

## Programme de formation

# Swift Programmation (iOS, iPhone, iPad, macOS)

### Maîtrisez Swift et ses différences par rapport à Objective-C

#### Durée

4 jours / 28 heures

#### Description

Swift est le nouveau langage de programmation innovant proposé par Apple pour remplacer progressivement Objective C. Il a été conçu pour être accessible au plus grand nombre des développeurs avec une dimension à la fois ludique et professionnelle. Sa syntaxe concise et expressive le rend facile à lire et à maintenir. Les performances du langage Swift sont supérieures à celle d'Objective-C mais les deux langages peuvent tout à fait cohabiter au sein d'un projet Xcode pour vos applications Cocoa et Cocoa Touch. Cette formation Swift vous permettra bien sur de maîtriser la syntaxe du langage (variables, opérateurs, structures de contrôles, etc.), mais aussi sa philosophie, les particularités de l'environnement Xcode pour Swift. Les spécificités de Swift et notamment les différences avec Objective-C seront également abordées.

#### Objectifs pédagogiques

- La syntaxe du langage (variables, opérateurs, structures de contrôles, etc.)
- Sa philosophie, les particularités de l'environnement Xcode pour Swift
- Les spécificités de Swift et notamment les différences avec Objective-C

#### Public

Cette formation de programmation Swift s'adresse aux développeurs, chefs de projet ou toute personne désirant s'approprier la philosophie de Swift

#### Pré-requis

Bonnes connaissances de la programmation et d'un langage orienté objet (C++, Java, C#, PHP5 objet, etc.)

#### Méthodes pédagogiques

50 % théorie / 50 % pratique

#### Profil intervenant

L'ensemble de nos formations sont animées par des formateurs expérimentés possédant une expérience terrain éprouvée.

#### Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers d'ateliers de mise en pratique des notions et concepts abordés pendant la formation.

## Programme

### Xcode

- Xcode est l'IDE (Integrated Development Environment) officiel d'Apple, pour élaborer des applications iOS ou macOS
- Prise en main de l'outil (projet, documentation, debug, etc.)

### Premier Aperçu de Swift

- Swift reprend le meilleur du C et d'Objective-C en laissant de côté les contraintes respectives de ces langages
- Le rôle de Swift
- Les motivations qui ont poussé Apple à le développer
- Les problématiques que doit résoudre Swift
- Les nouveautés de Swift 4

### Constantes et Variables

- Le nouveau langage d'Apple propose une approche originale des variables
- Apprendre comment Swift gère différents types de valeurs
- Gestion des optionnels et des génériques
- Utilisation des variables

### Opérateurs avec Swift

- Swift est doté de la majorité des opérateurs classiques présents dans d'autres langages
- Nous en ferons le tour et découvrirons également d'autres spécificités de Swift

### Chaînes de caractères avec Swift

- Le remplaçant d'Objective C offre une gestion moderne et performante des caractères et des chaînes
- Nous aborderons les différentes manières de les manipuler (et notamment les changements d'API et simplifications introduites par Swift 4 à ce niveau).

### Collections

- Swift fournit 3 principaux types de collections, les tableaux, les dictionnaires, et les ensembles
- Nous verrons comment les employer pour gérer des ensembles de données

### Contrôle de l'exécution

- Le langage Swift fournit les mécanismes standard de contrôle de flux, similaires à ceux présents en C, mais comprend aussi des dispositifs supplémentaires pour assurer l'exécution des différentes branches de votre code

### Les Fonctions en Swift

- Les fonctions sont des morceaux de code autonomes qui effectuent des tâches spécifiques
- Nous regarderons leur mise en œuvre dans Swift ainsi qu'une caractéristique nommée "tuple" qui permet à une fonction de retourner plusieurs valeurs

### Clôtures (closures)

- Les clôtures (ou fermetures) sont le pendant Objective-C des "Blocks". Ce sont des blocs de code autonomes qui peuvent être passés en argument dans l'appel de fonction ou de méthodes
- Nous révélerons leur potentiel

### Classes et Structs

- Les classes constituent le cœur des API Apple pour le développement iOS et macOS
- Vous étudierez ce qu'est une classe et comment les concevoir. Nous verrons aussi le principe d'héritage qui permet de les affiner en les hiérarchisant. Swift fait également beaucoup usages des structures (qui sont beaucoup plus évoluées que celles du C).

### Méthodes

- Les méthodes sont des fonctions associées à une classe ou à un type particulier
- Nous montrerons comment les définir et les exploiter

### Gestion de la mémoire

- Apple a nettement simplifié les opérations de gestion de mémoire dans nos applications. Il reste cependant utile de comprendre comment cela fonctionne
- Nous examinerons les natures de références et les principes de comptage et jetterons un d'œil à ARC (Automatic Reference Counting), qui est LE système automatique de gestion mémoire d'Apple
- Le debugger visuel de mémoire

### Playground (terrain de jeu)

- Les "terrains de jeu" (playgrounds) rendent la rédaction des programmes Swift facile et fructueuse
- Tapez une expression et le résultat apparaît dans l'instant
- Voyons comment profiter des terrains de jeu pour expérimenter avec des morceaux de code Swift

### Synergie avec Objective-C

- Vous pouvez enrichir avec du code Swift d'anciens projets en préservant le code Objective-C. Nous étudierons certaines des techniques utilisables pour entremêler Objective-C et Swift dans un même projet.