

Catalogue des formations 2010

Plans de cours et calendrier

Pythagore F.D.
11, rue du Faubourg Poissonnière
75009 PARIS
Tél : 01 55 33 52 10 – Télécopie : 01 55 33 52 11 – Site : www.pythagore-fd.fr

Pythagore F.D. : Apprendre à Apprendre

Nos domaines d'expertise :

- Java, serveurs d'applications J2EE (**Jboss, Websphere, Jonas**),
- TCP/IP (Architecture, Sécurité, Administration de réseaux IP, **VoIP**, ...)
- Unix (**AIX, HP-UX, Solaris**),
- Linux (**Debian, Mandriva, RedHat, Fedora, Suse**), et les applicatifs Apache, Openldap, Samba, Squid, Nagios, OpenOffice, ...

Sur chacun de ces domaines, notre équipe possède une excellente maîtrise : nos formations vont de l'introduction à Linux, à la Sécurité, la Haute Disponibilité, ou le clustering Jboss.

Tous nos stages sont l'occasion de nombreuses mises en pratique et exercices, de manière à permettre aux participants de bien assimiler les nouveaux concepts.

Notre centre de formation :

Sur le plan pratique, notre centre de formation est situé 11, rue du Faubourg Poissonnière, à Paris (9^e), à deux pas des grands boulevards, à proximité de nombreux restaurants, hôtels, etc

...

Du côté technique, nos salles sont équipées de manière à pouvoir tester et mettre en oeuvre tous les concepts abordés en cours :

- 7 salles de formation,
- 60 postes clients PC 2Ghz, 1Go Mo de mémoire, équipés de deux interfaces 100Mbps sur commutateur et une interface Wifi 802.11g+ à 108Mbps
- serveurs AIX (versions 4.3 et 5.2), serveurs HP-UX (version 11), serveurs Sun UltraSparc (Solaris 2.9)
- pour les formations VoIP, téléphones Linksys SPA901 avec configuration par synthèse vocale, avec support SIP v2, Codecs vocaux: G.711, G.723.1, G.729a
- Accès réseau internet haut débit sur tous les postes (23 Mo de bande passante)

Nos différentes formules :

Nos formations sont proposées en **inter-entreprises**, mais également en intra sur site ou dans nos locaux.

Si vous avez des besoins particuliers, n'hésitez pas à faire appel à nos services : nous pouvons élaborer des plans de formation et des supports de cours sur mesure.

De même, si vous souhaitez bénéficier d'un accompagnement dans la mise en oeuvre des techniques acquises, nous vous proposons des consultants qui pourront vous guider dans vos premiers pas.

Pythagore F.D. : Apprendre à Apprendre

Notre méthode pédagogique :

nos formations sont essentiellement basées sur des travaux pratiques.

Pour chaque chapitre, le déroulement est le suivant :

- présentation et explication par le formateur des nouveaux concepts
- démonstration sur des exemples
- réalisation par les participants de petits exercices d'application

Dès qu'un ensemble de notions cohérent a été étudié, un exercice plus global est réalisé, qui permet de mettre en oeuvre les différents points vus auparavant.

Notre démarche qualité :

Depuis la préparation des plans de cours, jusqu'à la réalisation des travaux pratiques, en passant par la veille technologique et la rédaction des supports, tous les éléments sont réunis afin de vous garantir **l'excellence technique et pédagogique.**

Les résultats obtenus sont excellents : la meilleure preuve est le taux de satisfaction des participants à nos formations.

Les nouveautés 2010 :

Dans le domaine du système : les formations "Programmation sur Android", ou sur iPhone, auxquelles viennent s'ajouter des formations sur l'embarqué "Linux Temps Réel et Embarqué" et "openWRT"

dans le domaine des réseaux : une refonte de notre gamme de formations sur l'administration avec "Supervision Nagios", "Administration Nagios" et "Performances et optimisation des réseaux";

dans le domaine Java, pour accompagner les évolutions du monde JEE, un stage sur la migration vers les EJB 3;

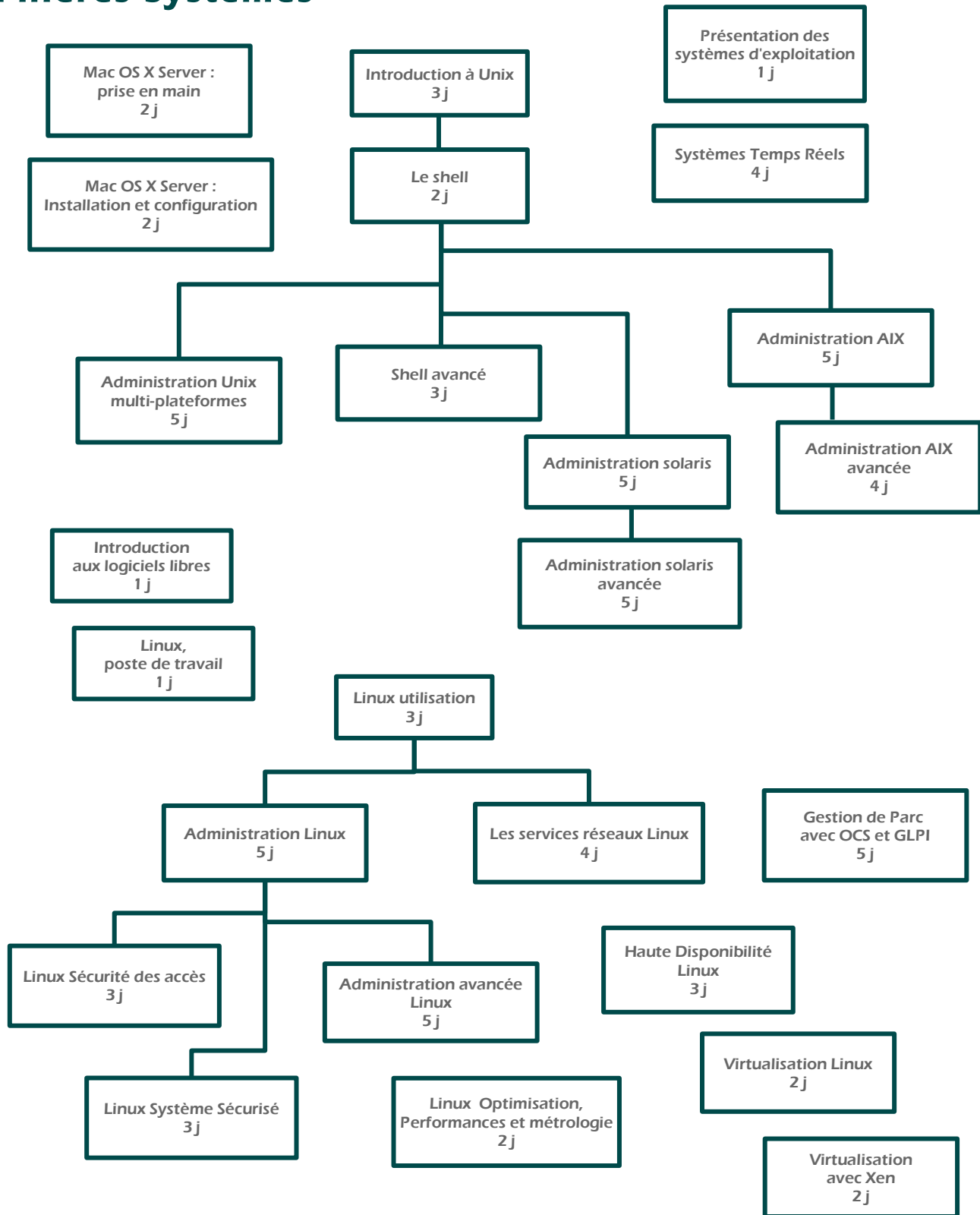
l'applicatif Internet n'est pas en reste avec une nouvelle formation sur Joomla!

Au total, nous vous proposons cette année plus de 100 stages pour répondre à vos besoins de formation, et d'approfondissement des nouvelles technologies

Pour toute demande d'information, n'hésitez pas à nous contacter :

01 55 33 52 10
www.Pythagore-fd.fr

Filières Systèmes



Présentation des systèmes d'exploitation

Durée: 1 jour

Prix et dates: nous consulter

Public:

Toute personne souhaitant connaître la notion de système d'exploitation, ses fonctionnalités et ses champs d'application.

Objectifs:

Comprendre le rôle et les principes de fonctionnement des systèmes d'exploitation.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

- Introduction** : Architecture d'un ordinateur : les différents éléments (processeurs, mémoire, périphériques, etc ...)
Les différentes fonctions à assurer : interface utilisateur, calcul, stockage
- Le système d'exploitation** : Notion de ressources.
Le rôle du système : gestion des périphériques, des fichiers, des connexions, etc ...
- Différents systèmes** : Quelques définitions : mono-utilisateur, mono-tâche, multi-utilisateurs, multi-tâches, etc ...
Positionnement de quelques OS : MVS, VM, VMS, Unix, Windows 98, NT, 2000, 2003.
- L'exploitation** : Les différentes phases dans l'entretien du système : notions d'installation, de configuration, d'administration, etc ...

Systèmes temps réel

Durée: 4 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant acquérir des connaissances de base sur les systèmes temps réel.

Objectifs:

Connaître les principes de fonctionnement des systèmes temps réel.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

- | | |
|------------------------|--|
| Introduction | : Définition d'un système temps réel. Les services offerts. Historique.
Caractéristiques des systèmes embarqué. Critères de choix: coût. Présentation de VxWorks. |
| Définitions | : Langages: C, assembleur, ADA
Horloge, Entrées/Sorties, automate, séquenceur |
| Fonctionnement | : Le contexte d'exécution. Les interruptions d'horloge. Alarmes. Timer cycliques et non cycliques.
Automate à états. |
| Ordonnancement | : Notion de processus, de ressource, de thread.
analyse et découpage des tâches.
Moniteur préemptif et non préemptif. Concurrence et coopération. Stratégies d'ordonnancement
Impact sur la gestion de contexte
Communication synchrone entre tâches, verrous. Section critique. Sémaphores.
Producteur et consommateur.
TP: mise en place d'un mini-moniteur réparti. |
| Entrées/Sorties | : Mappées, accès direct. Pré traitement d'E/S. alarmes sur seuil.
Processeurs dédiés.
E/S analogiques. |

Virtualisation avec Xen

Durée: 2 jours
990 € HT

les 4 et 5 Février
les 22 et 23 Avril
les 10 et 11 Juin

les 2 et 3 Septembre
les 14 et 15 Octobre
les 2 et 3 Décembre

Public:

Toute personne souhaitant mettre en place la virtualisation avec Xen.

Objectifs:

Savoir configurer et installer Xen.

Connaissances préalables nécessaires:

Une bonne connaissance des systèmes d'exploitation est nécessaire.

Programme:

- | | |
|--------------------|---|
| Xen Présentation | : Introduction aux solutions de virtualisation Xen.
Présentation de l'architecture Xen :
l'hyperviseur et les systèmes hôtes. |
| Installation | : Travaux pratiques d'installation à partir des RPMs
Configuration et démarrage.
Création d'un domaine. Arrêt d'un domaine. |
| Administration | : Les outils d'administration : Xend, Xm
Mise en œuvre de la console d'administration :
configuration de domaines, du réseau. |
| Mise en production | : La gestion des LVM, des processeurs
Sauvegarde et restauration de domaines
La sécurisation d'une architecture Xen |

Introduction à Unix

Durée: 3 jours
1130 €

du 1er au 3 Février
du 22 au 24 Mars
du 3 au 5 Mai
du 7 au 9 Juin

du 19 au 21 Juillet
du 6 au 8 Septembre
du 11 au 13 Octobre
du 29 Novembre au 1er
Décembre

Public:

Toute personne désirant utiliser Unix.

Objectifs:

Maîtriser les commandes de base et l'utilisation de l'éditeur Vi. Manipuler les fichiers. Gérer la sécurité des données.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

- Généralités** : Historique. Présentation du système. Ouverture et fermeture d'une session de travail.
Syntaxe générale des commandes.
- Organisation des données sur un serveur Unix** : L'arborescence. Organisation logicielle d'un serveur Unix .
Le système de fichiers
- Les fichiers** : Les types de fichiers. Chemin d'accès et nom de fichier.
Manipulations de fichiers et de répertoires
Recherche de fichiers : la commande find
- La sécurité** : Le fichier des utilisateurs et le fichier des groupes
Le mode d'un fichier. Modification des permissions.
- Les processus** : Gestion de la mémoire et des processus. Caractéristiques d'un processus
Processus en arrière-plan. Les travaux batch.
- Les redirections** : Les redirections d'entrées et sorties standards
Les filtres. Le mécanisme des tubes.
Exemples de commandes filtres. Grep et expressions régulières
- Les outils** : L'éditeur vi
Les sauvegardes

Le Shell

Durée: 2 jours
750 €

du 4 au 5 Février
du 25 au 26 Mars
du 6 au 7 Mai
du 10 au 11 Juin

du 22 au 23 Juillet
du 9 au 10 Septembre
du 14 au 15 Octobre
du 2 au 3 Décembre

Public:

Les développeurs d'applications sur UNIX, les personnes chargées de la mise en production des applications, les exploitants.

Objectifs:

Maîtriser la programmation en bourne Shell et en Korn Shell.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX.

Programme:

Généralités	: Modes d'exécution d'un script Les alias Les métacaractères
Les entrées/sorties	: Lecture Ecriture
Structures de contrôle	: Les instructions conditionnelles Gestion des boucles
Compléments	: Les fonctions Aperçu de AWK.

Shell avancé

Durée: 3 jours
1200 €

du 22 au 24 Février
du 6 au 8 Avril
du 21 au 23 Juin

du 20 au 22 Septembre
du 18 au 20 Octobre
du 13 au 15 Décembre

Public:

Les développeurs d'applications sur UNIX, et personnes chargées de la mise en production des applications, les exploitants.

Objectifs:

Acquérir une bonne connaissance du shell. Savoir écrire des interfaces, comme des menus de pilotage des applications et du système.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX .

