

CODE : F2AS3

TITRE : ActionScript 3.0 - La synthèse

DUREE : 1 jour (soit 7 heures)

PRIX : 695 € HT par participant HORS CURSUS
359 € HT si cursus STANDARD 5 jours
340 € HT si cursus PROFESSIONNEL 7 jours
323 € HT si cursus ENTREPRISE 10 jours

DESCRIPTION : Flex 2.0 est arrivé avec une nouvelle version du langage qui permet de la programmer, ActionScript 3.0. Si vous êtes débutant, vous connaîtrez l'essentiel de ce langage objet ; confirmé, vous aurez une synthèse des nouveautés. Idéal pour les programmeurs Java ou C# qui souhaitent rapidement être opérationnels.

AUDIENCE : Développeurs Flex souhaitant connaître la syntaxe d'ActionScript 3.0 ; développeurs Java ou C# ou C++ souhaitant rapidement retrouver leurs marques en ActionScript 3.0.

PRE-REQUIS : Connaissance d'un langage de programmation objet C++, Java, C#...

INSCRIPTIONS AU 01 73 02 33 44

CALENDRIER F2AS3 INTER-ENTREPRISE PARIS

04 Février 2008

12 Septembre 2008

10 Mars 2008

06 Octobre 2008

04 Avril 2008

26 Novembre 2008

05 Mai 2008

13 Juin 2008

Utiliser ActionScript 3.0 avec l'outil Flex 2

PROGRAMME

La syntaxe d'ActionScript 3.0

Tester AS3 avec trace() et un composant MXML
 Déclarer et utiliser les scalaires
 Comprendre les types de données et leur conversion
 Comprendre les opérateurs et leurs priorités
 Créer des tableaux indexés et associatifs pour manipuler des données complexes
 Utiliser l'objet tableau (Array) et les classes de collection

Les structures de contrôle d'ActionScript 3.0

Manipuler les chaînes de caractères (String)
 Comprendre et utiliser les opérateurs logiques et de comparaison
 Travailler avec les dates et l'heure en utilisant la classe Date
 Contrôler le flot d'exécution avec if/else et switch<:case
 Itérer des portions de code (créer des boucles)
 Créer des fonctions pour réutiliser et encapsuler des fonctionnalités

Les classes en ActionScript 3.0

Comprendre la terminologie de la technologie objet
 Définir des classes constructeur, méthode et propriétés
 Contrôler l'accès aux membres d'une classe
 Instancier (construire) des objets à partir des classes
 Comprendre le pattern de conception "Value Object"
 Implémenter implicitement et explicitement les getters et les setters
 Etendre une classe pour créer une classe plus spécialisée
 Redéfinir les méthodes d'une super-classe dans une sous-classe
 Appeler les méthodes d'une super-classe
 Créer une classe de service pour créer ou gérer des objets