



SARL au capital de 200 000 €
SIRET: 518 123 245 00012 - RCS 518 123 245 (Lille) - TVA: FR37518123245
33, rue du Bastion Saint André, Appt 233 - Bat 2, 59000 Lille
Tel: 06.27.69.58.14 - www.dlta-studio.com - contact@dlta-studio.com
N° activité: 32 59 09435 59

Offre de formation 2019

Version 1.4.1

Table des Matières

Introduction	5
DLTA Studio	6
1. <i>Présentation de la société</i>	6
2. <i>Nos références</i>	6
Tarifs intra-entreprise	7
Conditions générales de vente	8
Introduction au développement mobile	11
Architecture des applications mobiles	13
Développer pour Google Android	15
Développement Android avancé	17
Kotlin pour les développeurs objets	19
Introduction à Objective-C et au développement sous iOS	21
Swift pour développeur objet	24
Développer pour iOS en Swift	26
Objective-C pour développeur Objet	28
Développer pour iOS en Objective-C	29
iOS : Développement avancé en Objective-C	31
Les API HTML5 et CSS3	33
Introduction à Javascript	35
Développer avec Angular	37
Développer avec AngularJS	39

Développer avec Node.js	41
Développer des WebServices en Java	43
Java - Développement avancé et nouveautés JDK 8	44
C++ pour les développeurs objet	46
C++ Avancé pour développeur objet	48

Introduction

Ce document présente l'offre de formation de la société DLTA Studio.

DLTA Studio

1. Présentation de la société

DLTA Studio SARL a été fondée en 2009 par David Scrève et Patrice Trognon, après plus de 17 ans passés à développer des logiciels sur quasiment toutes les plateformes logicielles et matérielles du marché : Windows, Linux, Macintosh, Windows Mobile, Android, iPhone.

Le métier de DLTA Studio est centré autour de la formation et le développement de logiciels.

Fondée par des architectes de métier, notre société est habituée à travailler dans des architectures techniques hétérogènes et ne sacrifiera jamais une bonne architecture aux dépens de points techniques de réalisation.

Nous connectons quotidiennement pour nos clients des architectures mobiles à base d'iPhone, Android ou Windows Phone aux infrastructures serveurs types SOA à base de Java, PHP, postgres, Oracle ou mysql.

Notre offre de formation s'inscrit en complément d'une activité projet importante et s'appuie sur cette expérience pour proposer des formations qui vous permettent de mettre directement en pratique les points abordés dans nos cursus.

Depuis 2009, DLTA Studio a développé une offre de 16 cursus centrés sur la formation des développeurs et des architectes.

Ces formations sont dispensés en inter et intra-entreprises, via des organismes de formation ou en direct.

2. Nos références



Tarifs intra-entreprise

Les tarifs ci-dessous sont applicables dans le cadre d'un financement via votre OPCA ou OPACIF. Pour tout autre cadre de financement, veuillez nous consulter.

Ils sont applicables pour des sessions se déroulant au sein des locaux de l'entreprise cliente, pour des groupes de 1 à 10 stagiaires.

Ce tarif n'inclus pas les frais de location de salle ou de matériel ni les frais de repas en sus.

Code	Intitulé du cours	Durée	Prix catalogue
DEV-MOB-IN	Introduction au développement mobile	2 jours	2800 €
ARCH-MOB-IN	Architecture des applications mobiles	4 jours	5600 €
GAND	Développer pour Google Android	5 jours	7000 €
DLTA-GAND-AV	Développement Android avancé	3 jours	4200 €
KOTLIN	Kotlin pour développeur objet	2 jours	2800 €
IOS-PRG	Introduction à Objective-C et au développement sous iOS	5 jours	7000 €
SWIFT	Swift pour développeur objet	3 jours	4200 €
IOS-SW	Développer pour iOS en Swift	4 jours	5600 €
OBC-IN	Objective-C pour développeur objet	2 jours	2800 €
IOS-OBJC	Développer pour iOS en Objective-C	4 jours	5600 €
IOS-OBJC-AV	iOS : Développement avancé en Objective-C	4 jours	5600 €
HT5-CSS3	Les API HTML5 et CSS3	3 jours	4200 €
JVS-IN	Introduction à Javascript	3 jours	4200 €
DLTA-ANGULAR	Développer avec Angular	4 jours	5600 €
DLTA-NODEJS-IN	Développer avec Node.js	4 jours	5600 €
JVS-ANG	Développer avec AngularJS	3 jours	4200 €
DLTA-WS	Développer des WebServices en Java	3 jours	4200 €
JAV-AV	Java - Développement avancé et nouveautés JDK 8	3 jours	4200 €
CPP-IN	C++ pour les développeurs objet	5 jours	7000 €
CPP-ADVA	C++ Avancé pour développeur objet	5 jours	7000 €

Les prix s'entendent HT. TVA 20% en sus.

Conditions générales de vente



SARL au capital de 200 000€
SIRET: 518 123 245 00012 - RCS 518 123 245 (Lille) - TVA: FR37518123245
33, rue du Bastion Saint André, Appt 233 - Bat 2, 59000 Lille
Tel: 06.27.69.58.14 - www.dlta-studio.com - contact@dlta-studio.com
N° activité: 32 59 09435 59

Conditions générales de vente Régies par les articles L. 441-6 du Code de commerce.