Programme:

- Rappels** : Fonctionnement du shell. Les commandes de base.
- Variables** : Portées des variables: locales, globales, environnement.
Transmission de paramètres. Evaluation en ligne, tableaux.
Variables spéciales d'environnement.
Exploitation de la variable IFS.
- Structure** : Expressions mathématiques, expressions conditionnelles
Structure: tests avancés, boucles, choix multiples.
Fonctions, structuration des scripts shell. Exportations.
Autoprogrammation. Langages de macros: m4
- Exécution** : Programmation parallèle, programmation événementielle,
gestion des alertes. Gestion des travaux.
Exemple: création d'un système de surveillance de processus.
- Exploitation** : Redirections, interruption d'une procédure.
Suivi de consommation. Gestion des répertoires.
Simplification des environnements de travail.
- Programmation événementielle** : Signaux
Debugging, Tests
- Gestion de données** : Exploitation de la commande 'awk': Recherches et calculs sur
des fichiers. Expressions régulières.
Variables et fonctions. Opérateurs de calcul.
Jointure et comparaison de fichiers: join, diff, diff3, sed

Scripts Shell pour administrateurs

Durée: 2 jours

Prix et dates: nous consulter

Public:

Administrateurs de systèmes unix

Objectifs:

Maîtriser les aspects du shell utiles pour la création des procédures d'administration.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX et du shell.

Programme:

- Rappels** : Fonctionnement du shell. La programmation de scripts.
- Le besoin d'automatisation** : Les procédures d'administration : création d'utilisateur, surveillance des fichiers, sauvegardes, ...
- Gestion des utilisateurs** : Script d'automatisation de création d'utilisateur
- Sauvegardes** : rsync via réseau
dd du mbr
svn des fichiers de configuration
- Supervision** : Les objets à superviser
Commandes de base
Automatisation, et appel depuis un outil de supervision
- Gestion de données** : Exploitation de la commande 'awk': Recherches et calculs sur des fichiers
Variables et fonctions. Opérateurs de calcul. Expressions régulières.
Utilisation de IFS

Administration UNIX

Durée: 5 jours
1980 € HT

du 8 au 12 Février
du 12 au 16 Avril
du 14 au 18 Juin

du 30 Août au 3 Septembre
du 25 au 29 Octobre
du 6 au 10 Décembre

Public:

Exploitants et administrateurs de systèmes UNIX.

Objectifs:

Connaître les principes de fonctionnement système UNIX. Configurer et administrer un système (AIX, Solaris, HP-UX, Linux).

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du système Unix.

Programme:

- Introduction** : Rappels sur le rôle de l'administrateur.
- Installation** : Installation du système : les différentes étapes, ajout de logiciel, utilitaires de livraison
- Démarrage et arrêt du système** : Procédure de boot (/etc/inittab), lancement des processus, montages au boot.
Arrêt/relance du système (shutdown, halt, reboot, sync).
Commandes de base. Aide en ligne.
- Outils graphiques** : Outils spécifiques : smit, wsm, sam, admintool, smc.
Webmin : présentation, installation, configuration.
- Gestion des Utilisateurs** : Principes de base
Fichier /etc/passwd, les shadow passwd
Groupes, droits d'accès, politique d'accès
adduser, useradd, passwd
Contrôle des accès, contrôle des modifications. Quotas.
Fichiers .profile et .cshrc. Choix d'un shell.
- Systèmes de fichiers** : Définitions : partitions, inodes.
Organisation, gestion et maintenance :
mkfs, mount/umount.
Montage automatique.
Visualisation des file system.
Maintien de l'intégrité: fsck.

Administration UNIX

- Sauvegardes** : Définitions: sauvegardes, archivages, migrations.
Utilisation des sauvegardes; périodicité
incrémentales, complètes; restaurations
Outils sauvegarde/archivage/compression:
tar, dump, restore, dd, cpio, rsync
Sauvegarde du système
- Programmation de tâches** : Besoin, définition
Outils : at, cron, utilisation de la crontab.
- Services d'impression** : Lancement/arrêt des services d'impression.
déclaration d'une imprimante. /etc/printcap
gestion des files d'attente
- Réseau** : Les différents paramètres à prendre en compte. Configuration
du nom de l'adresse du système
Commandes génériques. Visualisation des paramètres locaux.
Routage. Fichiers de définitions statiques.
- Services de noms** : DNS : introduction, configuration du service
- NFS** : Partage de fichiers, mise en place d'un serveur, configuration
client, sécurisation NFS
TP: mise en place d'un partage de fichiers entre plusieurs
machines
- Automonteur** : Principe et travaux pratiques avec /home
- Idap : authentification centralisée** : Fonctionnement de base, mise en œuvre avec OpenLdap
- X11** : Environnement graphique. Principe de base. XDM. Contrôle
d'accès.
- Supervision** : Mise en œuvre de rrdtool.
Création de graphes.

Administration AIX

Durée: 5 jours
1980 € HT

du 15 au 19 Février
du 26 au 30 Avril
du 7 au 11 Juin

du 6 au 10 Septembre
du 15 au 19 Novembre
du 13 au 17 Décembre

Public:

Exploitants et administrateurs de systèmes AIX.

Objectifs:

Savoir configurer, administrer un système AIX.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système unix, et le shell

Programme:

- Introduction** : Rappel : le rôle de l'administrateur.
La norme POSIX. Les systèmes UNIX, le positionnement d'AIX.
- Installation du système** : Etapes.Installation de logiciels. Suppression. Ajout de logiciels.Recherche d'un package
Procédure de démarrage du système, fichiers de démarrage : /etc/inittab
Services. Sous systèmes.
- Utilisateurs** : les fichiers de configuration, les notions d'utilisateurs, de groupes, la gestion des mots de passe. Les rôles, les quotas. Le répertoire /etc/security: /etc/security/passwd
- Système de fichiers** : les types de systèmes de fichiers : JFS, CDRFS, NFS, UFS
montage et démontage de systèmes de fichiers : le fichier /etc/filesystems.
ajout/modification des caractéristiques d'un système de fichiers.
Prise en charge des fichiers volumineux (>2Go)
Auto-Monteur.
- Gestion des volumes** : Fonctionnalités. Mise en place.
les volumes physiques : ajout, liste du contenu, modification
les groupes de volume : activation, désactivation (varyon / varyoff); importation, exportation; les groupes de volumes en miroir;
les volumes logiques : ajout, retrait, copie.
crfs. Redimensionnement.
- Périphériques** : le répertoire /dev

Administration AIX

- Impressions** : Fonctionnement. Installation. Imprimante réseau.
Les commandes d'impression, la gestion des travaux, des files d'attente, d'un serveur d'impression : Ajout d'imprimantes, configuration.
Suivi du spooler, /etc/qconfig, lpq/lpr
- Pagination** : ajout d'un espace de pagination. Modification dynamique d'un espace de pagination. Activation d'un espace de pagination
- Réseau** : Les différents paramètres à prendre en compte. Configuration de l'interface. Configuration du nom de l'adresse du système
- NFS** : partage de fichiers. Configuration d'un serveur NFS, d'un client NFS.
Format du fichier d'export. Sécurisation NFS
- Sauvegardes/ restaurations** : Utilisation des sauvegardes; périodicité. Incrémentales, complètes; restaurations. Méthodes et techniques. Les outils standards : tar, cpio
Création d'une image système. mksysb, mkcd
Lecture/vérification d'image iso
- Programmation de tâches** : at : exécution différée, cron : pour le séquençement des travaux (jobs)
Présentation des outils d'automatisation du marché
- Surveillance** : Détection des problèmes, analyse des fichiers de logs. errdemon, alog.
- Outils** : La gestion de la documentation : HYPER, Administration : SMIT, et par un navigateur : WSM, ODM : principe de fonctionnement

Administration AIX avancée

Durée: 4 jours

Prix et dates: nous consulter

Public:

Exploitants et administrateurs de systèmes AIX.

Objectifs:

Savoir optimiser la configuration d'un système AIX, Savoir contrôler les performances du système et connaître les outils de mesure et d'optimisation.

Connaissances préalables nécessaires:

Les principes d'administration d'un système AIX doivent être acquis avant d'assister à ce cours.

Programme:

- ODM** : Les bases ODM.Administration ODM.
Gestion des informations. Classes et objets. Fichiers de configurations. Création d'objets: odm_create_class, odm_add_obj, odmcreate, odmdrop.
Création d'une nouvelle base
Les commandes en mode ligne: odmdelete, odmadd, odmchange, odmshow, odmget.
- Installation AIX** : Lancement.Options d'installation.Menu post-install.Fichiers d'installation.
SMS/AIX.
- Installation réseau** : Installation AIX en réseau, principe de NIM
Mise en place d'un maître NIM: nim_master_setup, nimconfig.
Informations sur l'environnement NIM.Clients NIM.NIM avec wsm.
nimol. Mise en œuvre.
- Configuration** : Matériel installé.Configuration de matériel.Connexions ODM.Informations sur le système
Informations sur le noyau.Passer en noyau 64 bits
Console système
- Disques** : comparaison de systèmes de fichiers sur deux machines différentes.
Mise en œuvre de rdist et install. distfile.
Réparation d'une image de boot corrompue.

Administration AIX avancée

- Partitionnement physique** : Répartition des accès disques: RAID 1, RAID 5. Utilisation des LVM. Création d'une image statique d'un volume physique, remise en place de cette image avec splitvg et joinvg. Le micro-partitionnement avec DLPAR
- Partitionnement logique** : LPAR : principe. Versions.
- Accès** : Le fichier /etc/security/user. Paramètres avancés de chuser , lsuser, mkuser et rmuser . Gestion des mots de passe. Gestion des rôles. Fichier login.cfg. Programme de connexion. Définition d'une séquence SAK. Autorisation de connexion. Masque applicatif. Localisation du compte. Accès à su. Shell sécurisé/base TCB
- Performances** : La mesure des performances : les temps de réponse. Les commandes de base :sar, vmstat,iostat. Le démon sadc. Présentation des ressources et impact sur les performances : la mémoire, l'espace disque, les accès disques, la cpu. Notion de charge machine. Entrées/sorties, disques fixes, threads et mémoire. Mode de fonctionnement: serveur, multi-utilisateur ou station de travail. Restreindre l'utilisation de ressources: le fichier limits, ulimit. Utilisation du Performance Diagnostic Tool. Structure du PDT, installation et activation. Configuration avec pdt_config.
- CPU** : Commandes de base: time, timex. Mesure de la consommation de programmes: tprof. Mesure de la consommation des threads du noyau: pprof. Mesure par l'accounting: prtacct, acctprc1, acctprc2.
- Mémoire** : Utilisation de la commande vmstat. Les options -l et -s. Mesure de la mémoire utilisée: svmon. Analyse de la charge mémoire VVM avec schedtune et vmtune.
- Disques** : Commandes de base: df, du. Mesure avec acctmerge. La commande dodisk et le fichier dacct.
- Le diagnostic** : Fichiers de logs : syslog, analyse par errpt
- Traces** : Utilisation de traces systèmes. Création d'un fichier trace. Exploitation avec trcrpt, filters.

Administration Solaris 9 et 10

Durée: 5 jours
1980 € HT

du 22 au 26 Février
du 19 au 23 Avril
du 21 au 25 Juin

du 13 au 17 Septembre
du 22 au 26 Novembre

Public:

Exploitants et administrateurs de systèmes Solaris.

Objectifs:

Savoir installer, configurer, administrer un système Solaris. Les participants travailleront sur des serveurs Sparc et des serveurs x86.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système unix, et le shell

Programme:

- Introduction** : Rappels : le rôle de l'administrateur.
Introduction à Solaris 10.Principaux apports de solaris 10.
- Installation** : Installation de solaris a partir d un dvd ou du reseau .Ajout de logiciel : pkgadd
Autoconfiguration du système : principe, périphériques
- Démarrage/arrêt** : Etapes de lancement, procédure de boot
Niveaux de boot, fichier /etc/inittab
Montages au boot : /etc/vfstab
Arrêt du système : halt, reboot, shutdown
Configuration matérielle
- SMF** : Service Management Facility: Présentation, utilisation. Gestion des services.
- Utilisateurs** : Fichier /etc/passwd
Principe des shadow passwords
Création de comptes utilisateurs, valeurs par défaut
Fichiers profils , choix d'un shell : .cshrc, .profile,
gestion des roles (RBAC)

Administration Solaris 9 et 10

- Fichiers** : Les inodes. Gestion des partitions.
Les différents systèmes de fichiers, le système UFS, le système ZFS
Commandes : format, fsck
Création d'un système ZFS
Le montage du système de fichiers.
Montage automatique, mount/umount.
Les pseudo-systèmes de fichiers
les ACLS sous solaris
- Sauvegardes** : Définitions, outils standards : tar, cpio, dd
Commandes ufsdump/ufsrestore
- Ordonnancement** : At, cron, fichier crontab
- NFS** : Network File System
Installation d'un serveur de fichiers : configuration serveur, client.
Sécurisation.
- Impressions** : Configuration des services d'impression
Installation d'une imprimante
- X11** : Principe de fonctionnement,ssh forwarding
- Introduction à la virtualisation** : Zones: présentation, définitions.
Création d'une zone minimale. Création d'une zone complète.
Utilisation, administration de zones.
- Administration au quotidien:** Administration graphique.
Surveillance des logs, processus, mémoire, fichiers.
Syslog, coreadm dumpadm.

Administration avancée Solaris 9 et 10

Durée: 5 jours
1990 € HT

du 1er au 5 Mars
du 3 au 7 Mai
du 20 au 24 Septembre
du 28 Juin au 2 Juillet du 29 Novembre au 3 Décembre

Public:

Administrateurs de systèmes Solaris.

Objectifs:

Savoir installer, configurer, administrer un parc de systèmes Solaris. Gérer une infrastructure Solaris cluster. Les participants travailleront sur des serveurs Sparc et des serveurs x86.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système unix, du shell et de l'administration Solaris de base.

Programme:

Introduction	: Rappels : le rôle de l'administrateur.
Installation avancée	: Configuration d'un serveur d'installation automatisée jumpstart. flash-archive
Virtualisation sous solaris	: Xen, virtual box et Zones.
DNS et LDAP	: Installation et configuration d'un serveur DNS et d'un serveur LDAP.
Configuration système avancée	: /etc/system modload/modunload, Swap Space, Core Dumps
Analyse de performances	: et prise de traces Vmstat, iostat, mpstat sar dtrace
Filesystems et gestion de volumes	: Svm, zfs, fssnap, rappels sur les sauvegardes
NFS	: Configuration avancée et analyse de comportement
Haute disponibilité	: Installation et configuration suncluster 3.2 principes, ressources, bascules ...

