

## Formation Objective-C

|   |  |
|---|--|
| <b>Durée :</b>                                | 5 jours  |
| <b>Public :</b>                               | Développeurs - Chefs de projet                                   |
| <b>Pré-requis :</b>                           | Notions de programmation   |
| <b>Objectifs :</b>                            | Apprendre à développer des applications complètes en Objective-C |
| <b>Sanction :</b>                             | Attestation de fin de stage mentionnant le résultat des acquis   |
| <b>Taux de retour à l'emploi:</b>             | Aucune donnée disponible   |
| <b>Référence:</b>                             | PRO731-F   |
| <b>Note de satisfaction des participants:</b> | Pas de données disponibles                                       |

### Introduction

Historique, versions  
Parenté C et Smalltalk  
Environnement de développement  
IDE et outils (Compilateur, débogueur)  
La documentation

### Atelier: Installer Xcode

### Premiers pas dans l'Objective-C

Organisation du Code  
Les différents fichiers (\*.h, \*.m)  
Méthode main()  
Les conventions  
Les pointeurs  
Le framework Foundation

### Syntaxe de base

Variables : déclaration, utilisation  
Opérateurs  
Les constantes  
Les booléens  
Instructions conditionnelles  
Instructions d'itération, les boucles

### L'objet en Objective-C

L'objet générique id  
Allocation mémoire et initialisation  
Gestion de la mémoire  
Les Objets : NSString, NSNumber, NSMutableString, etc...  
Les interfaces  
L'encapsulations et accesseurs  
Les méthodes d'instance et de classe  
Mots clés super et self  
Héritage

### **Atelier : Réalisation d'un mini jeu**

### **Les tableaux**

Les tableaux : NSArray, NSMutableArray  
Les dictionnaires : NSDictionary, NSMutableDictionary  
Initialisation, allocation mémoire  
Parcourrir un tableau rapidement : NSEnumerator

### **Les Protocoles**

Définition d'un protocole  
Les avantages  
Création et application (delegate)  
Les méthodes optionnelles et requises

### **Manipuler les fichiers**

La classe NSURL  
Lire et écrire dans un fichier  
Gérer les erreurs  
Archiver et Desarchiver un objet  
Le protocole NSCoder  
Les classes NSCoder, NSKeyedArchiver, NSKeyedUnarchiver

### **Atelier : Enregistrer nos données dans un fichier**

### **L'interface graphique**

Le framework AppKit  
Présentation d'Interface Builder  
Les fichiers \*.xib  
IBAction et IBOutlet  
Les objets graphiques : NSTextField, NSButton, etc

### **Atelier : Réaliser une interface graphique avec Cocoa**