Les conditions générales de vente décrites ci-après détaillent les droits et obligations de la société DLTA Studio et de son client et s'appliquent à toute offre et vente de biens ou de services. Toute prestation accomplie par DLTA Studio implique donc l'adhésion sans réserve de l'acheteur aux présentes conditions générales de vente.

Si l'une des clauses des présentes conditions générales de vente se trouvait nulle ou annulée, les autres clauses n'en seraient pas pour autant annulées. Le fait que DLTA Studio ne fasse pas application à un moment donné d'un quelconque article des présentes conditions, ne peut être interprété comme valant renonciation à se prévaloir ultérieurement des dites conditions générales de vente.

Clause n° 1 : Devis, Commande

Toute commande, y compris celle passée par téléphone ou par télécopie, doit faire l'objet d'une confirmation écrite. La commande pourra s'effectuer en retournant le devis de DLTA Studio dûment revêtu du cachet commercial, signé et revêtu de la mention manuscrite « bon pour accord » ou en adressant un document mentionnant notamment : la nature du service, la quantité, le prix convenu, les conditions de paiement, le lieu de la prestation ainsi que le lieu de facturation pour une prestation ; la quantité, la marque, le type, les références, le prix convenu, les conditions de paiement, le lieu et la date de livraison ou de l'enlèvement s'il est autre que le lieu de facturation, pour une marchandise.

Pour être réputé valable, tout additif ou modification de la commande ne lie DLTA Studio que si elle l'a accepté par écrit. Si, lors d'une précédente commande, l'acheteur s'est soustrait à l'une de ses obligations (défaut ou retard de paiement, par exemple), un refus de vente pourra lui être opposé, à moins que cet acheteur ne fournisse des garanties satisfaisantes ou un paiement à la commande. Aucun escompte pour paiement anticipé ne lui sera alors accordé.

Clause n° 2 : Prix

Les prix sont ceux en vigueur au jour de la prise de commande. Ils sont libellés en euros et calculés hors taxes. Par voie de conséquence, ils seront majorés du taux de TVA et des frais de transport applicables au jour de la facturation.

DLTA Studio s'accorde le droit de modifier ses tarifs à tout moment. Toutefois, elle s'engage à facturer les marchandises/ prestations commandées aux prix indiqués lors de l'enregistrement de la commande.

Clause n°3 : Escompte

Aucun escompte ne sera consenti en cas de paiement anticipé.

Clause n° 4 : Modalités de paiement

Le règlement des commandes s'effectue par virement bancaire sur le compte de DLTA Studio. le cas échéant d'autres moyens de paiement peuvent être précisés dans le devis ou le contrat. Un acompte peut être prévu lors de la signature du devis ou du contrat et mentionné sur lesdits documents. Le solde sera payé selon les dispositions prévus à la clause numéro 5

Clause n° 5 : Délais de paiement

En l'absence de dispositions particulières prévues au devis ou dans le contrat le paiement se fait à réception de facture. Dans tous les cas les délais de paiement se font dans le cadre de la loi à savoir :

SARL au capital de 3000€ SIRET: 518 123 245 00012 - RCS 518 123 245 (Lille) - TVA: FR37518123245 33, rue du Bastion Saint André, Appt 233 - Bat 2, 59000 Lille Tel: 06.27.69.58.14 - www.dlta-studio.com - contact@dlta-studio.com N° activité: 32 59 09435 59
--

- Le délai convenu entre entreprises pour régler les sommes dues ne peut dépasser 45 jours fin de mois ou 60 jours à compter de la date d'émission de la facture ;
- En cas de facture périodique (récapitulative), le délai de paiement ne peut dépasser 45 jours à compter de la date d'émission de la facture ;

Clause n° 6 : Retard de paiement

En cas de défaut de paiement total ou partiel à la date prévue lors de la commande, l'acheteur doit verser à DLTA Studio une pénalité de retard égale à trois fois le taux de l'intérêt légal.

Le taux de l'intérêt légal retenu est celui en vigueur au jour de la livraison des marchandises.

Cette pénalité est calculée sur le montant TTC de la somme restant due, et court à compter de la date d'échéance du prix sans qu'aucune mise en demeure préalable ne soit nécessaire.

En sus des indemnités de retard, toute somme, y compris l'acompte, non payée à sa date d'exigibilité produira de plein droit le paiement d'une indemnité forfaitaire de 40 euros due au titre des frais de recouvrement.

Clause n° 7 : Clause résolutoire

Si dans les quinze jours qui suivent la mise en oeuvre de la clause " Retard de paiement ", l'acheteur ne s'est pas acquitté des sommes restant dues, la vente sera résolue de plein droit et pourra ouvrir droit à l'allocation de dommages et intérêts au profit de DLTA Studio.

Clause n° 8 : Propriété intellectuelle

DLTA Studio est propriétaire de l'ensemble des supports de formation. Aussi, toute reproduction et divulgation à des tiers de tout ou partie de ces supports, sous quelque forme que ce soit, est interdite.