MAC OS X Server : Prise en main

Durée: 2 jours
910 € HT

les 25 et 26 Janvier
les 22 et 23 Mars
les 17 et 18 Mai

les 26 et 27 Juillet
les 27 et 28 Septembre
les 22 et 23 Novembre

Public:

Utilisateurs de MAC OS X Server

Objectifs:

Savoir utiliser l'interface graphique, la manipulation des fichiers, le lancement d'application, la notion d'utilisateur.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

Introduction	: Historique Etat de l'art (version actuelle, l'avenir) Qu'est-ce que Mac OS X Server Présentation rapide de l'interface, du Finder et de Spot
Installation	: Les étapes d'installation Partitionnement du disque au moment de l'installation Personnalisation de l'installation
Configuration	: Présentation des différentes configuration Standard, workgroup, Advanced Travaux pratiques en Standard et Workgroup
Outils	: Server Preferences System Preferences Server Admin
Utilisateurs et groupes	: Création d'utilisateurs avec System Preferences
Partage de fichiers	: Avec AFP et Samba
Serveur Jabber (iChat)	: Mise en place avec Server Preferences
Serveur de Calendrier	: Mise en place avec Server Preferences
Acces distant	: Partage d'écran SSH, VPN
Sauvegarde	: Mise en place avec Time Machine Sauvegarde du serveur

MAC OS X Server : Prise en main

Logiciel Opensource : Fink/DarwinPorts

Présentation de ressources : Ou trouver des compléments d'information : Apple
externes (discussions.apple.com), mosx.org, blog de l'admin

MAC OS X Server : Installation et **MC012** configuration

Durée: 3 jours
1360 € HT

du 27 au 29 Janvier

du 28 au 30 Juillet

du 24 au 26 Mars du 29 Septembre au 1er Octobre

du 18 au 21 Mai

du 24 au 26 Novembre

Public:

Administrateurs de MAC OS X Server.

Objectifs:

Savoir installer et administrer Mac OS X Server, connaître les outils disponibles,

Connaissances préalables nécessaires:

Connaissance des mécanismes de base de MAC OS X Server, connaissances de bases sur TCP/IP

Programme:

Introduction	: Historique Etat de l'Art (version actuelle, l'avenir)
Installation sur une machine distante	: Configurations: Standard, Workgroup, Advanced.
Outils	: Server Admin. Workgroup Manager. Terminal.
Serveur DHCP	: Mise en place. Plages statiques.
Serveur DNS	: Cheminement d'une requête DNS. Mise en place. Notion de zone, d'enregistrement de zone, d'enregistrement MX, de reverse DNS.
Serveurs de courrier	: Acheminement d'un message électronique. Mise en place. Notions de POP, IMAP, SMTP. les spams, virus et solutions.
OpenDirectory	: Définition. Mise en place.
Utilisateurs et groupes	: Création d'utilisateurs réseau (home sur le serveur et préférences générales) utilisateurs nomades Création à la ligne de commande (dscl)
Serveur Jabber	: (iChat) Mise en place avec Server Admin
Serveur de calendrier	: Mise en place avec Server Admin
Acces distant	: Partage d'écran. ssh, VPN
Sauvegarde	: Mise en place avec Time Machine. Sauvegarde du serveur. Sauvegarde de client.

MAC OS X Server : Installation et MC012

Logiciel OpenSource

: Fink/DarwinPorts

Ressources externes

: Présentation, ou trouver des compléments d'information :
Apple (discussions.apple.com), mosx.org, blog de l'admin

Linux utilisation

Durée: 3 jours
1020 € HT

du 22 au 24 Février
du 26 au 28 Avril

du 25 au 27 Août
du 18 au 20 Octobre

Public:

Utilisateurs de Linux.

Objectifs:

Connaître les principes de fonctionnement du système Linux.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

- Introduction** : Présentation de Linux, définitions de base, version de noyau, distributions. Obtenir de l'information.
Rappel rapide sur l'organisation d'un système. Arrêt/relance du système (shutdown, halt, reboot, sync). Commandes de base.
Aide en ligne.
- Interface graphique** : startx. Lancement d'applications, modification des menus, mini-applications d'interfaces
Présentation de Gnome, KDE, WindowMaker, XFCE
lancement d'applications, modification des menus
- Manipulations** : Traitements sur les fichiers: cp, rm, mv, ls ...
Administration par webmin. Ajout de logiciels.
Utilisation de périphériques: lsmod, insmod, lspci
Impression : lpr, configuration depuis StarOffice
montage de cdrom, disquette (mount, /etc/fstab, automonteurs)
- Applications** : Présentation des applications courantes:
- bureautique: StarOffice/OpenOffice, GnomeOffice, kOffice, ...
- dessins: gimp, ggv, xpdf, acrobat
- utilisateur WEB: lynx, firefox, opera
- réseau: Samba, PostFix, openLdap
Applications lourdes: Serveurs JEE, WebSphere, Tivoli, Oracle
- Installation** : Déroulement d'une installation de base
Partitionnement du disque, configuration réseau, configuration carte graphique

Linux, poste de travail

Durée: 1 jour
460 € HT

le 5 Février
le 9 Avril
le 25 Juin

le 10 Septembre
le 22 Octobre
le 17 Décembre

Public:

Chefs de projet, architectes

Objectifs:

présenter les distributions linux et les outils disponibles pour le poste de travail

Connaissances préalables nécessaires:

pas de connaissance particulière

Programme:

- Introduction** : Présentation de Linux,
définitions : opensource, noyau, distributions
Les apports de Linux, les risques, la sécurité
- Interface graphique** : Les outils disponibles : selon les distributions, les standards
- Applications** : Présentation des applications courantes du poste client :
- bureautique: OpenOffice, GnomeOffice, kOffice, ...
- dessins: gimp, ggv, xpdf
- utilisateur WEB: firefox, thunderbird
Les solutions en ligne : le poste client léger
Les limites : logiciels métiers
Les évolutions des projets, les paramètres de choix de la solution logicielle
- Compatibilité windows** : Gestion de postes clients hétérogènes en entreprise :
les risques et solutions
compatibilité OpenOffice , formats de fichier, clients messagerie
- Migration** : Conseils sur l'organisation d'une migration :
les outils à migrer dans un premier temps
les précautions à prendre
- Les acteurs** : Comparaison des différentes distributions : spécificités et orientations (serveurs, développement, poste client)
La position des distributions majeures : RedHat, Debian, Ubuntu, ...

Introduction au logiciel libre

Durée: 1 jour
460 € HT

le 4 Février
le 8 Avril
le 24 Juin

le 9 Septembre
le 21 Octobre
le 16 Décembre

Public:

Chefs de projet, architectes

Objectifs:

Présenter les caractéristiques et le mode de fonctionnement du logiciel libre.

Connaissances préalables nécessaires:

pas de connaissance particulière

Programme:

- | | |
|--|--|
| Introduction | : Le principe du logiciel libre, historique.
La notion de licence, la Free Software Foundation
Les licences "ouvertes" : GPL, BSD, X11,
Les projets OpenSource : GNU, hurd, FreeBSD, ... |
| La place de l'Opensource aujourd'hui | : Quelques chiffres, les projets significatifs
Les sites de références, de documentation.
L'offre de services. Les apports, les risques. |
| Le système GNU/Linux | : Origine, historique
La place de Linux dans l'OpenSource : définitions : noyau, distributions
Solutions disponibles sous Linux : en serveurs (produits réseaux, systèmes),
sur postes clients (outils bureautiques, client web, messagerie, ...) |
| Les outils JEE | : Rappel de l'architecture JEE,
positionnement des outils opensource : apache, jetty, tomcat, JBoss, jonas, geronimo |
| Les environnements de développement | : Eclipse, Netbeans, Ant
Python, Ruby, RubyOnRails |
| Liens avec les solutions propriétaires | : Gestion de postes clients hétérogènes en entreprise :
compatibilité OpenOffice , formats de fichier, clients messagerie
Environnement de développement : plugins
Les limites : logiciels métiers |

Administration Linux

Durée: 5 jours
1980 € HT

du 8 au 12 Février
du 12 au 16 Avril
du 31 Mai au 4 Juin

du 30 Août au 3 Septembre
du 11 au 15 Octobre
du 6 au 10 Décembre

Public:

Administrateurs, et toute personne souhaitant maîtriser l'installation, la configuration d'un système Linux : Debian, Mandriva, RedHat ou Suse.

Objectifs:

Savoir installer, administrer un système Linux. Chaque participant dispose des différentes distributions et peut, si il le souhaite, tester les travaux pratiques sur le système de son choix.

Connaissances préalables nécessaires:

Des connaissances de base des systèmes unix et/ou linux sont nécessaires, ainsi que du shell.

Programme:

- Introduction** : Linux et l'opensource : historique, caractéristiques de linux
Les distributions, les différences et points communs.
Rappel rapide sur l'organisation d'un système.
Arrêt/relance du système (shutdown, halt, reboot, sync).
- Installation** : Les phases d'installation d'un système Linux.
Options dans les chargeurs : grub.
Les outils d'installation. Gestions de paquets. Les différentes méthodes.
- Environnement graphique** : Présentation, gestionnaire de fenêtres. Différentes solutions :
gnome, KDE, Windowmaker, xfce.
- Outils d'administration** : Webmin: Présentation, installation, configuration.
- Systèmes de fichiers** : Définitions : inodes, filesystem, partition
Organisation, gestion et maintenance. Maintien de l'intégrité des systèmes de fichiers : mkfs, mount, umount, df.
Les différents types de systèmes de fichiers: ext3, reiserFs, xfs, jfs.
Les autres systèmes de fichiers: fat, vfat, nfs, smb.
Partition: création d'images de partitions. Synchronisation de données. Chiffrement des données.
Organisation, gestion et maintenance. Maintien de l'intégrité
Outils d'analyse et de réparation d'un système de fichiers.
- Périphériques** : montage de cdrom, disquette (mount, /etc/fstab, automontage).

Administration Linux

- Utilisateurs** : gestion des comptes utilisateurs: adduser, useradd, passwd, /etc/profile, /etc/group, /etc/shadow, groupes, droits d'accès, politique d'accès
Contrôle des connexions de root.
Introduction à PAM : Pluggable Authentication Modules.
- Processus** : Les processus. Les threads. Gestion des priorités.
Utilisation des pseudo-processus /proc: stat, cpuinfo, ...
- Sauvegardes** : Outils sauvegarde/archivage/compression: gzip, zip, tar, dd, cpio
Sauvegarde du système, création de CD de secours.
TP: sauvegarde par cpio, d'un archivage par tar
- Impressions** : Les services d'impression, installation d'une imprimante, modification d'un pilote : lpr, cups, printtool, system-config-printer.
- Programmation de tâches** : Outils : at, cron, le fichier crontab.
Les produits du marché : openPBS, fcron
- Exploitation** : Journaux : /var/log/messages
- Réseau IP** : Configuration, nommage/activation des interfaces réseau, drivers.
les utilitaires ssh, clients windows (Putty, WinSCP)
- Intégration système d'information** : Samba: Principe. Installation et configuration de samba. Partage de répertoires, d'imprimantes, création de comptes
Ldap : authentification centralisée
Fonctionnement de base, mise en œuvre avec OpenLdap
Liens pam/ldap, samba/ldap

Administration avancée Linux

Durée: 5 jours
2020 € HT

du 22 au 26 Février
du 26 au 30 Avril
du 21 au 25 Juin

du 13 au 17 Septembre
du 15 au 19 Novembre
du 13 au 17 Décembre

Public:

Administrateurs, et toute personne souhaitant approfondir l'administration d'un système Linux.

Objectifs:

Savoir installer, administrer, faire évoluer une distribution. Ce cours a lieu sur Linux RedHat, et sur Debian pour la partie "apt". Il est essentiellement basé sur des travaux pratiques.

Connaissances préalables nécessaires:

Connaître les techniques d'administration d'un système unix ou linux.

Programme:

Distribution

: Présentation : RedHat Package Manager.
Les distributions qui utilisent les rpm.
Fonctionnalités : sécurité, méta-données, gestion des dépendances,
Mise en œuvre : installation, désinstallation, requêtes, documentation.
Construction de RPMs : depuis les sources jusqu'au package.
Description des paquets DEB: fonctionnement apt, dpkg, dselect, debconf.
L'outil apt : principe, les répertoires apt, fichiers release.
Les commandes apt-get, apt-cache.
Les frontaux apt : apt-shell, aptitude, synaptic

Démarrage/Installation

: Analyse du mode de démarrage : grub, Anaconda
Le système kickstart.
Analyse d'une image initrd. Modification d'un initrd, ajout de modules.
Création de média d'installation.
TP : boot sur un périphérique USB depuis un CD.

Systemes de fichiers journalisés

: XFS : fonctionnement, mise en œuvre, administration
compatibilité NFS

Administration avancée Linux

- LVM** : Logical Volume Manager.
Présentation. Définitions : VFS, EVMS,
Volumes physiques, groupes de volumes, volumes logiques,
extension logique.
TP: mise en place de partitions LVM. Formatage en xfs.
Utilisation des LVM : les snapshots, le redimensionnement, la
concaténation de groupes de volumes.
- RAID** : Définitions. Mise en place de RAID 0, RAID 1, RAID 0+1 et RAID
5.
Mise en évidence des reprises sur incidents.
Analyse des performances.
- Authentification** : pam: gestion des modules d'authentification.
Principe de base, configuration, mise en œuvre.
Les modules : access, chroot, cracklib, etc ...
Ldap : Lightweight Directory Access Protocol
Les modèles, la conception d'une arborescence.
Interface pam/ldap. Mise en œuvre avec Openldap et
l'automonteur
- Performances** : Suivi des ressources processeurs et mémoire: vmstat, top. Suivi
des ressources réseaux: netstat, ntop.
Gestion de la fragmentation, pagination.
Utilisation des informations de /proc/stat, /proc/cpuinfo et de
l'accounting.
- Ressources** : Les quotas disques : principe, mise en place.
- Noyau** : Compilation du noyau : présentation, mise en œuvre.
TP: mise en place des ACLs sur un système de fichiers.
Présentation de la virtualisation avec UML.
- Périphériques** : Périphériques non standards.
Installation de modules: modprobe, insmod.
Le répertoire hotplug.

