Smartfuture
Certifications | Training | Technology solutions

Formation & Certification • App Development with Swift Associate

App Development with Swift – Associate

Formation en distanciel (présentiel sur demande). Acquérez une base solide en programmation Swift et en développement d'applications iOS avec Xcode, tout en comprenant l'impact du numérique sur la société et les organisations.

Distributeur officiel Certiport | Centre d'examen Certiport | Learn • Practice • Certify

Durée
20–24 h

Examen
App Development with Swift Associate

Modalité
Distanciel

Niveau
Associate – Début à intermédiaire

INSCRIPTION / RÉSERVATION

Je m'inscris maintenant

- **Learn** : accès aux ressources pédagogiques officielles Apple & Certiport couvrant l'ensemble des domaines de l'examen.
- **Practice** : entraînement structuré à travers des exercices pratiques en Swift, projets Xcode et simulations d'examen.
- **Certify** : voucher d'examen App Development with Swift Associate inclus pour passer la certification dans notre centre.

OBJECTIFS PÉDAGOGIQUES

- Ce programme couvre la création d'applications iOS, de l'idée au prototype, avec Xcode et Interface Builder.
- Il enseigne les bases de Swift et la connexion de l'interface au code.
- Il prépare aussi à corriger les erreurs courantes et à l'examen App Development with Swift Associate.

PUBLIC CIBLE

- Ce programme s'adresse aux débutants et étudiants souhaitant découvrir le développement iOS.
- Il cible aussi les enseignants et formateurs voulant intégrer Swift dans leurs cours.Il est utile aux professionnels et aux personnes en reconversion souhaitant comprendre les apps mobiles..

PRÉREQUIS

- Connaissances de base en informatique et usage d'un ordinateur.
- Aucune expérience préalable en programmation exigée (un plus si déjà initié).
- Accès à un Mac avec Xcode installé recommandé pour la pratique.

© 2026 Smartfuture — App Development with Swift – Associate

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Planification, conception et théorie

Résumer le cycle de conception

- Brainstorming, planification, prototypage, évaluation.
- Résumer comment les données sensibles peuvent être protégées et compromises
- Partage d'informations personnelles et d'application.
- Défis de sécurité dans les apps et les services en ligne.
- Impacts juridiques, éthiques et socio-économiques du numérique.

Navigation dans le projet

Différencier les types de fichiers de base

- Identifier les principaux types de fichiers dans un projet Xcode (code, interface, ressources, etc.).

Reconnaitre les ressources disponibles dans un projet

- Comprendre le rôle des images, icônes, polices et autres assets dans une app.

Définir comment les ressources sont utilisées

- Lier et utiliser correctement les ressources dans les vues et le code.

Importer une ressource dans un projet et l'utiliser correctement

- Ajouter de nouveaux assets dans Xcode et vérifier leur intégration dans l'app.

Sélectionner les actions appropriées pour masquer ou afficher différentes zones de l'interface utilisateur

- Gérer les panneaux et zones de travail dans Xcode pour optimiser la productivité.

Interface Builder / iOS

À partir d'un scénario, sélectionner l'objet ou les objets appropriés sur le storyboard ou dans l'Outline

- Choisir les contrôles UIKit adaptés (labels, boutons, champs de texte, images, etc.).

Utiliser l'inspecteur d'attributs pour modifier non programmatiquement les propriétés des objets et/ou d'une vue

- Changer les textes, couleurs, contraintes et autres réglages d'interface.

© 2026 Smartfuture — App Development with Swift – Associate

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Connecter des objets UIKit du storyboard à un fichier Swift

- Différencier un IBOutlet d'un IBAction.
- Déterminer quand connecter un objet comme IBOutlet ou IBAction.

Modifier par programmation les propriétés des objets et/ou d'une vue

- Mettre à jour les contenus, les styles ou le comportement des vues par le code Swift.

Utilisation du langage Swift

Écrire, appeler et/ou évaluer l'exécution de fonctions

- Évaluer l'utilisation des labels d'arguments, des paramètres et des valeurs de retour.

Calculer les résultats en utilisant divers opérateurs

- Utiliser les opérateurs arithmétiques, logiques et de comparaison en Swift.

Créer et évaluer des structures

- Déclarer les propriétés d'une structure.
- Initialiser les propriétés d'une structure.
- Définir des méthodes.
- Créer une instance de structure.
- Utiliser une instance de structure..

Créer et manipuler des tableaux

- Déclarer et/ou initialiser un tableau avec des valeurs.
- Identifier et/ou modifier un élément de tableau à l'aide de son index.
- Utiliser et/ou évaluer les propriétés et/ou méthodes des tableaux.

Démontrer comment contrôler le flux d'exécution

- Créer, analyser et prédire les structures de boucle et leurs résultats.
- Créer et interpréter le résultat d'instructions conditionnelles.

Créer, utiliser et/ou comparer des énumérations personnalisées

- Définir des enums Swift et les exploiter dans le code.

© 2026 Smartfuture — App Development with Swift – Associate

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Déclarer et/ou évaluer des constantes et variables de différents types de données

- Différencier constantes et variables.
- Appliquer l'inférence de type.
- Utiliser le typage explicite.

Utiliser les conventions de nommage appropriées

- Utiliser correctement le camelCase.
- Appliquer les règles de nommage des identifiants Swift.

Débogage

Utiliser l'inspecteur de connexions pour évaluer si une erreur de connexion s'est produite

- Vérifier les connexions entre storyboard et code (IBOutlets / IBActions).

À partir d'un scénario d'erreur de connexion, déterminer une solution

- Identifier les références manquantes, dupliquées ou incorrectes.

Différencier les erreurs de syntaxe et d'exécution lors de la construction et de l'exécution d'une app

- Analyser les erreurs signalées par le compilateur et celles apparaissant à l'exécution.

Interpréter les messages d'erreur de la console

- Lire et comprendre les messages de la console Xcode pour diagnostiquer les problèmes.

Reconnaître l'objectif des points d'arrêt (breakpoints)

- Utiliser les breakpoints pour inspecter l'état de l'application pendant l'exécution.

© 2026 Smartfuture — App Development with Swift – Associate