Clause n° 9 : Responsabilité

DLTA Studio ne pourra en aucun cas être déclaré responsable d'un préjudice financier, commercial ou d'une autre nature, causé directement ou indirectement par les prestations de formation fournies.

Clause n° 10 : Condition d'annulation d'une session de formation

Toute demande d'annulation doit être formulée par courrier et par email. Si l'annulation intervient avant le 20ème jour ouvré avant le début de la formation, celle-ci ne générera pas de facturation. Dans le cas contraire, DLTA Studio se réserve le droit de facturer l'intégralité de la formation

Clause n° 11 : Force majeure

La responsabilité de DLTA Studio ne pourra pas être mise en oeuvre si la non exécution ou le retard dans l'exécution de l'une de ses obligations décrites dans les présentes conditions générales de vente découle d'un cas de force majeure. À ce titre, la force majeure s'entend de tout événement extérieur, imprévisible et irrésistible au sens de l'article 1148 du Code civil.

Clause n° 12 : Tribunal compétent

En application du décret n°2015-282 du 11 mars 2015, une tentative de résolution amiable des différends par la partie demanderesse est obligatoire avant tout recours judiciaire, en cas de litige concernant tant la validité, l'interprétation, l'exécution des présentes ou de leurs accords subséquents que la responsabilité des parties, est soumis au droit français et à défaut d'accord amiable, de la compétence exclusive du Tribunal de Commerce de Lille

Introduction au développement mobile

Code : DEVMOB-IN

Objectifs pédagogiques

- Appréhender les problématiques liées au développement mobile
- Découvrir les architectures mobiles
- Gérer un projet mobile
- Connaitre les environnements techniques de développement

Niveau requis

- Connaissance du développement logiciel, utilisation de terminaux mobiles

Public concerné

- Architecte, développeurs, chef de projet, Product Owner

Durée

- 2 jours

L'écosystème mobile

- Architecture des applications mobiles
- Plateformes matérielles et logicielles disponibles
- Topologie d'application

Présentation d'iOS

- Historique
- Architecture des applications
- Les outils
- Les langages de programmation
- Bonnes pratiques
- Les différents SDK
- Déploiement d'application

Présentation d'Android

- Historique
- Architecture des applications
- Les outils
- Les langages de programmation
- Bonnes pratiques
- Les différents SDK
- Déploiement d'application

Méthodologie projet

- Spécificités du développement mobile
- Conduite de projet
- Mise en place des équipes
- Outils de développement

Architecture des applications mobiles

Code : ARCH-MOB-IN

Objectifs pédagogiques

- Comprendre les enjeux du développement mobile
- Connaître les différentes architectures mobiles
- Se familiariser avec les principales plateformes mobile
- Connaître les différentes topologies d'application mobile
- Savoir éviter les principaux pièges d'un projet mobile

Niveau requis

- Connaissances de l'informatique, de la gestion de projet.

Public concerné

- Développeurs, chefs de projet, architecte.

Durée

- 4 jours

Introduction à la mobilité

- Définition de la mobilité
- Historique
- Plateformes matérielles
- Etat du marché
- Les outils
- Les spécificités liées au développement mobile

Architecture SOA

- Les architectures traditionnelles
- Pourquoi une nouvelle architecture ?
- Les architectures SOA

Base de données

- Qu'est ce qu'une base de données ?
- Base de données SQL
- Autres bases de données

Services REST

- Principes de l'approche REST
- Le protocole HTTP
- Le format JSON

Offre de formation

- Le contrat de service
- Les bonnes pratiques

Topologie des applications mobiles

- Web Responsive
- Web Embarquée
- Native
- Native hybride
- Nos recommandations

Méthodologie projet

- Spécificités du développement mobile
- Conduite de projet
- Mise en place des équipes
- Outils de développement

Développer pour Google Android

Code : GAND

Objectifs pédagogiques

- Maîtriser Android Studio et les SDK fournis par Google
- Maîtriser les différents états d'une application
- Concevoir une interface graphique
- Gérer les données de l'application locales et distantes
- Accéder aux capteurs du smartphone
- Déployer sur Google Play.

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, C++, C#).

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 ou 5 jours

Environnement de développement

- Android Studio
- Gradle
- ADT
- Le simulateur

Architecture d'une application

- Le fichier AndroidManifest.xml
- Les Activités
- Les Services
- Les Receivers
- Les Fournisseurs de contenus
- Les Ressources

Interface graphique

- Les Layouts
- Création d'une IHM
- Les composants graphiques
- Les boîtes de dialogue

Ecrans multiples

- Activités multiples
- Fragments
- La bibliothèque de support

Interface graphique avancée

- WebView
- ActionBar
- NavigationDrawer
- NavigationDrawer & ActionBar
- TabHost
- ViewPager

Réseau & Service Web

- Interfaces Http
- Services Web asynchrones
- Services REST

Persistance de données

- Stockage clé-valeur
- Fichiers
- Base de données

APIs et fonctionnalités multimédia

- Affichage de document
- Prise de photo
- Sons
- Gestion des SMS
- Push Notification
- Centre de notification
- Géolocalisation
- Cartographie avec GoogleMap
- Impression
- Graphisme
- Code barre

Déploiement

- Préparation
- GooglePlay
- En entreprise
- Internationalisation

Développement Android avancé

Code : DLTA-GAND-AV

Objectifs pédagogiques

- Découvrir les fonctionnalités avancées du développement sous Android.