Les services réseaux Linux

Durée: 4 jours
1760 € HT

du 11 au 14 Janvier
du 8 au 11 Mars
du 3 au 6 Mai

du 19 au 22 Juillet
du 4 au 7 Octobre
du 22 au 25 Novembre

Public:

Administrateurs systèmes et réseaux.

Objectifs:

Savoir installer, configurer et sécuriser les principaux services réseaux sur Linux.

Connaissances préalables nécessaires:

Des notions de base sur le système Unix ainsi que sur TCP/IP sont souhaitées.

Programme:

- Réseau : Configuration des interfaces réseau.
Installation des pilotes. Notions de modules dynamiques.
- Les services : Serveurs de configuration
dns, dhcp, bootp (présentation de kickstart)
distribution Linux en réseau
Serveurs de fichiers : nfs, ftp, tftp, http
Serveurs d'accès : routage, firewall, proxy
TP: Mise en œuvre de ftp: configuration, droits sur les répertoires, gestion des utilisateurs, surveillance, fichiers de logs
- NTP : Définition.Fonctionnement. Déclaration d'un point de synchronisation.
Configuration d'un serveur d'horloge. Configuration de clients.
Architecture. Contrôles d'accès.
- Serveurs Intranet : Messagerie: Postfix, pop3, imap4 (protocoles, configuration, logs)
DNS: bind (définitions, fonctionnement, configuration d'un serveur DNS et des postes clients)
dhcp
TP : mise en place d'une architecture type avec postfix bind et dhcp
- Serveur Web : Installation Apache : configuration de base, configuration multi-sites (httpd.conf)
Suivi: access_log, error_log
Principe des scripts CGI, et des modules.les hôtes virtuels

Les services réseaux Linux

- NFS** : présentation, fonctionnement, configuration d'un serveur NFS et des postes clients
partage de fichiers
Les groupes de confiance, méthodes de protection.
- Intégration hétérogène** : interconnexion Unix/Windows:
samba : configuration et installation
accès depuis des clients windows, et Linux
en mode texte : smbclient,
smb4K : le navigateur samba
- Sécurisation des accès réseau** : Connexion directe, distante, liste des points d'entrée dans le système.
Les 'tcp wrapper' et xinetd.
Analyse des fichiers journaux du réseau
Vérification de l'intégrité du système à l'aide des outils:
tcpdump, sniffit, cop, satan
ping, traceroute
- Administration distante** : Webmin : installation, présentation de l'interface et des fonctionnalités de webmin

Linux système sécurisé

Durée: 3 jours
1390 € HT

du 20 au 22 Janvier
du 24 au 26 Mars
du 25 au 27 Mai

du 26 au 28 Juillet
du 27 au 29 Septembre
du 8 au 10 Novembre

Public:

Toute personne souhaitant mettre en place une sécurité optimale sur un système Linux, et plus particulièrement les administrateurs système et sécurité.

Objectifs:

Savoir configurer les mécanismes de sécurité de Linux.

Connaissances préalables nécessaires:

Une bonne connaissance du système Unix/Linux est nécessaire.

Programme:

- Introduction** : Le besoin, définition du D.I.C.
- Gestion utilisateurs** : Rappels sur les notions de base de sécurité sur Unix: modes d'accès, comptes utilisateurs, groupes, utilisateurs génériques de gestion de ressources.
Fichiers /etc/passwd, /etc/group, /etc/shadow
Gestion des groupes
Vérification de cohérence : pwck.
Connexions du compte root, contrôle de connexions.
Outil de recherche de mots de passe.
Prise de privilèges: sudo, sudoers.
- Authentification** : pam: gestion des modules d'authentification.
Principe de base, modification de fonctionnement.
Les modules : access, chroot, cracklib, env, ftp, groups, limits, listfile, mkhomedir, tally, time, unix, wheel
- Sécurisation traitements** : Les risques : le déni de service, exemples de virus.
TP : exploitation d'un débordement de pile.
Les moyens de détection, la surveillance, les traces : syslog, l'accounting.
L'audit de sécurité.
- Sécurité du noyau** : Présentation de GrSecurity et SELinux. Installation, administration avec grAdm.
Mise en place des règles d'ACL. L'ACL GrSec.
Restrictions d'accès aux appels systèmes. Masquage de processus. Visibilité du répertoire /proc. Restrictions chroot.
Introduction à UML (UserModelLinux) en mode SKAS.

Linux système sécurisé

- Les données** : Contrôle du système de fichiers : fsck.
Sauvegardes :
Utilisation des sauvegardes pour la disponibilité des données.
Outils sauvegarde/archivage/compression:
gzip, zip, tar, dd, cpio
Création de CD ou disquettes de secours.
- Sécurité système de fichiers** : Sécurité: mise en place des contrôles d'accès (ACL)
TP: mise en place des ACL sur xfs
Quotas. Options de montage: nosuid, nodev, noexec, ro
TP: mise en place des quotas

Linux sécurité des accès

Durée: 3 jours
1390 € HT

du 15 au 17 Février
du 6 au 8 Avril
du 31 Mai au 2 Juin

du 25 au 27 Août
du 18 au 20 Octobre
du 1er au 3 Décembre

Public:

Toute personne souhaitant sécuriser les accès à un système Linux

Objectifs:

Savoir configurer les mécanismes de sécurité réseau de Linux.

Connaissances préalables nécessaires:

Une bonne connaissance du système Unix/Linux est nécessaire.

Programme:

- Sécurisation des accès réseau** : Vérification de l'intégrité du système à l'aide des outils: tcpdump, sniffit, cop, satan, nessus, nmap.
Protection de services réseaux au travers de xinetd.
Les tcp-wrappers: telnet, tftp, snmp, ftp, pop3s, imap4s
Les accès réseaux : sftp, les r-commandes.
Sécurisation http (apache), SSL.
- VPN , tunnels, iptables** : Définitions : DMZ, coupe-feux, proxy.
VPN et tunnels.
Présentation des tunnels chiffrés.
TP: installation d'un tunnel pour chiffrer SMTP
Pare-feux : les iptables, définition d'une politique de sécurité, mise en place.
- Proxy Squid** : Présentation, sécurité, architecture externe.
Exemple d'utilisation, systèmes d'exploitation concernés, logiciels complémentaires.
configuration manuelle, automatique. Scripts d'auto-configuration, filtrage suivant DNS, par protocole. Clients en mode texte ,robots. Installation dans le navigateur.
principe et syntaxe des ACL. Optimisation de l'utilisation du serveur. Restriction d'accès par hôte, par réseau, par plage horaire, par jour, par site.
Mise en cache des données. Méthodes d'authentification.

Haute disponibilité Linux

Durée: 3 jours
1390 € HT

du 1er au 3 Février
du 19 au 21 Avril
du 7 au 9 Juin

du 30 Août au 1er Septembre
du 11 au 13 Octobre
du 29 Novembre au 1er
Décembre

Public:

Administrateurs Linux, ou toute personne souhaitant mettre en œuvre un système Linux avec des contraintes de haute disponibilité.

Objectifs:

Connaître et savoir mettre en œuvre les mécanismes disponibles sur Linux pour offrir un service continu.

Connaissances préalables nécessaires:

Une bonne connaissance d'un système Unix est nécessaire. Des notions d'administration sont souhaitées.

Programme:

- Introduction** : Le besoin : pourquoi la haute disponibilité, mesure de la disponibilité.
Quelques définitions : tolérance aux pannes, fail-over, RAID, Mirroring, redondance, MTBF, etc ...
Les acteurs du marché, positionnement de Linux.
Présentation de l'architecture LVS.
- clustering** : Les différentes fonctions de clustering :
répartition des accès disques, répartition de la charge CPU,
basculement automatique ou programmé sur un autre processeur, exécution simultanée sur plusieurs processeurs.
- Linux Virtual Server** : Architecture : heartbeat, mon, ldirector, fake, coda
Utilisation de mon pour la détection des services défaillants.
Installation, configuration de heartbeat et ldirector
Pilotage heartbeat par l'interface graphique
- Adresses réseau** : Principe du basculement d'adresses.
Solution avec Fake.
Agrégation d'interfaces réseau
- IPVS** : Répartition de charge
Préparation d'un noyau IPVS, configuration passerelle.
Exemple de mise en place d'un cluster.

Haute disponibilité Linux

- Applications** :
- Intégration LVS avec Keepalived
 - Présentation de la RedHat Cluster Suite
 - Répartition de requêtes http, gestion des sticky session. Mise en cluster d'applications Web. Répartition de charges, routage de niveau 7. Présentation des solutions WebSphere, JBoss et Jonas.
- Données** :
- Logical Volume Manager
 - Le RAID, RAID logiciel sous Linux : raidtool, mdadm.
 - Mise en œuvre de disques SCSI partagés entre machines
 - Les systèmes de fichiers haute disponibilité : DRDB (Distributed Replicated Block Device)
 - La réplication des données avec PostGreSQL, MySQL

Linux : optimisation performances métrologie

UX119

Durée: 2 jours
990 € HT

les 4 et 5 Février
les 1er et 2 Avril
les 3 et 4 Juin

les 23 et 24 Septembre
les 2 et 3 Novembre

Public:

Administrateurs, et toute personne souhaitant connaître les éléments permettant d'améliorer les performances d'un système Linux.

Objectifs:

Connaître les points du système à mesurer. Comprendre leur impact sur les performances globales du système et savoir les adapter à un mode de fonctionnement (client, serveur, station, base de données, messagerie, ...)

Connaissances préalables nécessaires:

Une bonne connaissance d'un système Linux est nécessaire. Des notions d'administration sont souhaitées.

Programme:

- Introduction** : Qu'est ce que la gestion des performances?
- Mesures** : Les éléments à prendre en compte, les points de mesures.
TP : création d'un utilitaire d'extraction des informations système.
- Systèmes de fichiers** : Les différents types de systèmes de fichiers.
Les systèmes natifs: ext2, ext3, xfs,
Gestion de la fragmentation, pagination
Les systèmes émulés: vfat, ntfs
Les systèmes distribués: nfs, smb
Options: rsize, wsize, timeo, retrans,...
TP: outil de mesure des accès.
- Processus** : Les processus. Les threads.
Utilisation de la commande vmstat
TP: déclenchement d'une alarme en cas de surcharge système.
- Utilisateurs** : accounting, quotas, fichiers de logs.
- Réseau** : Utilisation de la commande netstat
Exploitation des éléments statistiques produits
- Outils** : Utilisation des pseudo-processus /proc: stat, cpuinfo, ...
Utilisation des processus système: kswapd, swpctl, syslogd

Virtualisation Linux

Durée: 2 jours
990 € HT

les 14 et 15 Janvier
les 11 et 12 Mars
les 10 et 11 Mai

les 12 et 13 Juillet
les 7 et 8 Octobre
les 4 et 5 Novembre

Public:

chefs de projet, administrateurs souhaitant choisir une solution de virtualisation.

Objectifs:

Connaître les différentes solutions de virtualisation sur linux, et leurs caractéristiques.

Connaissances préalables nécessaires:

Une bonne connaissance du système Unix/Linux est nécessaire.

Programme:

- Introduction** : Objectifs d'un système d'exploitation, gestion de ressources.
Partager des ressources entre plusieurs applications, systèmes, ...
Notion de virtualisation , quelle granularité (diques, système d'exploitation, machines physiques, ...)
Historique : VM (Virtual Machine) , VMWare, UML, Xen , ..
- Les différentes techniques possibles :** : conteneurs d'application, noyaux secondaires, machines virtuelles, hyperviseur, virtualisation matérielle, ...
- User Mode Linux** : Exécution de plusieurs instances Linux sur la même machine.
Création d'une distribution virtuelle.
TP: exécution d'une Debian, d'une Suse et d'une Mandrake en parallèle.
Mise en évidence de l'étanchéité. Analyse des performances.
Mise en place de SKAS.
- Xen** : Présentation de l'architecture de virtualisation Xen.
Compilation d'un noyau Xen. Création d'un domaine. Arrêt d'un domaine. Console d'administration.
- VmWare** : Installation du produit. Création d'une machine virtuelle.
Installation du système d'exploitation. Lancement.
- VirtualBox** : Principe et caractéristiques du produit. Les différentes Editions.
Configuration des machines virtuelles en XML
- VServer** : Approche VServer : partitionnement des ressources. Installation et configuration.

Gestion de Parc avec OCS et GLPI

Durée: 3 jours
1430 € HT

du 11 au 13 Janvier
du 1er au 3 Mars
du 3 au 5 Mai

du 7 au 9 Juillet
du 15 au 17 Septembre
du 8 au 10 Novembre

Public:

Administrateur devant gérer un parc de systèmes Linux dans un environnement de production.

Objectifs:

Savoir installer et configurer les outils OCS et GLPI.

Connaissances préalables nécessaires:

Une bonne connaissance des systèmes Unix/Linux est nécessaire.

