts du secteur du développement ou de la conception, des personnalités importantes de la communauté, ou des personnes à qui l'idée d'app serait profitable.

Les juges peuvent faire connaissance avec le club et ses membres avant la présentation. Vous pouvez les inviter à intervenir et à partager leur expertise avec les élèves pendant les phases de réflexion et de planification de la conception de leur app.

4. Choisissez un gagnant. La grille d'évaluation située à la page suivante est mise à la disposition des juges pour évaluer les argumentaires d'app et fournir un retour aux élèves. Vous pouvez aussi transmettre cette grille d'évaluation aux développeurs avant la présentation, dans le cadre de la phase d'évaluation du processus de conception d'app.

5. Partagez et inspirez. Vous pouvez enregistrer les présentations des élèves puis les partager au niveau local et réaliser un montage des meilleurs moments pour attirer de nouveaux membres dans le club.



Grille d'évaluation

Télécharger >

Catégorie	Novice (1 point)	Intermédiaire (2 points)	Compétent (3 points)	Expert (4 points)	Points
Contenu de la présentation	Comporte des informations essentielles, par exemple l'objectif et le public destinataire.	Détaille clairement l'objectif de l'app, sa conception et la façon dont elle répond aux besoins des utilisateurs.	Présente des explications claires et percutantes sur le problème à résoudre, la demande du marché, le public et la façon dont l'app a été conçue pour répondre aux besoins des utilisateurs.	Présente l'app de manière convaincante, en se fondant sur des faits, au sujet de la façon dont l'app répond, dépasse ou redéfinit les besoins des utilisateurs.	
Qualité de la présentation	Informative ; seul un membre de l'équipe intervient.	Confiante, enthousiaste ; plusieurs membres de l'équipe interviennent.	Attrayante, bon usage des visuels pour appuyer le propos ; l'équipe met en valeur la contribution de chacun de ses membres.	Créative, propos marquant ; support visuel attrayant ; transitions fluides entre les membres de l'équipe.	
Interface utilisateur	Écrans pertinents par rapport à l'objectif de l'app.	Design clair et fonctionnel, doté d'éléments reconnaissables ; le prototype prend en charge les tâches essentielles.	Design élégant, clair, plaisant, avec un emploi bien pensé des couleurs, de la mise en forme et de la lisibilité ; le prototype est plaisant à l'utilisateur durant la navigation.	Le design permet à l'utilisateur d'interagir avec le contenu ; le prototype s'appuie sur l'animation, les couleurs et la mise en forme pour une utilisation fluide et attrayante.	
Expérience utilisateur	Intention claire ; les utilisateurs sont en mesure d'atteindre un ou plusieurs objectifs.	Navigation cohérente, conforme aux normes et intuitive en matière de contenu.	Adaptable aux besoins des utilisateurs ; garantit l'accessibilité, la confidentialité et la sécurité.	Innovante, surprenante et agréable ; crée une expérience utilisateur nouvelle qui la différencie de ses concurrentes.	
Concepts de programmation	Lien entre les fonctionnalités de l'app et le code sous-jacent.	Explications sur l'utilisation de concepts de codage généraux, par exemple les types de données, la logique conditionnelle ou les événements tactiles.	Description des tâches de codage particulières nécessaires à la conception de leur app ; démonstration sur la façon dont le code en question est à l'origine des fonctionnalités de l'app.	Explications sur l'architecture de l'app, la structure des données, les algorithmes et les fonctionnalités ; discussion sur le processus de décision qui a mené à cette approche.	
Révision technique (facultatif) <i>Pour les prototypes d'apps fonctionnels en Xcode. Les membres du jury doivent connaître les bonnes pratiques du développement iOS et Swift.</i>	Le code Swift est compatible avec certains exemples particuliers ; le code reste élémentaire et ne comporte aucune abstraction.	Le code s'exécute sans erreurs, quel que soit le scénario ; le code reste élémentaire avec quelques abstractions.	Le code est structuré autour de conventions de nommage Swift claires ; multiples abstractions ; respect des directives iOS.	Le code est bien étayé de commentaires ; utilisation efficace des fonctionnalités Swift ; emploi de motif d'architecture tel que Model-View-Controller.	
Commentaires :					0 Note globale



Club de programmation Swift

Swift Playgrounds

Certificat de réussite

Attribué à

Pour



Signature

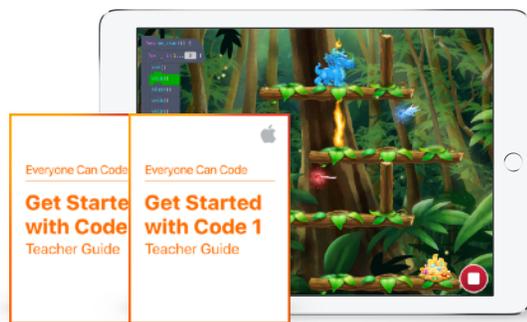
Date

Pour aller plus loin

Le club de programmation Swift est une première étape dans votre apprentissage de la programmation. Le programme Le code à la portée de tous propose des ressources amusantes pour passer des bases de la programmation sur iPad à la création de véritables applications sur Mac.

Et vous pouvez aller au-delà des activités proposées par le club. Les guides d'enseignement complets permettent également aux enseignants d'intégrer la programmation en classe, avec des cours progressifs adaptés au programme et destinés à tous les élèves de la maternelle à l'université.

[Voir toutes les ressources Le code à la portée de tous >](#)



[En savoir plus sur le programme Débuter en programmation >](#)



[En savoir plus sur le programme Swift Playgrounds >](#)



[En savoir plus sur le programme Développement d'applications avec Swift >](#)