Niveau requis

- Connaissance et maîtrise du développement sous Android Studio.

Public concerné

- Développeurs, architectes, chefs de projets techniques souhaitant s'améliorer en développement sous Android

Durée

- 3 jours

Environnement de développement

- Android Studio
- Gradle
- ADT

Utilisation de Gradle

- Utilisation en ligne de commande
- Structure d'un projet (modules, sources, configuration)
- Plug-ins
- BuildType et Flavor
- Création d'AAR
- Signature numérique
- Gestion des dépendances
- Génération de la Javadoc
- Utilisation des repositories d'entreprise
- Scripts Kotlin

Qualité

- Qualité de code : Lint, Sonar, FindBugs, Android Studio
- Tests automatisés : JUnit, AndroidTest, Espresso

Obfuscation

- Proguard
- Pourquoi ?
- Quand la mettre en place ?
- Mise en oeuvre

Android Studio avancé

- Profiling des applications
- Gestion des images
- ConstraintLayout

API Avancées

- DataBinding
- RecyclerView
- BottomNavigationView
- Animations
- Reconnaissance de forme avec AndroidVision

Frameworks externes

- ButterKnife
- Retrofit / Moshi
- EventBus
- AndroidAnnotation

Langage avancé

- Java 8
- Kotlin
- C++

Kotlin pour les développeurs objets

Code : KOTLIN

Objectifs pédagogiques

- Appréhender les outils de développement pour Kotlin
- Programmer objet avec Kotlin
- Programmation fonctionnelle
- Comprendre les liens entre Kotlin et les autres langages : Java, JavaScript, C

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, C++, C#).

Public concerné

- Développeurs, architectes, chefs de projet techniques.

Durée

- 2 jours

Introduction à Kotlin

- Les variables & constantes
- Types de données
- Les opérateurs
- Les structures de contrôle
- Les fonctions
- Valeurs optionnelles
- Données structurées

Programmation orientée objet

- Classe
- Propriété
- Méthode
- Propriétés dynamiques
- Polymorphisme
- Conversion de types
- Interface
- Data Class

Kotlin avancé

- Egalité
- Classes avancées
- Chaînage optionnel
- Extension
- Destructuration
- Délégation
- Généricité
- Opérateurs & fonctions infix

Gestion des erreurs

- Fonctionnement général

Offre de formation

- Remontée d'erreur
- Interception
- Gestion des ressources

Programmation fonctionnelle

- Fonctions & types de fonction
- Fonctions imbriquées
- Closures
- Coroutines

Kotlin & les autres langages

- Kotlin au sein de Java
- Appeler du Java depuis Kotlin
- Kotlin et Javascript
- Kotlin et C/C++

Introduction à Objective-C et au développement sous iOS

Code : IOS-PRG

Objectifs pédagogiques

- Apprendre Objective-C
- Maîtriser Xcode et les outils du SDK fourni par Apple
- Ajouter des préférences à une application
- Maîtriser les différents états d'une application
- Concevoir une interface graphique
- Gérer les données de l'application locales et distantes
- Accéder aux capteurs du smartphone
- Déployer sur l'AppStore.

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, C++, C#).

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 5 jours

Objective-C

L'ide Xcode

- Organisation du code
- Fichiers .h et .m
- Imports
- Types de projet
- Compilation et exécution
- Property List (plist)

Définition de classes

- Classes et interfaces
- Encapsulation et propriétés
- Construction : alloc et init
- Méthodes et échange de messages
- Exceptions et NSError

Hiérarchies de classes

- Surchage
- Héritage et polymorphisme
- Contrats et protocoles
- Héritage de protocoles
- Méthodes optionnelles

Gestion mémoire

- Décompte de références
- Retain et release
- Auto release pool

Offre de formation

- Garbage Collection

Ns...

- Utilisation des préfixes
- Librairie NextStep
- NSString
- NSArray, NSSet, NSDictionary
- NSDate
- NSNumber et formats numériques

Présentation d'iOS

- Historique
- Stratégie produit
- Structure d'une application
- Outils et SDK

Conception d'IHM

- Le modèle MVC
- Les contrôleurs
- Interface Builder
- Les Storyboards
- Les composants graphiques
- La gestion des événements

Les collections

- TableView
- CollectionView

Bonnes pratiques

- Contexte d'utilisation
- Gestion mémoire avec iOS
- Ergonomie
- Performances
- Vie privée

Réseaux et services Web

- API réseaux
- Appel de services Web
- Traitement asynchrone
- Synchronisation avec l'IHM
- Traitement de fichiers XML

Persistance de données

- Préférences utilisateurs
- Système de fichiers
- CoreData

Capteurs

- Géolocalisation
- Accéléromètre
- Gyroscope
- Caméra / Appareil photo

Graphisme

- Dessin
- CorelImage
- Animation
- Cartographie

Applications fournies

- Agenda
- Carnet d'adresse

Déploiement

- Contraintes de validation
- Licences de développement
- Signature d'une application
- Déploiement grand public
- Déploiement en entreprise
- Le portail de développement et de publication

Swift pour développeur objet

Code : SWIFT

Objectifs pédagogiques

- Appréhender Xcode et ses outils
- Programmer objet avec Swift
- Gérer la mémoire et les références
- Comprendre les liens entre Swift et Objective C
- Appeler du code Objective C depuis Swift.