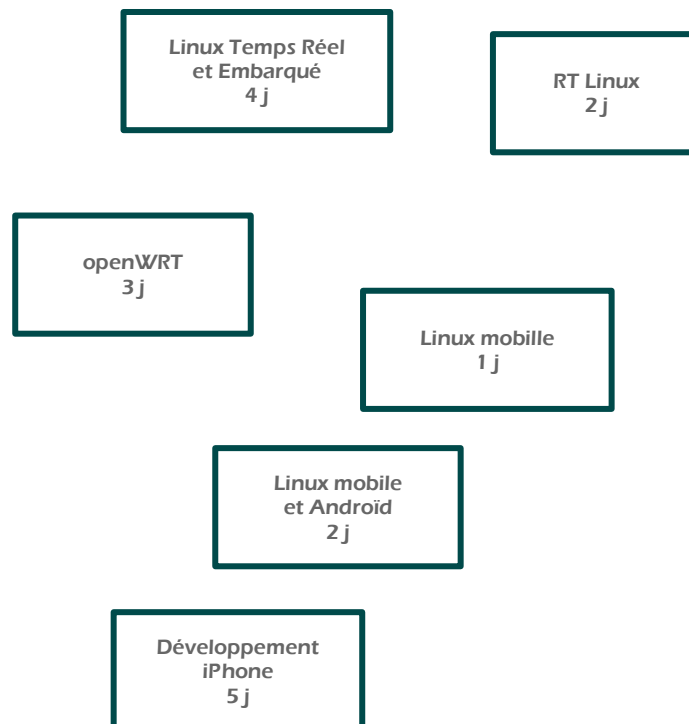
Programme:

- Introduction** : Le besoin : inventaire et suivi des configurations matérielles et logicielles
Présentation OpenComputer and Software Inventory Next Generation
Fonctionnalités, informations collectées
- Architecture OCSNG** : architecture client/serveur
Les composants du serveur OCSNG : base de données, serveur de communication, agents, serveur de déploiement, console d'administration
Intégration avec GLPI
- Installation** : systèmes supportés, installation depuis les packages
création des utilisateurs
Les agents
procédure d'installation et configuration du lancement au démarrage du système
- Configuration** : configuration par l'interface web: fonction IPDISCOVER, détection des doublons (adresses Mac)
notion de TAG : quelques exemples d'utilisation
Les différents paramètres de chaque machine gérée.
- Consultations** : Requêtes prédéfinies. Mise à jour des TAGS, recherches multi-critères, recherche par analyse du TAG
- Télédéploiement de paquets** : Principe et architecture. Notion de priorité et action à exécuter
- Fonctionnalités complémentaires de GLPI** : Gestion du helpdesk, statistiques, réservation de ressources, génération de rapports

Gestion de Parc avec OCS et GLPI

UX124

Filières Systèmes embarqués et mobilité



Linux temps réel et embarqué

Durée: 4 jours
1860 € HT

du 18 au 21 Janvier
du 8 au 11 Mars
du 17 au 20 Mai

du 6 au 9 Juillet
du 4 au 7 Octobre
du 15 au 18 Novembre

Public:

Tout développeur ou ingénieur système souhaitant mettre en œuvre des systèmes Linux avec des contraintes industrielles.

Objectifs:

Connaître les fondements des systèmes temps réels linux.

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

- Introduction** : Définition d'un système temps réel. Les services offerts. Solutions Temps Réel sur Linux. Présentation de RTLinux et de RTAI. Architectures et caractéristiques. Critères de choix: coût.
- Définitions** : Interruptions. Horloge, Entrées/Sorties, automate, séquenceur. Contexte d'exécution. Interruptions d'horloge. Alarmes. Timer cycliques et non cycliques. Automate à états. TP: mise en place d'un mini-moniteur réparti.
- Ordonnancement** : Modification de l'ordonnanceur. Concurrence et coopération. Stratégies d'ordonnancement. Impact sur la gestion de contexte. Mesure de la charge. TP: Mise en évidence du fonctionnement de l'ordonnanceur de RT Linux
- Entrées/Sorties** : Mappées, accès direct. Pré traitement d'E/S. Alarmes sur seuil. Accès direct aux entrées/sorties de bas niveau. E/S série Temps réel. Structure des pilotes de périphériques. TP: mise en place d'une communication inter-processus sur lien série
- Embarqué** : Définitions. Contraintes. L'offre du marché. Les distributions Linux embarquées DD-WRT, openWRT.

Linux temps réel et embarqué

- openWRT** : Présentation. Installation d'une chaîne de compilation croisée. Flashage du micro logiciel. Système de récupération. Introduction à mtd, uci et nvram. Gestion des partitions jffs2.
- Ressources** : Gestion du réseau et des interfaces. Gestion des partitions.
- Applications** : Ajout d'un paquet à partir des sources. Portage d'un paquet "lourd" sur openWrt.
TP: écriture d'un mini-serveur HTTP embarqué. Portage de haproxy et netsnmp.
Compilation d'une MIB spécifique au matériel et intégration dans la distribution embarquée.
- Linux temps réel embarqué** : Portage de RTAI sur openWRT. Problèmes et solutions. Mise en œuvre de klinux, temps réel sur PC104.

RT Linux

Durée: 3 jours

Prix et dates: nous consulter

Public:

Toute personne intéressée par les systèmes temps réel sur Linux

Objectifs:

Connaître les principes de fonctionnement et savoir installer un système RT Linux.

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

- Introduction** : Définition d'un système temps réel. Les services offerts. Solutions Temps Réel sur Linux. Architectures et caractéristiques. Critères de choix: coût.
- Définitions** : Interruptions. Horloge, Entrées/Sorties, automate, séquenceur. Contexte d'exécution. Interruptions d'horloge. Alarmes. Timer cycliques et non cycliques. Automate à états. TP: mise en place d'un mini-moniteur réparti.
- Ordonnancement** : Notion de processus, de ressource, de thread. Analyse et découpage des tâches dans RT Linux. Modification de l'ordonnanceur. Concurrence et coopération. Stratégies d'ordonnancement. Impact sur la gestion de contexte. Mesure de la charge. TP: Mise en évidence du fonctionnement de l'ordonnanceur de RT Linux
- Communication inter-processus** : Communication synchrone entre tâches, verrous. Section critique. Sémaphores. Communications entre le noyau RT et le noyau Linux. Utilisation de mémoire partagée ou de canaux fifos.
- Entrées/Sorties** : Mappées, accès direct. Pré traitement d'E/S. Alarmes sur seuil. Accès direct aux entrées/sorties de bas niveau. E/S série Temps réel. Structure des pilotes de périphériques. TP: mise en place d'une communication inter-processus sur lien série

RT Linux

- Interruptions** : Installation des procédures d'interruption RT Linux (ISR).
Emulation des interruptions Linux.
Partage des interruptions
- Horloges** : Horloges matérielles et logicielles. Alarme POSIX.
Mesure de la résolution des horloges RT Linux

openWrt

Durée: 3 jours
1400 € HT

du 22 au 24 Février
du 26 au 28 Avril
du 28 au 30 Juin

du 20 au 22 Septembre
du 18 au 20 Octobre
du 6 au 8 Décembre

Public:

Intégrateurs de systèmes openWRT.

Objectifs:

Connaître les principes de fonctionnement et savoir installer un système openWrt Linux. Les travaux pratiques seront réalisés sur des plateformes WRT54 et NSLU2.

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

- Embarqué** : Définitions. Contraintes. L'offre du marché. Les distributions Linux embarquées DD-WRT, openWRT. Architectures et caractéristiques. Critères de choix: coût.
- openWRT** : Présentation. Installation d'une chaîne de compilation croisée. Flashage du micro logiciel. Système de récupération. Introduction à mtd, uci et nvram. Busybox. Présentation, ajouts de fonctionnalités, limitations.
- Configuration** : Fichiers de configuration. Paramètres système. Gestion du temps: ntp. Services réseau: dropbear, dnsmasq. Gestion des services dhcp et sshd. Gestion des clés ssh. Modification de la configuration avec uci. Mode maintenance. Récupération après erreur de configuration.
- Réseau** : Gestion du réseau et des interfaces. Configuration réseau avec une adresse par port. Présentation du coupe-feux. Fichiers de configuration. Gestion du Wifi.
- Partitions** : Organisation du système de fichiers: linux,rootfs,nvram,rootfs_data. Gestion des partitions jffs2. Autres types de systèmes de fichiers. Le CFE: accès, gestion, configuration. L'utilitaire nvserial.

openWrt

Applications

- : Mise en place d'un dépôt de paquets.
TP: Portage de netsnmp.
Compilation d'une MIB spécifique au matériel et intégration dans la distribution embarquée.
Mise en œuvre dans un environnement multi-plateformes.
Centralisation des journaux syslog. Supervision par Nagios.

Développement

- : Création d'un paquet applicatif.
Ajout d'un paquet à partir des sources. Portage d'un paquet "lourd" sur openWrt.
TP: écriture d'un mini-serveur HTTP embarqué.

Déploiement

- : TP: Portage de haproxy. Mise en œuvre sur de la répartition de charge HTTP. Frontal à un cluster JBoss. Visualisation des performances.

Linux mobile

Durée: 1 jour
450 € HT

le 1er Février
le 3 Mai

le 29 Septembre
le 29 Novembre

Public:

Toute personne intéressée par les systèmes Linux sur appareils mobiles

Objectifs:

Comprendre les différentes architectures techniques possibles, et connaître les solutions disponibles.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

- Introduction** : Définitions. Contraintes. L'offre du marché. Les principales distributions Linux mobiles: Android, Ubuntu Mobile, Ubuntu Remix, MobLin, Easy Peasy, Linpus.
- Architectures** : Architectures et caractéristiques. Critères de choix: coût. Plateformes supportées. Comparaison des approches. La couche noyau Linux.
- Utilisation** : Applications types. Intégration Web2.0. UME Launcher et le menu NBR sur Ubuntu.
- Applications** : Installation. Mises à jour. Applications bureautiques. Disponibilités. Coûts. Clutter, AudioManager, Flash.
- Gestion des communications** : Configuration de la couche réseau. intégration de Bluez, Carrick, Connection Manager, SyncEvolution
- Développement** : Les différentes plateformes. Emulateurs de test. Présentation des SDK, installation. Langages utilisés. JVM utilisées. TP déploiement d'une application de base sur chacune des plateformes. Comparaisons.

Linux mobile avec Android

Durée: 2 jours
930 € HT

les 2 et 3 Février les 30 Septembre et 1er Octobre
les 4 et 5 Mai les 30 Novembre et 1er Décembre

Public:

Développeurs et intégrateurs d'applications sous Android.

Objectifs:

Connaître les principes de fonctionnement et savoir développer des applications sous Android

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

- | | |
|-------------------------|---|
| Introduction | : Présentation du système d'exploitation android v1.x. Architecture. La couche noyau Linux. Bibliothèques de base. Applicatifs java. |
| Applicatif | : Les principales applications existantes. Installation/désinstallation d'une application. Accès distant à la mémoire flash. Arborescence des fichiers. |
| Développement | : Présentation du SDK, installation. Utilitaires: émulateurs, simulateur de carte. Développement d'une application de base. Transfert sur un matériel de type HTC-Dream. Présentation des appels en call-back. Structure générale des applications. Les quatre modèles d'applications: Activity, Services, Broadcast receivers, Content receivers. Cycle de vie des composants. |
| Interface utilisateur | : Définitions. Présentation des layout. Récupération du contexte applicatif. Gestion des menus. Boîtes de dialogue. Thèmes. Notifications (Toast, Status Bar, Dialog). Ecriture d'une application calculatrice. |
| Graphique et multimédia | : Développement 2D et 3D. Les APIs. Utilisation du MediaPlayer. Visualisation de vidéo. Enregistrement de sons. MediaRecorder. |
| API | : Accès réseau, accès au système de fichiers. Mise en œuvre des APIs réseau. Capteurs internes. Ecriture d'un code exploitant les capteurs de position. |

Linux mobile avec Android

- Interactions d'applications** : Principe de réutilisation d'extraits applicatifs. Ecriture d'une application exploitant GoogleMaps.
Déclaration des possibilités applicatives: les intents filters.
- Système** : Recompilation locale d'Android à partir des sources.
Présentation de l'interaction Linux/Java.
La JVM embarquée: Dalvik VM.

Developpement iPhone

Durée: 5 jours
2150 € HT

du 18 au 22 Janvier
du 22 au 26 Février
du 22 au 26 Mars
du 26 au 30 Avril
du 31 Mai au 4 Juin
du 28 Juin au 2 Juillet

du 6 au 10 Septembre
du 11 au 15 Octobre
du 22 au 26 Novembre
du 13 au 17 Décembre

Public:

Développeurs sur mobiles iPhone

Objectifs:

Connaître les possibilités du développement sur iPhone, maîtriser le langage, les APIs disponibles. Savoir sécuriser les applications.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation comme le langage C, par exemple.

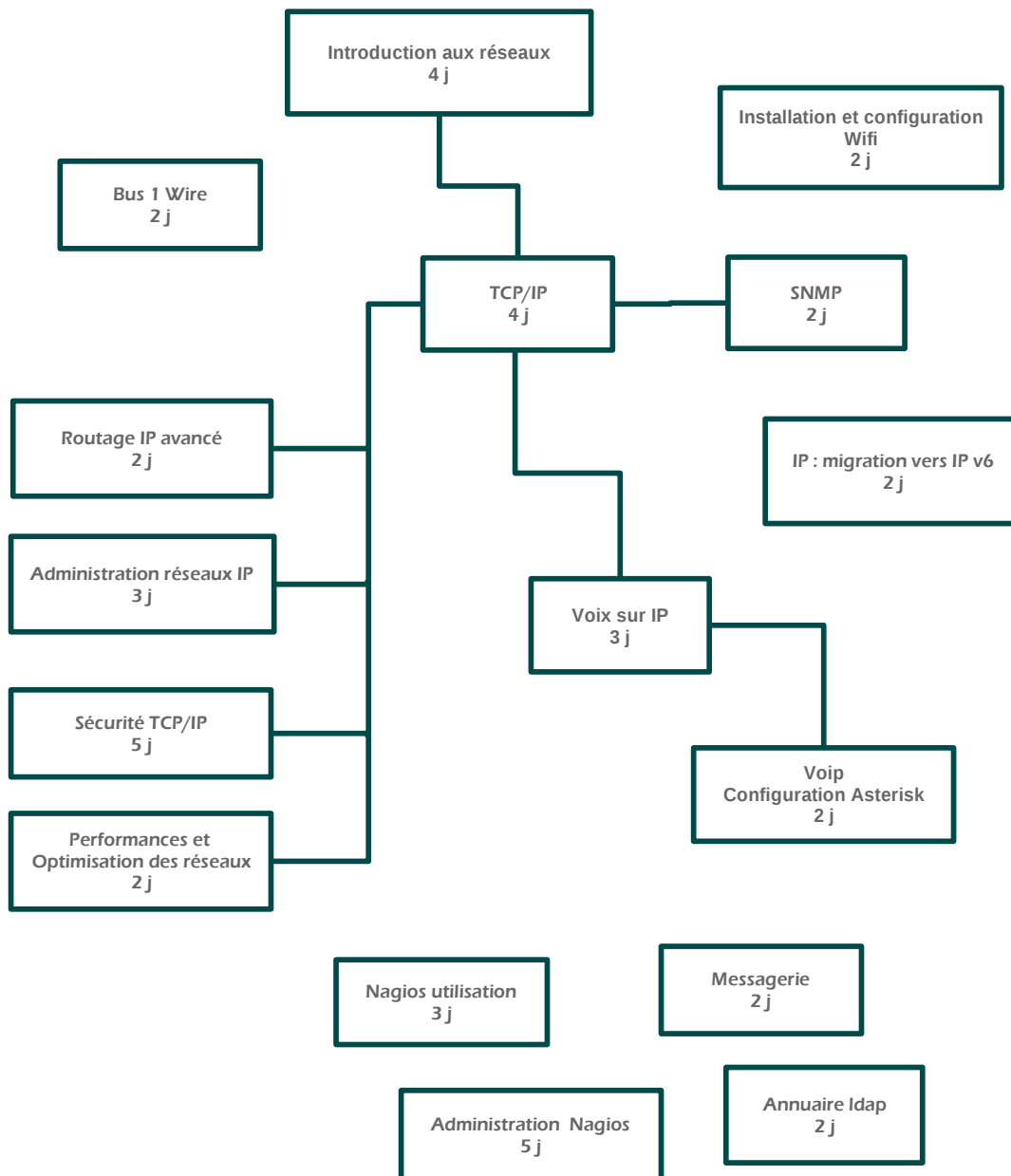
Programme:

