

## Objectifs

Comprendre ce qu'est le langage Swift. Appréhender les différences entre la syntaxe Swift. Disposer des connaissances nécessaires à la création d'un projet en Swift.

## Participants

Développeurs

## Prérequis

Avoir de bonne base de programmation

## Pédagogie

La pédagogie est basée sur le principe de la dynamique de groupe avec alternance d'apports théoriques, de phases de réflexion collectives et individuelles, d'exercices, d'études de cas et de mises en situations observées. Formation / Action participative et interactive : les participants sont acteurs de leur formation notamment lors des mises en situation car ils s'appuient sur leurs connaissances, les expériences et mettront en oeuvre les nouveaux outils présentés au cours de la session.

## Profil de l'intervenant

Consultant-formateur expert sur cette thématique. Suivi des compétences techniques et pédagogiques assurée par nos services.

## Moyens techniques

Encadrement complet des stagiaires durant la formation. Espace d'accueil, configuration technique des salles et matériel pédagogique dédié pour les formations en centre. Remise d'une documentation pédagogique papier ou numérique à échéance de la formation.

## Méthodes d'évaluation des acquis

Exercices individuels et collectifs durant la formation. Evaluation des acquis et attestation de fin de stage adressés avec la facture.

## Programme

### Vue d'ensemble des systèmes d'exploitation OS X

Présentation des frameworks Cocoa et Cocoa Touch

Gestion de la mémoire avec Cocoa

Objective C et Swift

Présentation de l'IDE XCode

Le mode playground

### Bases du langage Swift

Instructions en Swift

Variables et constantes

Annotations des types de constante et de variable

Sécurité des types

Déduction des types

Alias de types

## Les types numériques

Les booléens

Les entiers (integer)

Nombres à virgule flottante

Nombres littéraux

Conversion entre types

## Les opérateurs

Opérateur d'assignement

Opérateurs arithmétiques

Opérateurs d'affectation combinés

Opérateurs de comparaison

Opérateurs d'intervalles

Opérateurs logiques

Opérateurs avancés

## Chaines et caractères

Chaines de caractères littéraires

Mutabilité d'une chaîne

Comparaison de valeurs textuelles

Chaines en Unicode

## Types de collections

Mutabilité des collections

Notion de tuple

Les tableaux

Les dictionnaires

Les littéraux de type conteneur (tableau et dictionnaire)

## Flux de contrôle

Les boucles (for, while)

Gestion des conditions (if, switch, where...)

Gestion du transfert de contrôle (continue, break, fallthrough...)

Les fonctions

Les fermetures

## Focus sur les fonctions

Paramètres et valeurs de retour

Noms de paramètres

Valeurs des paramètres par défaut

Paramètres variadiques

Paramètre in et ou

Types de fonctions

Fonctions imbriquées

## Focus sur les fermetures

Syntaxe spécifique associée

Faire passer une fermeture en argument avec les trailing closures

Capturer des valeurs

## Les énumérations en Swift

La syntaxe

Les propriétés de type « lazy-stored »

Les valeurs associées

## Structures et classes

Propriétés

Propriétés « Lazy-Stored » associées

Définir des property observers pour gérer les propriétés

Les méthodes d'instance

Méthodes reliées à des types

Les subscripts

Concept d'héritage

Notion d'overriding

Type casting en Swift

Initialisation et initialiseurs

Désinitialisation

Types imbriqués

Extensions

## Méthode de l'ARC (Automatic Reference Counting)

Présentation et intérêt

Cycles de référence

Références faibles

Références « unowned »

## Les « optionnels »

Unwrapping forcé

Binding

Processus d'enchaînement d'optionnels

## Les protocoles

Syntaxes et pré-requis



Utiliser les protocoles en tant que types

Notion de délégation

Collections de protocoles-types

Héritage de protocoles

Composition de protocoles

## Les génériques dans Swift

Fonctions d'un générique

Paramètres de type

Types de générique

Contraintes

Types associés

## Débogage

Les assertions

Le LLDB et le REPL de Swift