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, C++, C#).

Public concerné

- Développeurs, architectes, chefs de projet techniques.

Durée

- 3 jours

Swift et Xcode

- Xcode
- Playground
- Types de projet pour Swift
- Compilation et exécution

Particularités

- Typage
- Tuple et fonctions
- Structs
- Tableaux et dictionnaires
- Enumérations
- Chaînage
- Pattern matching
- Généricité
- Automatic Reference Counting

Définition de classes

- Classes et interfaces
- Encapsulation et propriétés
- Initialisation et désinitialisation - Subscripts
- Héritage
- Protocoles
- Extensions

Gestion des erreurs

- Type d'erreur
- Remontée / Interception
- Prémonitions
- Gestion des ressources

Programmation fonctionnelle

- Fonctions globales
- Fonctions imbriquées - Closures

Swift et Objective-C

- Utilisation de code C depuis Swift
- Intégration d'Objective C avec Swift
- Utilisation des bibliothèques Cocoa

Développer pour iOS en Swift

Code : IOS-SW

Objectifs pédagogiques

- Maîtriser Xcode et les outils du SDK fourni par Apple
- Ajouter des préférences à une application
- Maîtriser les différents états d'une application
- Concevoir une interface graphique
- Gérer les données de l'application locales et distantes
- Accéder aux capteurs du smartphone
- Déployer sur l'AppStore.

Niveau requis

- Connaissance pratique du langage Swift

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 jours

Présentation d'iOS

- Historique
- Stratégie produit
- Structure d'une application
- Outils et SDK

Conception d'IHM

- Le modèle MVC
- Les contrôleurs
- Interface Builder
- Les Storyboards
- Les composants graphiques
- La gestion des événements

Les collections

- TableView
- collectionView

Bonnes pratiques

- Contexte d'utilisation
- Gestion mémoire avec iOS
- Ergonomie
- Performances
- Vie privée

Réseaux et services Web

- API réseaux
- Appel de services Web
- Traitement asynchrone
- Synchronisation avec l'IHM
- Traitement de fichiers XML

Persistance de données

- Préférences utilisateurs
- Système de fichiers
- CoreData

Capteurs

- Géolocalisation
- Accéléromètre
- Gyroscope
- Caméra / Appareil photo

Graphisme

- Dessin
- CoreImage
- Animation
- Cartographie

Applications fournies

- Agenda
- Carnet d'adresse

Déploiement

- Contraintes de validation
- Licences de développement
- Signature d'une application
- Déploiement grand public
- Déploiement en entreprise
- Le portail de développement et de publication

Objective-C pour développeur Objet

Code : OBC-IN

Objectifs pédagogiques

- Appréhender Xcode et ses outils
- Programmer objet avec Objective-C
- Gérer la mémoire et les références
- Comprendre les liens entre C, C++, Swift
- Utiliser quelques classes de base.

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, C++, C#).

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 2 jours

Les bases d'Objective-C

- Introduction
- Présentation générale
- Déclaration d'une classe
- Déclaration de méthode
- Constructeur / Destructeur
- Foundation
- Les tableaux

Objective-C avancé

- Gestion de la mémoire
- Quelques mots clés
- Les accesseurs (getter/setter)
- Les Catégories
- Les Protocoles
- Traitement des erreurs
- Bilan : Java vs Objective C

Programmation fonctionnelle

- Introduction
- Les blocks

Objective-C & Swift

- Objective-C et C/C++
- Swift au sein d'Objective-C
- Appeler de l'Objective-C depuis Swift

Développer pour iOS en Objective-C

Code : IOS-OBJC

Objectifs pédagogiques

- Maîtriser Xcode et les outils du SDK fourni par Apple
- Ajouter des préférences à une application
- Maîtriser les différents états d'une application
- Concevoir une interface graphique
- Gérer les données de l'application locales et distantes
- Accéder aux capteurs du smartphone
- Déployer sur l'AppStore.

Niveau requis

- Connaissance pratique du langage Objective-C

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 jours

Présentation d'iOS

- Historique
- Stratégie produit
- Structure d'une application
- Outils et SDK

Conception d'IHM

- Le modèle MVC
- Les contrôleurs
- Interface Builder
- Les Storyboards
- Les composants graphiques
- La gestion des événements

Les collections

- TableView
- collectionView

Bonnes pratiques

- Contexte d'utilisation
- Gestion mémoire avec iOS
- Ergonomie
- Performances
- Vie privée

Réseaux et services Web

- API réseaux
- Appel de services Web
- Traitement asynchrone
- Synchronisation avec l'IHM
- Traitement de fichiers XML