- Introduction** : Présentation technique du mobile iPhone :
caractéristiques matérielles et logicielles
Les développements applicatifs :
environnement Cocoa, langage Objective-C,
le modèle MVC,
architectures applicatives,
Le principe AppStore.
- Outils de développement** : Présentation des différents outils
Le kit SDK, simulateur, outils associés,
debugging
- Objective-C** : Les types
Classes, objets, ..
structure d'un programme
- Gestion multimédia** : Graphiques et sons
Evènements
Gestion des capteurs,
Les fonctions disponibles,
Travaux pratiques sur l'accéléromètre, la géolocalisation,
Les alertes
- Programmation système et réseau** : Système de fichiers
Accès réseaux
Gestion de la mémoire

Developpement iPhone

La sécurité : Gestion des profils et relations développeurs et l'App Store

Filières Réseaux et TCP/IP



Introduction aux réseaux

Durée: 4 jours
1810 € HT

du 11 au 14 Janvier
du 8 au 11 Mars
du 17 au 20 Mai

du 5 au 8 Juillet
du 20 au 23 Septembre
du 2 au 5 Novembre

Public:

Toute personne souhaitant acquérir des connaissances générales sur les réseaux, et plus particulièrement sur la mise en œuvre d'un réseau et les outils nécessaires à son exploitation.

Objectifs:

Comprendre les composants fonctionnels d'un réseau informatique. Analyser les possibilités d'interconnexion entre les différents réseaux. Connaître l'état de l'art de la conception, de la gestion et du suivi de réseaux hétérogènes.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours

Programme:

Introduction	: Le besoin de communication Quelques définitions. Les couches ISO. Normalisation et standards (ISO, IEEE, IETF, ATM Forum, ...)
Architectures de base	: Topologies filaires, topologies sans fils Réseau maillé, Doublement de lignes, sécurisation
Supports physiques	: Evolutions technologiques et mutation des réseaux. Acteurs du marché: opérateurs, fournisseurs, intégrateurs, distributeurs. Câblage : topologies et architectures. Usage des locaux techniques. Brassage. Radio : le besoin, les limites, l'état du marché.
Transmissions	: Pourquoi et comment transmettre les informations ? Des transmissions série , parallèle ou hertzienne aux protocoles. Concepts de base et terminologie. Composants des réseaux (produits CISCO, 3COM, ...).
Technologies	: Présentation rapide Ethernet, Giga Ethernet, Token-Ring, FDDI, Frame Relay, RNIS, ATM Les Ethernet: du 10M au 10G. Les normes 802.3ab et 802.3ae
Réseaux sans fils Wlan	: HiperLAN IEEE 802.11

Introduction aux réseaux

- Normes Wifi** : Présentation
Points forts, points faibles
Architecture des réseaux Wifi : 802.11, exemple d'ESS, le monde ad hoc, OLSR
le matériel, interopérabilité
- Utilisation du Wifi** : Points d'accès
Modes de fonctionnement, mode répéteur, Mode pont
Alignement d'antennes, Supervision de réseaux
- TCP/IP** : Définitions, adressage
Exemple d'application
Le protocole IP, la trame IP,
TCP, UDP
- Outils réseau** : Outils de trace, tcpdump, outils de diagnostic actifs/passifs,
analyseurs de flux, ...
- Interconnexion de réseau et routage** : Technologies, commutation
Routage IP
Fragmentation , VLAN
Outils de gestion du routage
Plan d'adressage
OoS
- IPV6** : Besoin, fonctionnalité
La trame IPV6, adressage
- Sécurisation** : VPN et tunnels: Objectif, fonctionnement
DMZ et Pare-feux: Définition, serveur Proxy, fonctionnement
pare-feux et tunneling
Filtrage: les iptables, politique par défaut, état des connexions,
traduction d'adresses,
traduction de ports, connexion à internet
- Voix sur IP** : Commutation de paquets
Avantages de la voix sur IP
Les protocoles : H323, SIP
Introduction RTP : définition et applications, RTP et Nat
Utilisation du registrar SIP avec Asterisk
Création des comptes téléphones, du dialplan, vérification et tests
Enregistrements SRV : serveurs DNS et Asterisk
Transport de données
Bande passante et qualité de service (QoS)
- Evolutions** : L'adressage IP, la sécurité, les réseaux de stockage.

Bus One-wire

Durée: 2 jours
900 € HT

les 25 et 26 Février
les 29 et 30 Avril
les 1er et 2 Juillet

les 23 et 24 Septembre
les 21 et 22 Octobre
les 9 et 10 Décembre

Public:

Toute personne intéressée par l'intégration d'un système 1wire pour la commande à distance.

Objectifs:

Connaître les principes de fonctionnement du bus 1wire et comprendre sa mise en œuvre.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

- Introduction** : Présentation de quelques technologies réseaux: Ethernet généraliste, CAN, FlexRay et Most en industriel, X10, I2C et 1wire en domotique.
- Bus 1wire** : Spécifications. Mode de fonctionnement. Mode vampire, mode trois fils. Présentation des produits: identifiant, capteur de température, interrupteurs, jauges, mémoires, horloges, convertisseurs analogiques, potentiomètres numériques.
- Mise en œuvre** : Ecriture d'un pilote 1wire. Connexion et configuration du pilote matériel DS2480. Solution 9097/9490.
Connexion des capteurs. Mise en œuvre dans une application de supervision de capteurs: tout ou rien, jauge, temperature.
Mesure des vitesses de transfert.
- Embarqué** : Portage du système 1wire sur une plateforme embarquée openWrt.
Extension du 1wire sur des réseaux étendus avec le coupleur 2409.

TCP/IP : protocoles et mise en œuvre

Durée: 4 jours
1540 € HT

du 1er au 4 Février
du 6 au 9 Avril
du 14 au 17 Juin

du 13 au 16 Septembre
du 18 au 21 Octobre
du 6 au 9 Décembre

Public:

Toute personne souhaitant mettre en œuvre TCP/IP et les outils nécessaires à son exploitation.

Objectifs:

Maîtrisez les fonctionnalités du protocole TCP/IP, sa position par rapport aux autres protocoles. Savoir configurer un routeur et les différents composants d'un réseau local. Savoir mettre en œuvre les aspects fonctionnels et les services applicatifs.

Connaissances préalables nécessaires:

Connaissances de bases sur les réseaux et les systèmes d'exploitation.

Programme:

- Introduction** : Définitions : IP, TCP. Historique. IP dans le modèle ISO.
- IP** : Le protocole IP: trame, adressage, principes de routage.
Configuration des adresses et des masques réseaux.
Accès à la couche réseau sur différents systèmes d'exploitation.
Configuration de l'interface réseau.
- Routage** : Interconnexion de réseaux, répéteurs, les ponts. La commutation.
Routeurs et passerelles.
Définition d'une topologie. Principe de routage, algorithmes.
Configuration des routeurs et des postes clients.
Visualisation des chemins utilisés via traceroute.
Routage dynamique: RIP, OSPF.
- TCP/UDP** : Les protocoles UDP/TCP: mode non connecté/connecté.
Connexion virtuelle. Les ports TCP bien-connus (well known ports)
- Applications** : Les services du niveau application: telnet, ftp, ssh, scp, traceroute, ping (connexion, transfert de fichiers, contrôle), modèle client-serveur.
Serveurs de noms : DNS (Domain Name System). Principe, traitement des requêtes. Mise en œuvre : configuration client, serveur, accès inversé.
SNMP (Simple Network Management Protocol) : fonctionnalités, appports SNMP V2.

TCP/IP : protocoles et mise en œuvre

IPv6

: Adressage actuel, attribution des adresses.
Le travail de l'IETF (Bradner/Mankin). Plan d'adressage sur 128bits.
Agrégateurs : découpage TLA/NLA/SLA/IID. Intégration des Regional Registries
Fonctionnement : Surcharge d'entêtes. Structures des trames.
Les nouveaux mécanismes: fragmentation: MTU universelle, DHCPv6, dynamic DNS, renumérotation simplifiée d'un plan d'adressage

Sécurité

: Ipsec (IP Security Protocol)
TP de mise en œuvre

Voix sur IP

Durée: 3 jours
1340 € HT

du 25 au 27 Janvier
du 22 au 24 Mars
du 31 Mai au 2 Juin

du 6 au 8 Septembre
du 25 au 27 Octobre
du 13 au 15 Décembre

Public:

Les administrateurs réseau, et toute personne souhaitant mettre en place un système de communication audio sur IP.

Objectifs:

Comprendre le mécanisme de fonctionnement de la voix sur IP, connaître les outils et produits permettant une utilisation efficace de la voix sur IP dans l'entreprise.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

- Présentation** : Définitions: RTC, commutation de paquets.
Avantages et inconvénients de la voix sur IP
- Architectures** : H323. SIP. Comparaison. Type d'appels: téléphone vers téléphone, PC vers PC, PC vers téléphone.
Echanges RTCP, RTP. Adressage: affectation d'un numéro à une adresse IP. Allocations des numéros.
- Qualité** : Qualité de la voix. Compression. Détection des défauts.
Codecs: G711, G723, G726, G728, G729.
Bande passante utilisée. Calibrage. Utilisation de RSVP (Resource Reservation Protocol), MPLS (Multiprotocol Label Switching) ou TOS. Mesure du trafic. Utilisation des iptables.
Gestion de la qualité de service et répartition voix/données.
TP: Mise en œuvre d'une liaison PC/téléphone vers PC/téléphone en voix sur IP
- Etat de l'art** : Les produits du marché : constructeurs et éditeurs

Voip - Configuration Asterisk

Durée: 2 jours
910 € HT

les 28 et 29 Janvier
les 25 et 26 Mars
les 3 et 4 Juin

les 9 et 10 Septembre
les 28 et 29 Octobre
les 16 et 17 Décembre

Public:

Les administrateurs réseau, et toute personne souhaitant utiliser le logiciel Asterisk en téléphonie IP.

Objectifs:

Comprendre le fonctionnement d'Asterisk, et savoir paramétrer le logiciel afin d'utiliser l'ensemble des fonctionnalités proposées.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les principes de la Voix sur IP

Programme:

- Asterisk introduction** : Présentation, fonctionnalités.
Installation et configuration de base.
- Utilisation du registrar SIP** : Notion de registrar
Implémentation :
création des comptes téléphones et du "dialplan"
- Enregistrements SRV** : Communications entre domaines
Création de DNS et configuration Asterisk
Mise en œuvre et tests
- Plan de numérotation** : Définitions : contextes, applications
Configuration /etc/asterisk/extensions.conf :
syntaxe générale, contextes, extensions, priorités.
Configuration des Applications :
Dial(), Record(), CDR (Call Details Record)
Syntaxe détaillée : les opérateurs, expressions, motifs,
branchements conditionnels, fonctions et macros.

Installation et Configuration WIFI

Durée: 2 jours
890 € HT

les 18 et 19 Janvier

les 12 et 13 Juillet

les 18 et 19 Mars les 30 Septembre et 1er Octobre

les 25 et 26 Mai

les 29 et 30 Novembre

Public:

Les ingénieurs réseaux souhaitant aborder les technologies sans fil, les chefs de projet et toute personne souhaitant comprendre les mécanismes du wifi.

Objectifs:

Connaître les principes de fonctionnement du wifi, les conditions d'utilisation, et savoir élaborer une architecture de réseau à base de wifi.

Connaissances préalables nécessaires:

De bonnes connaissances réseau sont souhaitées.

Programme:

- Les réseaux sans fil : le besoin
avantages, inconvénients par rapport aux réseaux fixes
les différentes catégories de réseaux sans fil
- technologie WIFI : la norme 802.11, lien avec Ethernet,
l'architecture, le mode ad hoc,
les performances,
Les fréquences utilisées.
- Mise en pratique : les équipements, les outils disponibles sur le marché
Configuration d'un réseau WIFI :
les points d'accès, mise en œuvre en mode répéteur, mode
pont
l'alignement d'antennes
- Sécurité : les problèmes
le WEP, les insuffisances
chiffrement RC4, Mise en place de WPA et de EAP-TLS
Comment fiabiliser : Radius, IPsec, VPN
- Utilisation du Wifi avec Linux : Les pilotes
Wireless Tools. Utilisation de hostapd pour créer un point
d'accès.
- Le roaming : Présentation de la norme 802.11f, la gestion des utilisateurs
itinérants

Administration réseaux IP

Durée: 3 jours
1370 € HT

du 15 au 17 Février
du 19 au 21 Avril
du 21 au 23 Juin

du 25 au 27 Août
du 11 au 13 Octobre
du 29 Novembre au 1er
Décembre

Public:

Les administrateurs réseaux et tout exploitant de réseaux TCP/IP.

Objectifs:

Savoir configurer, tester, surveiller un réseau TCP/IP. Gérer les montées de version.

Connaissances préalables nécessaires:

Notions sur le câblage, transmissions et les bases de TCP/IP.

