

# Swift, les bases du langage

Cours Pratique de 2 jours - 14h

Réf : FIW - Prix 2025 : 1 480€ HT

Nouveau langage de programmation créé par Apple, Swift se présente comme une alternative à Objective-C, moderne et plus performante. Vous apprendrez dans cette formation à développer des applications Swift en environnement MacOS. Vous découvrirez les concepts de base de Swift et ses aspects novateurs.

## OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Appréhender les concepts de base du langage Swift

Créer un projet Swift

Utiliser la Swift Standard Library

## TRAVAUX PRATIQUES

Exercices pratiques sous MacOS X et XCode 8.

## LE PROGRAMME

dernière mise à jour : 05/2024

### 1) Swift

- Pourquoi un nouveau langage ?
- Ses apports et nouveautés.
- Les playgrounds : terrains de jeu.
- Intégration aux outils existants.
- Swift système ouvert.

### 2) XCode l'environnement de développement

- Présentation de l'outil et ses add-ons.
- L'utilisation de XCode.
- Interface Builder et les divers outils disponibles.
- Le débogueur.
- Les outils de mesure : instruments.
- Les portails du développeur Apple.

*Travaux pratiques : Mise en place de l'environnement de développement et premier exercice avec XCode et Swift. Utilisation du débogueur.*

### 3) Les bases du langage Swift

- Les fonctionnements du compilateur et l'architecture.
- Les bases : variables, constantes et tuples, les chaînes de caractères.
- L'optional et sa gestion, l'unwrapped.
- Les opérateurs et la surcharge.
- Les structures de contrôle.
- Les closures et les fonctions.
- Classes, structures, énumérations, propriétés.
- L'héritage et les protocoles.

*Travaux pratiques : Réalisation de plusieurs applications sur le langage et ses basiques.*

### 4) La bibliothèque Swift

- Que contient la Swift Standard Library ?

## PARTICIPANTS

Développeurs, chefs de projet.

## PRÉREQUIS

Connaissance des bases algorithmiques. Pratique d'un langage de programmation structurée. Il n'est pas nécessaire d'être développeur Apple.

## COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

## MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

- Gestion des tableaux, dictionnaires et Set.
- Les protocoles Equatable et Comparable : utilisation.
- Autres classes et structures disponibles.

*Travaux pratiques : Mise en oeuvre des protocoles Comparable et Equatable.*

### 5) Concepts avancés en Swift

- Les Subscripts : concept et implémentation.
- La gestion des erreurs et exceptions : try, catch, throw, throws, try, defer.
- La gestion de l'asynchronisme avec Grand Central Dispatch.
- La classe Thread, endormir un thread, annuler un thread, notification.
- Les références strong, weak et unowned.
- Le Lazy Loading.
- Créer des extensions en Swift.
- La généricité et les génériques.

*Travaux pratiques : Mise en oeuvre de la généricité et des extensions.*

### 6) Interaction Swift et Objective-C

- Comparaison des syntaxes entre Objective-C et Swift.
- Comparaison des types de données.
- Mixing and matching.
- Héritage entre Swift et Objective-C.
- Points particuliers et limitations.

*Travaux pratiques : Mise en oeuvre du mix and matching avec XCode.*

## LES DATES

---

Nous contacter