Persistance de données

- Préférences utilisateurs
- Système de fichiers
- CoreData

Capteurs

- Géolocalisation
- Accéléromètre
- Gyroscope
- Caméra / Appareil photo

Graphisme

- Dessin
- CoreImage
- Animation
- Cartographie

Applications fournies

- Agenda
- Carnet d'adresse

Déploiement

- Contraintes de validation
- Licences de développement
- Signature d'une application
- Déploiement grand public
- Déploiement en entreprise
- Le portail de développement et de publication

iOS : Développement avancé en Objective-C

Code : IOS-OBJC-AV

Objectifs pédagogiques

- Maîtriser les concepts avancés de gestion de la mémoire
- Manipuler et combiner des contrôleurs de vues
- Dessiner et animer vos propres vues
- Persister les données de l'application via le Core Data
- Accéder aux informations personnelles
- Gérer les threads et l'asynchronisme
- Générer des actions push

Niveau requis

- Connaissance pratique du langage Objective-C et maîtrise du développement iOS de base

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 jours

Gestion avancée de la mémoire

- Rappel sur la gestion de base de la mémoire avec Objective-C.
- Utilisation de l'Automatic Reference Counting (ARC). Débrayer ARC.
- Mesurer vos applications avec Instruments (allocation et fuite mémoire, activité des threads, exécution...).
- Optimiser la gestion de la mémoire. La recherche des objets zombies.
- Les blocks : fonctionnement avancé et bonnes pratiques.

UIK avancé

- Rappel sur le contrôleur de navigation.
- Combinaisons de contrôleurs et de vues.
- Le UIPickerViewController (gestion photos et vidéos).
- Le Page View contrôleur.
- Le Split View Controllers. Popovers. In-App Mail
- Affichage avancé de collection

Réseaux et services Web

- API réseaux
- Appel de services Web
- Traitement asynchrone
- Synchronisation avec l'IHM
- Traitement de fichiers XML

Persistance de données

- Préférences utilisateurs
- Système de fichiers

Offre de formation

- CoreData

Graphisme

- Dessin
- CoreImage
- Animation

API avancées

- Impression
- Agenda
- Carnet d'adresse
- Achats intégrés

Notifications

- La gestion des notifications push
- Le process de gestion des notifications : serveur, APNs Apple et sur le mobile
- Le fonctionnement Apple Push Notification Service
- Les certificats spécifiques "ready to push notification"
- Envoi de messages "push" via le payload en JSON

Les API HTML5 et CSS3

Code : HT5-CSS3

Objectifs pédagogiques

- Connaître les API HTML5 et leur utilité
- Valider des formulaires avec WebForms 2
- Dessiner avec Canvas et SVG
- Gérer la déconnexion
- Communiquer avec les
- WebSockets
- Gérer les tâches longues avec les WebWorkers
- Réaliser des animations avec CSS3
- Comprendre le responsive design et les liens entre HTML5 et la mobilité.

Niveau requis

- Pratique du langage JavaScript et maîtrise de HTML4

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 3 jours

Le Web version HTML5

- Définition et limites de HTML5
- Support des navigateurs
- Modernizr et bibliothèques pour la compatibilité
- Impact sur les architectures Web
- HTML5 pour les mobiles

Structure des pages HTML5

- Simplifications
- Doctype
- Balises sémantiques - Micro formats

WebForms 2

- Nouveaux champs de saisie
- Sliders, datalist et placeholder - Expressions régulières
- Validation automatique
- Validation dans le code

Multimédia et graphisme

- Audi et vidéo
- Canvas
- SVG
- WebGL

Communications

- XHR2
- CORS
- JSON
- Messaging
- WebSocket

WebWorkers

- Modèle mono-thread
- Worker API
- Synchronisation
- Shared Workers

Fichiers et ressources locales

- LocalStorage
- SessionStorage
- ApplicationCache
- IndexedDB
- File API

Device API

- Géolocalisation
- Orientation
- Batterie
- Caméra et micro
- WebRTC

CSS3

- Fonts
- Sélecteurs CSS3
- Bordures
- Couleurs et opacité
- Transitions et transformations
- Animations

Responsive Design

- Vision OneWeb
- Responsive Web Design
- Progressive Enhancement
- Media Query

Introduction à Javascript

Code : JVS-IN

Objectifs pédagogiques

- Connaître les bases de JavaScript et du DOM
- Juger de l'intérêt de jQuery pour la programmation cross-browser
- Gérer les événements et les manipulations dynamiques
- Savoir les règles d'or de la - programmation avec JavaScript
- Réaliser des appels synchrones (Ajax).

Niveau requis

- Connaissance de XHTML et CSS.

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 3 jours

Un langage pour le Web

- Historique de JavaScript
- JavaScript et Ajax
- Bibliothèques JavaScript - Avenir de JavaScript

Bases du langage

- Syntaxe de base
- JSON
- Pièges du typage dynamique
- Programmation objet
- Constructeurs et this
- Fonctions et programmation fonctionnelle - Objet Window ou le contexte global
- Contextes d'exécution

Document Object Model

- Les objets du navigateur
- Récupérer des éléments
- Ajouter des balises
- Modifier les contenus
- Manipulation des CSS
- Gestion des événements
- Cross browsing
- Amélioration avec jQuery

Ajax

- Architecture des sites Web avec et sans Ajax - Asynchronisme dans le navigateur
- XMLHttpRequest et HTML5
- Gestion d'erreurs et timeout
- Ajax avec jQuery

jQuery

- Les bibliothèques jQuery, Dojo, Sencha ... - Fonctions indispensables
- Effets visuels
- jQuery et le DOM
- Chaînage des appels
- Fonctions avancées : proxy, merge et extend
- Widgets
- jQuery Mobile

Développer avec Angular

Code : DLTA-ANGULAR

Objectifs pédagogiques

- Découvrir et mettre en pratique le langage TypeScript
- Connaître les problématiques des Single Page Application (SPA)
- Savoir y répondre avec Angular
- Utiliser et définir des composants, des modules et des directives
- Accéder au serveur depuis Angular
- Gérer la navigation entre les vues et l'historique
- Lier les composants et les modèles.
- Utiliser les formulaires et les différents types de binding
- Tester les composants automatiquement

Niveau requis

- Avoir une connaissance pratique de développement Web et d'un langage orienté objet.