Programme:

Adressage IP	: Utilisation de l'outil "ping" pour vérifier l'existence d'une machine. Création d'un outil de recherche automatique des machines actives sur le réseau.
RIP	: Routage IP dynamique. Mise en œuvre de RIP sur différents routeurs, visualisation par traceroute.
DNS	: Définition.Fonctionnement. Configuration. Mise en place de plusieurs serveurs DNS primaire et secondaires. Interconnexion.
DHCP	: Mise en place d'un serveur DHCP: partage d'une plage d'adresses. Activation de la RFC 1542.
NTP	: Définition.Fonctionnement. Déclaration d'un point de synchronisation. Configuration d'un serveur d'horloge. Configuration de clients. Architecture. Contrôles d'accès.
bootp, tftp	: Configuration à distance Surveillance du réseau
Analyseurs de trames	: Mise en place d'un outil de surveillance des paquets transitant sur un réseau (tcpdump, sniffit).
MRTG	: Surveillance de flux, de charge.
SNMP	: Rappels du principe de fonctionnement. Principe des MIB. Surveillance des différentes ressources d'un poste. Exécution de processus distants.
Extensions SNMP	: ASN-1, modification de MIB

Administration réseaux IP

- Administration Web : webmanagement avec Webmin. Installation/mise en route.
Présentation de l'interface. Configuration de clusters.
- OoS : régulation du flux en fonction du protocole
TP: mise en œuvre de CBO puis de HTB.

Routage IP avancé

Durée: 2 jours

Prix et dates: nous consulter

Public:

Les administrateurs réseaux et tout exploitant de réseaux TCP/IP.

Objectifs:

Savoir configurer un routeur et les différents composants d'un réseau local.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

- | | |
|-------------------|---|
| Introduction | : Routage, partage, commutation
Commutation de niveau 3
Pontage sous Linux.LARTC, ebttables, brctl |
| Rappels sur IP | : Plan d'adressage IP
notion de classes
les masques de sous-réseaux |
| Routage | : Définition d'une topologie.
Principe de routage.
Configuration des routeurs et des postes clients.
Visualisation des chemins utilisés via traceroute. |
| Routage dynamique | : RIP, OSPF
TP: mise en évidence des transfert RIP
TP: mise en œuvre d'OSPF.
Présentation du routage des systèmes autonomes:
IGRP
BGP, BGP-4
Routage OSPF sous Linux et sous IOS Cisco. |

IP: Migration vers IPv6

Durée: 2 jours
910 € HT

les 18 et 19 Février
les 22 et 23 Avril
les 3 et 4 Juin

les 16 et 17 Septembre
les 8 et 9 Novembre
les 9 et 10 Décembre

Public:

Toute personne souhaitant migrer vers l'adressage IPv6

Objectifs:

Connaître les caractéristiques d'IPv6, et savoir élaborer les méthodes de migration.

Connaissances préalables nécessaires:

Connaissances de bases sur les réseaux et les systèmes d'exploitation.

Programme:

- Rappels sur la version 4** : Le protocole IP: trame, adressage, principes de routage.
Configuration des adresses et des masques réseaux.
- IPv6** : Adressage actuel, attribution des adresses.
Le travail de l'IETF (BradnerMankin). Plan d'adressage sur 128bits.
Agrégateurs : découpage TLA/NLA/SLA/IID. Intégration des Regional Registries
Fonctionnement : Surcharge d'entêtes. Structures des trames.
Les nouveaux mécanismes: fragmentation: MTU universelle, DHCPv6, dynamic DNS, renumérotation simplifiée d'un plan d'adressage
- Migration v4/v6** : Les différentes approches: double pile, encapsulation statique, encapsulation dynamique
Présentation de l'encapsulation v6 dans v4 et l'extension Teredo pour les réseaux à translation d'adresses
Comparaison des différentes approches: isanat/Teredo
Choix d'un tunnel broker
- Services connexes** : Cas des DNS
La sécurité: IPsec, les pare feux et les filtres.
Supports natifs sur les applications réseaux: messagerie, connexions (ssh), web (apache, mozilla), ldap, dhcp, proxies, ...

Performances et optimisation des réseaux

Durée: 2 jours
910 € HT

Public:

Consultants réseaux, responsables techniques, responsables réseaux, architectes, intégrateurs, chefs de projet possédant des connaissances générales en réseaux.

Objectifs:

Identifier les besoins et les critères de performances des flux transmis sur les réseaux IP. Evaluer l'impact des applications sur les performances d'un réseau. Utiliser des techniques d'estimation et de mesures pour valider les performances. Concevoir des réseaux évolutifs et optimisés.

Connaissances préalables nécessaires:

Notions sur le câblage, transmissions et les bases de IP.

Programme:

- Mesure de la charge ponctuelle** : Présentation des outils de base de surveillance des paquets: tcpdump, sniffit, wireshark.
iptraf, snmp, ntop
TP: mise en œuvre d'un script de détection de surcharge sur un brin.
- Mesure de la charge entre deux points** : Calcul de la bande passante disponible. Mesure de base avec ping. Mesure du taux de perte de paquets, du délai de transit et valeur de la gigue avec iperf.
Présentation des produits Argus, et de l'architecture de collecte nfdump/nfsen.
- Outils de graphes** : MRTG, Cricket, Cacti. Comparaison.
- Solutions** : Problèmes liés à l'architecturer: pose de liens inter-routeurs et de dérivation.
Amélioration du routage: RIP, OSPF, découpage en systèmes autonomes. Gestion des VLANS. Duplication de port sur un commutateur.
Gestion de la qualité de service.
régulation du flux en fonction du protocole
TP: mise en œuvre des tc avec CBO puis HTB sur une interface.

Sécurité TCP/IP

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant maîtriser la sécurité sur TCP/IP, et plus particulièrement les administrateurs et les architectes réseaux.

Objectifs:

Savoir mettre en œuvre les mécanismes de sécurité, analyser les traces, configurer les systèmes de protection, concevoir une architecture de réseau fiable.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

- Introduction** : Analyse des risques.
Exemples avec l'étude des flux: tcpdump, sniffit,
TP: visualisation des mots de passe transitant par le réseau.
- Contrôle des accès système** : Protection de services réseaux: telnet, tftp, snmp, ftp, ...
Le 'tcp wrapper'
Verrouillage des accès physiques à distance.
Connexions sécurisée: SSH (configuration, connexion automatique), ssl, sftp, scp, tunneling X11
Contrôle de la messagerie: clamAV, p3scan, pop3s, imap4s
Gestion des accès: Radius
- Architecture de sécurité** : : Coupe feux: DMZ, Proxy. Pose de filtres sur un routeur.
TP: mise en place d'un proxy ftp
iptables, PAT, stunnel, VPN (openvpn, freeSWAN), VLAN
Les apports d'IPsec.
TP: mise en place d'une architecture openVPN
- Sécurisation échanges** : Chiffrage des données, mécanisme des certificats
- Surveillance** : Le protocole SNMP; la surveillance d'applications
TP: écriture d'un analyseur de topologie
Contrôle des flux. Analyseurs de trames: ethereal, tcpdump,
analyse de failles: nessus

SNMP

Durée: 2 jours
910 € HT

les 14 et 15 Janvier
les 18 et 19 Mars
les 10 et 11 Mai

les 12 et 13 Juillet
les 27 et 28 Septembre
les 25 et 26 Novembre

Public:

Les administrateurs réseau, et toute personne souhaitant mettre en place un système de supervision par SNMP.

Objectifs:

Comprendre le mécanisme de fonctionnement de SNMP, connaître les outils et produits permettant une utilisation efficace de SNMP dans la supervision du réseau.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Le protocole	: Simple Network Management Protocol RFC 1213
Fonctionnement	: Le principe des MIB. La hiérarchie SNMP. Surveillance des différentes ressources d'un poste. Exécution de processus distants
Mise en pratique	: Réalisation d'un analyseur de topologie.
Sécurité	: Authentification Protection du contenu
L'usage de SNMP sur le marché	: Les produits d'analyse, les MIBs développées par les constructeurs.
Développement	: Développement d'une MIB. Description de la structure en ASN-1. Conversion en C et compilation dans l'agent SNMP. TP: ajout d'OID surveillant la température du processeur. TP: ajout d'OID surveillant le nombre de threads d'un serveur JEE.

Annuaire LDAP

Durée: 2 jours
880 € HT

les 18 et 19 Janvier

les 18 et 19 Mars

les 27 et 28 Mai

les 22 et 23 Juillet

les 30 Septembre et 1er Octobre

les 29 et 30 Novembre

Public:

Administrateurs réseaux, intégrateurs d'application souhaitant configurer un annuaire ldap.

Objectifs:

Comprendre le mécanisme de ldap, et savoir mettre en place un service d'annuaire. Les travaux pratiques ont lieu avec Openldap.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

- | | |
|----------------|---|
| Introduction | : le besoin, historique. Définitions. |
| Procotole LDAP | : Lightweight Directory Access Protocol
Principe de fonctionnement.
Les modèles, la conception d'une arborescence : construction, importation de schema |
| Sécurité | : Authentification, contrôle d'accès, chiffrement des transactions
Annuaire et PKI |
| Architecture | : Distribution, réplication d'annuaires. |
| Mise en œuvre | : Travaux pratiques avec Openldap.
Installation, configuration. |
| Intégration | : Dans le réseau d'entreprise :
Exemples : pam/ldap, samba/ldap. |
| Le marché | : Présentation des principaux annuaires. |

Messagerie

Durée: 2 jours
880 € HT

les 18 et 19 Février
les 22 et 23 Avril
les 24 et 25 Juin

les 2 et 3 Septembre
les 14 et 15 Octobre
les 2 et 3 Décembre

Public:

Administrateurs réseaux.

Objectifs:

Savoir installer, configurer et administrer une messagerie sous Unix.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur les réseaux TCP/IP.

Programme:

- TCP/IP : Mode de fonctionnement: Adressage IP, nommage DNS (serveur de noms)
TP: écriture d'un plan d'adressage et mise en œuvre
- DNS : Fonctionnement, configuration du service.
- Concepts de messagerie : Terminologie : SMTP, POP3, IMAP4, MTA, MDA, MUA...
Le routage de messages.
Anatomie d'un message, les champs d'entête
- Architecture distribuée : Les clients, modes d'accès au courrier
les protocoles : POP, IMAP, principes de fonctionnement. Etude du protocole POP3.
Les extensions SMTP. Gestion des pièces jointes.
TP : mise en place d'un système complet de messagerie, configuration d'un serveur SMTP (Postfix)
- Marché : Les produits du marché
Présentation des serveurs sendmail, Postfix.
- Exploitation : Définitions d'alias. Traitements à l'arrivée (procmail, formail).
Anti-spam, clamAV, p3scan.
Sécurisation, chiffrement, authentification.

Nagios Centreon : utilisation

Durée: 3 jours
1340 € HT

du 1er au 3 Février
du 15 au 17 Mars
du 10 au 12 Mai

du 19 au 21 Juillet
du 27 au 29 Septembre
du 24 au 26 Novembre

Public:

Exploitants et utilisateurs d'un système de supervision Nagios, et de l'interface Centreon.

Objectifs:

Connaître les fonctionnalités de Nagios, maîtriser l'interface d'exploitation Centreon, savoir ajouter de nouveaux test, savoir mettre en place une politique d'alerte, créer de nouveaux types de notification.

Connaissances préalables nécessaires:

Connaissance de l'architecture d'un système d'information, bases tcpip et bases systèmes.

Programme:

- Introduction** : Les objectifs de la supervision, les techniques disponibles.
- Les fonctionnalités** : Supervision, exploitation.
Surveillance des services réseaux,
Surveillance des ressources (charge CPU, espace disque)
Envoi d'alarme vers des contacts déterminés;
Déclenchement de scripts pour corriger les problèmes.
- Utilisation** : Page d'accueil
Les hôtes et services
Cartographie du réseau
Visualisation des tests
Détection des pannes
Recherche d'hôte
Arrêts programmés
L'interface Centreon
- Paramétrage** : Objets à définir :
hôtes, groupes, services, dépendances,
notifications, escalades
- Déploiement** : Sur les hôtes, principes de NRPE, NSCA.
Exemples de scripts de déploiement. (NRPE)
- Les plugins** : Principe de fonctionnement.
Quelques plugins courants

Administration Nagios

Durée: 5 jours
2140 € HT

du 8 au 12 Février
du 12 au 16 Avril
du 14 au 18 Juin

du 13 au 17 Septembre
du 18 au 22 Octobre
du 15 au 19 Novembre

Public:

Les administrateurs systèmes, administrateurs réseau, et toute personne souhaitant mettre en place un système de supervision.

Objectifs:

Connaître les fonctionnalités de Nagios. Savoir installer, configurer et administrer le produit. Savoir développer des nouveaux plugins.

Connaissances préalables nécessaires:

Notions sur le réseau, bases de TCP/IP. Connaissance d'un langage de script.