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 jours

Introduction à TypeScript

- Déclaration de variable
- Structures de contrôles
- Typage statique
- fonction

TypeScript orienté objet

- Classes
- Interface
- héritage & polymorphisme

TypeScript avancé

- Modules
- Namespaces
- Enumérations
- Génériques
- Décorateurs
- Mixins

Principes de base d'Angular

- Application SPA
- Model View Controller : les types de MVC
- Le MVC à la mode Angular
- Bénéfices du Two Way Data Binding
- Injection de dépendances
- Directives

Architecture d'une application Angular

- Modules
- Services
- Routage et gestion de l'historique
- Événements

Angular Avancé

- Gestion des formulaires
- Appels REST
- Framework REST

Angular : un Framework testable

- Tests unitaires avec Angular
- ngMock
- End to End testing
- Karma Test Runner

Développer avec AngularJS

Code : JVS-ANG

Objectifs pédagogiques

- Connaître les problématiques des Single Page Application (SPA)
- Savoir y répondre avec AngularJS
- Utiliser et définir des directives
- Accéder au serveur depuis Angular
- Gérer la navigation entre les vues et l'historique
- Lier les composants et les modèles.

Niveau requis

- Avoir une connaissance pratique de JavaScript et jQuery

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 3 jours

Architecture d'une SPA

- Rôles du client et du serveur
- Accès aux données par un service REST
- Angular côté client
- JSON au milieu
- Gestion de l'état applicatif
- Synchronisation des données
- Navigation dans une application mono-page - Gestion des URL

Principes de base d'Angular

- Model View Whatever : les types de MVC
- Le MVC à la mode Angular
- Bénéfices du Two Way Data Binding
- Injection de dépendances

Figures imposées

- Gestion des formulaires
- Angular Templates et expressions
- Navigation, hashbang et deeplinking
- Accès serveur simplifié
- Contrôleurs et modèles
- Décoration par les CSS
- Internationalisation
- Routage et gestion de l'historique

Particularités d'Angular

- Processeur HTML - Filtres
- Directives
- Scopes

Offre de formation

- Modules - Providers - Services

Les directives en détail

- - Directives pré-définies
- - Directives Custom
- - Scope et cycle de vie
- - Accès au DOM, événements et templates - Transclusion

Angular : un Framework testable

- - Tests unitaires avec Angular
- - ngMock
- - End to End testing
- - Karma Test Runner
- - Protractor

Angular partout ?

- Angular comparé aux autres frameworks MVC
- Modularité et applications multi-vues
- Angular UI
- Intégration d'Angular avec d'autres librairies
- Modularité d'Angular
- Les limites d'AngularJS

Développer avec Node.js

Code : DLTA-NODEJS-IN

Objectifs pédagogiques

- Comprendre l'architecture de Node.js et ses enjeux
- Développer des serveurs Web
- Développer des applications REST et les faire interagir avec des clients SPA (Angular) • Déployer des applications Web avec Node.js

Niveau requis

- Connaissance pratique du développement Web
- Maîtrise de Javascript
- Maîtrise d'un framework SPA, idéalement Angular ou ReactJS

Public concerné

- Développeurs, architectes, chefs de projets techniques.

Durée

- 4 jours

Présentation de Node.js

- Historique
- Objectifs
- Fonctionnalités
- Architecture asynchrone • Outils

Premiers pas

- Exécution d'un script
- Utilisation de modules
- Quelques modules standards : process, path, fs, Events, Streams etc...
- Streams

Programmation Web

- Modules dédiés au développement Web: http, dns, url, net, dgram • Création d'un client Web
- Création d'un serveur Web :
- Traitement des requêtes : GET, PUT, POST
- Gestion des formulaires et utilisation de formidable

Programmation avancée

- Templating avec EJS et Pug
- Connexion aux bases de données : SQLite, Postgres, Mysql, Redis, MongoDB. • Traitement réseau avec SocketIO
- Intégration d'Angular dans les projets

Déploiement

- Tests automatisés
- Installation Mocha, Chai et Sinon.js • Configuration
- Exécution des tests
- Construction de build avec Grunt et Browserify
- Intégration d'une application SPA dans le déploiement

Développer des WebServices en Java

Code : DLTA-WS

Objectifs pédagogiques

- Comprendre les Services Web
- Appréhender les services REST
- Développer des services REST en Java
- Déployer & tester des services REST

Niveau requis

- Connaissance pratique de Java et du monde Web

Public concerné

- Développeurs, architectes, chefs de projet techniques.

Durée

- 3 jours

Comprendre les services Web

- Architecture
- Enjeux et utilisations
- Contrat de service

Appréhender les services REST

- REST : Signification
- Le protocole HTTP
- Les framework Java

Développer des services REST

- Introduction à Jersey
- Les annotations
- traitement des erreurs
- Le format JSON

Déployer & tester des services REST

- Déploiement sur tomcat
- Tests : Outils et méthodes

Java - Développement avancé et nouveautés JDK 8

Code : JAV-AV

Objectifs pédagogiques

- Maîtriser le multi-threading
- Gérer l'asynchronisme
- Créer vos propres annotations
- Utiliser la réflexion
- Interagir avec du code natif
- Intégrer le scripting dans vos applications
- Optimiser la gestion mémoire
- Connaître les nouveautés des derniers JDK.

Niveau requis

- Avoir une connaissance pratique du langage Java

Public concerné

- Développeurs, architectes et chefs de projets techniques.

Durée

- 3 jours

Nouveautés des JDK 7 et 8

- Diamond Operator
- Lambda expressions et streams
- Le try-with-resource et l'interface AutoCloseable
- NIO 2 et Non Blocking IO

Threading et concurrence

- Thread et runnable
- Problèmes du multithreading
- Attente et synchronisation
- Thread Pools et environnements "Thread safe"
- Package java.util.concurrent
- Le Framework Executor
- Asynchronisme en Java
- Future et Callable

Annotations et réflexion

- Annotations prédéfinies
- Le processeur d'annotations
- Définition d'interface (@Interface)
- Traitement à la compilation ou à l'exécution (@Retention)

Offre de formation

- Cible des annotations (@Target)
- Cas d'utilisation des annotations
- L'API de Reflection
- Chargement et appel dynamique
- DynamicProxy
- Aller plus loin avec l'AOP

Appels natifs

- JNI vs JNA
- Appel natif depuis Java
- Déclaration des signatures et utilisation de JNAerator
- Passage de paramètres
- Chargement des bibliothèques natives
- Intégration Java dans un processus natif

Scripting

- Cas d'utilisation
- BeanShell, JavaScript et autres langages compatibles
- Intégration des interpréteurs et ScriptEngineManager
- Java Scripting API
- Exécution d'un script
- Accès aux fonctions et méthodes
- Contraintes sur les variables et paramètres

Optimisation

- Gestion mémoire et WeakReference
- Paramétrage de la mémoire et du GarbageCollector
- Pistes dans le code
- Pistes à la compilation
- Pistes à l'exécution
- Outils de diagnostic

C++ pour les développeurs objet

Code : CPP-IN

Objectifs pédagogiques

- Revoir les basiques de la programmation orientée objet
- Définir ses propres classes
- Découvrir la copie d'objet
- Appréhender le polymorphisme
- Gérer la mémoire
- Gérer les exceptions
- Créer et manipuler les opérateurs
- Manipuler les flux
- Utiliser la généricité et les templates

Niveau requis

- Connaissance pratique d'un langage de programmation objet (Java, Swift, C#).

Public concerné

- Développeurs, architectes, chefs de projet techniques.

Durée

- 5 jours

La programmation orientée objet

- Objet / classe
- Polymorphisme
- Héritage et décapsulation
- Compilation et exécution

Définir ses propres classes

- Constructeurs
- Destructeur
- Attributs
- Méthodes

Héritage & Polymorphisme

- Les différents types d'héritage
- Utilisation de la ZIM
- Chaînage des destructeurs

Gestion de la mémoire

- Contexte static
- Opérateurs new et delete

Gestion des exceptions

- Lever des exceptions

Offre de formation

- Intercepter des exceptions
- Gestion des ressources
- Bonnes pratiques

Les opérateurs

- Les principaux opérateurs
- Surcharge d'opérateur

Les flux

- Les flux standards
- Les fichiers
- Les opérateurs associés

Généricité

- Les templates
- Template de fonction
- Template de classe

C++ Avancé pour développeur objet

Code : CPP-ADVA

Objectifs pédagogiques

- Gérer la mémoire & les autoPtr
- Appréhender le polymorphisme
- Découvrir la copie d'objet
- Créer et manipuler les opérateurs
- Découvrir la STL & ses design patterns
- Gérer les exceptions
- Utiliser les espaces de nom
- Manipuler les pointeurs de fonctions
- Utilisation du RTTI
- Découvrir les nouveautés de C++11 et la programmation multithread

Niveau requis

- Connaissance pratique du C++

Public concerné

- Développeurs, architectes, chefs de projet techniques.

Durée

- 5 jours

Gestion de la mémoire

- Contexte static
- Opérateurs new et delete

Opérateurs

- Les principaux opérateurs
- Surcharge d'opérateur

STL & Design patterns

- Les conteneurs
- Les algorithmes
- Les itérateurs

Gestion des Exceptions

- Lever des exceptions
- Interceptor des exceptions
- Gestion des ressources
- Bonnes pratiques

Fonctionnalités avancées

- Namespace
- RTTI

Offre de formation

- AutoPtr

Nouveautés de C++11

- Multithreading
- Inférence de type
- nullptr