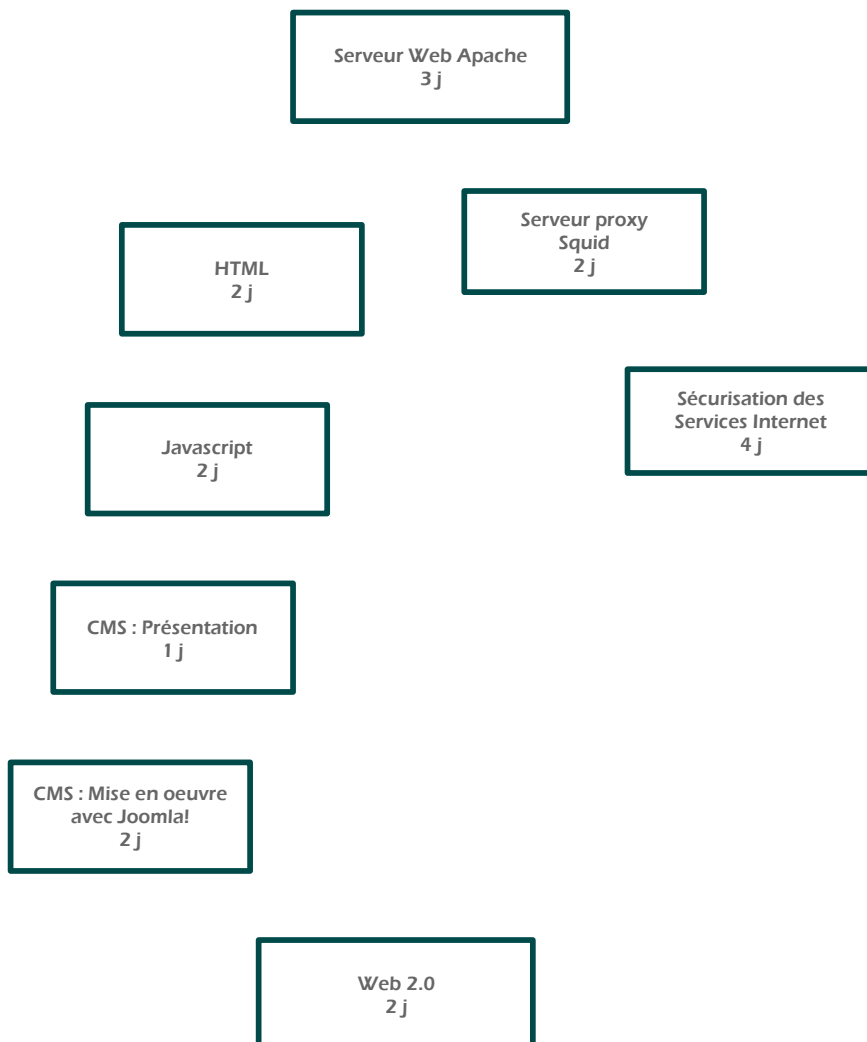
Programme:

- Introduction** : Les objectifs de la supervision, les techniques disponibles.
Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING
Définition des ressources à surveiller.
- Présentation Nagios** : Les fonctionnalités
Supervision, exploitation.
Surveillance des services réseaux,
Surveillance des ressources (charge CPU, espace disque)
- Architecture** : Principe de fonctionnement et positionnement des différents modules.
Les plugins et extensions
- Installation** : Configuration requise. Site de référence.
Installation et mise à jour,
Paramétrage de base, démarrage Nagios
- Configuration** : Etude du fichier de configuration standard nagios.cfg.
Description des serveurs à surveiller, création de groupes de serveurs.
Description des contacts, et création de groupes de contact, escalades
Définition des services et groupes de services.
Les notions de hiérarchie, dépendances : hôtes et services.
Configuration de l'interface web d'administration
- Optimisation de l'ordonnanceur** : Méthode, Délais entre tests, Tests concurrents, fréquence, ...

Administration Nagios

- Les plugins : Principe de fonctionnement.
Personnalisation de Nagios par développement de nouveaux plugins.
- Lien SNMP : Présentation du protocole SNMP
Tests actifs et passifs
: Déploiement
Sur les hôtes, principes de NRPE, NSCA.
Exemples de scripts de déploiement. (NRPE)
- Superviseurs redondants : La haute disponibilité : mode fail-over, configuration d'un superviseur secondaire
- Centralisation NDO : Fonctionnalités et composants
Mise en œuvre de NDO
- Extensions Nagios : Intégration avec Cacti, Centreon

Filières Internet et Web



Serveur WEB : apache

Durée: 3 jours
1230 € HT

du 25 au 27 Janvier
du 15 au 17 Mars
du 10 au 12 Mai

du 19 au 21 Juillet
du 27 au 29 Septembre
du 24 au 26 Novembre

Public:

Toute personne souhaitant configurer, installer et exploiter un serveur web à base d'apache.

Objectifs:

Comprendre le fonctionnement d'apache, et savoir le mettre en œuvre, l'installer, le configurer et l'administrer.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

- Introduction** : Présentation, historique
Installation, configuration de base
Principe, le rôle des modules.
- Configuration du serveur** : Environnement, gestion des processus, requêtes, connexions client : le fichier httpd.conf.
- Configuration d'un site** : Nom interne du serveur, rappel sur les DNS
page d'accueil, types de fichiers,
les alias, définition de chemins relatifs, la balise index
- Sécurité** : Exécution, contrôle d'accès, restriction d'accès par répertoire, par méthode, ...
Authentification, SSL
- Hôtes virtuels** : Principe, configuration, hôtes virtuels basés sur l'adresse IP, sur le nom
- WebDynamique** : Principe des scripts CGI, fonctionnement, mise en place.
- Exploitation Apache** : Administration du site,
personnalisation des pages d'erreurs
les fichiers journaux: analyse directe, analyse par webalizer
Mise en charge et visualisation du comportement.

Serveur Proxy Squid

Durée: 2 jours
850 € HT

les 1er et 2 Juillet
les 21 et 22 Janvier les 30 Septembre et 1er Octobre
les 6 et 7 Avril les 2 et 3 Décembre

Public:

Toute personne souhaitant mettre en place un serveur Squid afin d'optimiser et sécuriser les accès Internet de l'entreprise.

Objectifs:

Comprendre les principes de base d'un serveur de cache Internet. Savoir élaborer les configurations optimisant le fonctionnement du serveur Squid. Mettre en place les contrôles d'accès dans une architecture solide et sécurisée.

Connaissances préalables nécessaires:

Des connaissances minimales sur TCP/IP (adressage, fonctionnement) ainsi que sur le fonctionnement du Web sont nécessaires pour suivre ce cours.

Programme:

- Besoin** : Serveur Proxy, fonctionnement, multi serveurs proxys, hiérarchie de serveurs cache, cache transparent, accélérateur Web ou proxy inversé.
- Squid** : Présentation, sécurité, architecture externe. Exemple d'utilisation, systèmes d'exploitation concernés, logiciels complémentaires.
- Installation** : Installation à partir de paquetages, fichiers de configurations, configuration de base, test du serveur.
- Configuration des postes clients** : configuration manuelle, automatique. Scripts d'auto-configuration, filtrage suivant DNS, par protocole. Clients en mode texte ,robots. Installation dans le navigateur.
- Configuration du serveur** : principe et syntaxe des ACL. Optimisation de l'utilisation du serveur. Restriction d'accès par hôte, par réseau, par plage horaire, par jour, par site. Mise en cache des données. Méthodes d'authentification.
- Administration** : Surveillance, support SNMP. Configuration par WebMin. Fichiers journaux
- SquidGuard** : Présentation, les groupes source, groupes de destination. Réécriture d'URL, règles d'accès. Principe de la base de données, utilisation, considérations de performances

Sécurisation des services internet

Durée: 4 jours
1810 € HT

du 25 au 28 Janvier
du 29 Mars au 1 Avril
du 7 au 10 Juin

du 6 au 9 Septembre
du 2 au 5 Novembre
du 13 au 16 Décembre

Public:

Les administrateurs réseaux, et toute personne souhaitant sécuriser les services internet et/ou intranet.

Objectifs:

Connaître les moyens de sécuriser l'architecture et les services réseaux IP.

Connaissances préalables nécessaires:

Connaissances de bases sur les réseaux et les systèmes d'exploitation.

Programme:

- Introduction**
 - : Définitions: DMZ, proxy, pare-feux, routeurs filtrants
Fonctionnement, mise en œuvre
 - : Description du filtrage avec iptables.
Traduction d'adresses. Traduction de ports.
Sécurisation de l'ouverture d'un port. Mise en œuvre d'une fonction d'activation d'ouverture avec les iptables.
- Proxy**
 - : Proxies: mise en œuvre avec squid. Les ACL sous Squid.
Exemple d'authentification NCSA, LDAP.
- VPN et tunnels**
 - : Objectifs. Fonctionnement. Mise en œuvre. Tunnels ssh, stunnel, openVPN.
- Services réseaux**
 - : Sécurisation DNS: architecture redondante, DNS fermé, Hidden master, Stealth DNS. Mise en œuvre.
Authentification. Les signatures TSIG. ACL.
- Outils**
 - : Traçage des flux réseaux: wireshark, tcpdump.
Détection: snort, nessus.
- Messagerie**
 - : Sécurisation. Notion de relais ouvert. Outils de test du serveur.
Mise en œuvre avec postfix. Cyrus. saslauthd.
- Authentification des accès réseaux**
 - : Authentification par un serveur radius.
Techniques de filtrage à base d'iptables. Mise en œuvre d'un système d'authentification par utilisateur et non par adresse: NuFW.

HTML

Durée: 2 jours
750 €

du 18 au 19 Janvier
du 19 au 20 Avril
du 28 au 29 Juin

du 18 au 19 Octobre
du 29 au 30 Novembre

Public:

Développeurs de pages HTML.

Objectifs:

Etre capable de développer des documents HTML. Explorer l'ensemble des possibilités offertes par HTML.

Connaissances préalables nécessaires:

Connaissances générales sur Internet.

Programme:

- Présentation** : HTML : le langage du Web. Contexte historique. Le concept des balises. Le bon usage des balises. Particularités et pièges du HTML. Le document HTML minimum.
- Mise en forme** : Document: Objets d'un document. En-tête d'un document HTML.
Texte: Titres. Paragraphes. Contrôle de passage à la ligne. Formatage, alignement, taille, couleur, police. Caractères spéciaux. Commentaires. Autres balises de texte.
- Les listes** : Numérotées. A puces. Imbriquées. Les listes de définition.
- Les tableaux** : Cellules. Fusion. Gestion de la taille du tableau. En-tête et légende. Bordures. Groupes de colonnes et de lignes.
- Les hypertextes** : Balise de liens. Liens externes, liens internes. Types d'URL: site, adresse électronique, téléchargement. Target. Attribut title. Couleur des liens. Liens et feuilles de style.
- Images** : Formats des images du Web. Insertion d'une image, espace autour d'une image, alignement, couleur d'arrière-plan, image d'arrière-plan, lien sur une image. Images réactives
- Les éléments multimédias** : Flash, Real, Quicktime, Pdf et d'applets Java
- Frames** : Page de cadres élémentaire. Cadre avec colonnes. Cadre avec lignes. Attributs des cadres. Liens entre les cadres. Cadres en ligne (balise iframe). Cadre ou tableau.

HTML

Formulaires

: Déclaration de formulaire.
Zone de texte mono-ligne, multi-lignes. Menu déroulant.
Boutons radio,checkbox. Envoi, annulation, commande.
Formulaires cachés,formulaires de transfert de fichier, de mot
de passe. Organisation des éléments d'un formulaire.

Web Dynamique avec JavaScript

Durée: 3 jours
1130 €

du 20 au 22 Janvier
du 21 au 23 Avril
du 30 Juin au 2 Juillet

du 30 au 22 Octobre
du 1er au 3 Décembre

Public:

Toute personne désireuse d'acquérir la maîtrise de la programmation en langage JavaScript, son intégration dans les pages HTML, et la méthode AJAX.

Objectifs:

Développer l'interactivité de sites Web grâce à l'utilisation du langage JavaScript et exploiter les fonctionnalités des feuilles de style à travers Javascript.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître le principe de fonctionnement d'Internet et du Web, le langage HTML. La connaissance d'un langage de programmation est utile.

Programme:

- Introduction** : Besoin, historique, cadre d'utilisation.
- Rappels sur les formulaires HTML** : Mode d'appels des scripts depuis les pages HTML.
Comment créer une zone de saisie? Les différents modes de saisie HTML
L'interpréteur JavaScript: entrées/sorties
- Les bases du langage** : La balise "SCRIPT". Syntaxe du langage. Appel de fonctions, récupération des saisies utilisateurs, passage d'arguments, instructions et opérateurs, chaînes de caractères.
Structure d'un programme: contrôle: tests, branchements, boucles
- Organisation** : La hiérarchie des objets: l'objet "document"
Les noms de variables, les constantes, les caractères spéciaux
Programmation des objets de base.
- Développement avancé** : Les ordres standards: les liens, les boutons radios, les zones de saisies (JavaScripts), gestion de la souris
Gestion des événements, des cookies.
Interfaçage avec Java
- Interaction avec les feuilles de style** : CSS multiples, héritages multiples, règles de priorité
Les blocs de texte, les calques, les couleurs, mises en forme de paragraphes, DIV ...
Accès et modification des styles via Javascript.

CMS : Présentation

Durée: 1 jour
450 € HT

le 22 Février
le 26 Avril
le 28 Juin

le 1er Septembre
le 2 Novembre
le 15 Décembre

Public:

Chefs de projets, développeurs web, administrateurs web

Objectifs:

Comprendre le fonctionnement d'un système de gestion de contenu (Content Management System)

Connaissances préalables nécessaires:

Connaissance des technologies Web.

Programme:

- Introduction** : Le besoin : création et mises à jour de contenus.
Gestion de différents types de documents.
Traitement séparé de la forme et du fond.
Respect des critères d'accessibilité.
Interventions du public sur les sites, avec éventuellement actions simultanées
- Les solutions techniques** : Interface web
Stockage des contenus dans une base de données
Format XML
Gestion des style par CSS
Utilisation de balises simples pour l'édition des pages
- Les apports des CMS** : Mises à jour facilitées
Modification du style rapide
Possibilité de modifications simultanées
- Produits disponibles** : état de l'art, comparaisons des différents produits
Plone, spip, drupal, joomla, eZPublish, ..

CMS : mise en œuvre de Joomla

Durée: 2 jours
980 € HT

les 23 et 24 Février
les 27 et 28 Avril
les 29 et 30 Juin

les 2 et 3 Septembre
les 3 et 4 Novembre
les 16 et 17 Décembre

Public:

Chefs de projets, développeurs web, administrateurs web

Objectifs:

Comprendre le fonctionnement de Joomla, savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissance des technologies Web.

Programme:

- Introduction** : Présentation de Joomla
Historique du projet
Les apports et points forts
- Installation et configuration** : Prérequis matériel et logiciel
OS supportés
Procédure d'installation et démarrage
- Les utilisateurs** : Enregistrement des utilisateurs :
différentes possibilités, restriction,
formulaire d'enregistrement,
création de groupe d'utilisateurs, ...
- Le site web** : interface administrateur,
changement de style,
moteur de recherche, ..
- Les contenus** : notions d'articles, de catégories, de sections
restriction et ouverture des accès
gestion des articles, menus, contacts, hyperliens, ...
- La sécurité** : Les risques inhérents à la publication web
Quelques solutions :
sauvegardes, mises à jour du produit, sécurisation du serveur

Présentation Web 2.0

Durée: 1 jour
410 € HT

le 25 Février
le 29 Avril
le 28 Juin

le 10 Septembre
le 5 Novembre
le 14 Décembre

Public:

Chefs de projet, architectes, et toute personne souhaitant savoir quelles sont les particularités des technologies web 2.0., et les outils disponibles.

Objectifs:

Connaître les principes techniques du web 2.0, et les principales fonctionnalités et produits disponibles.

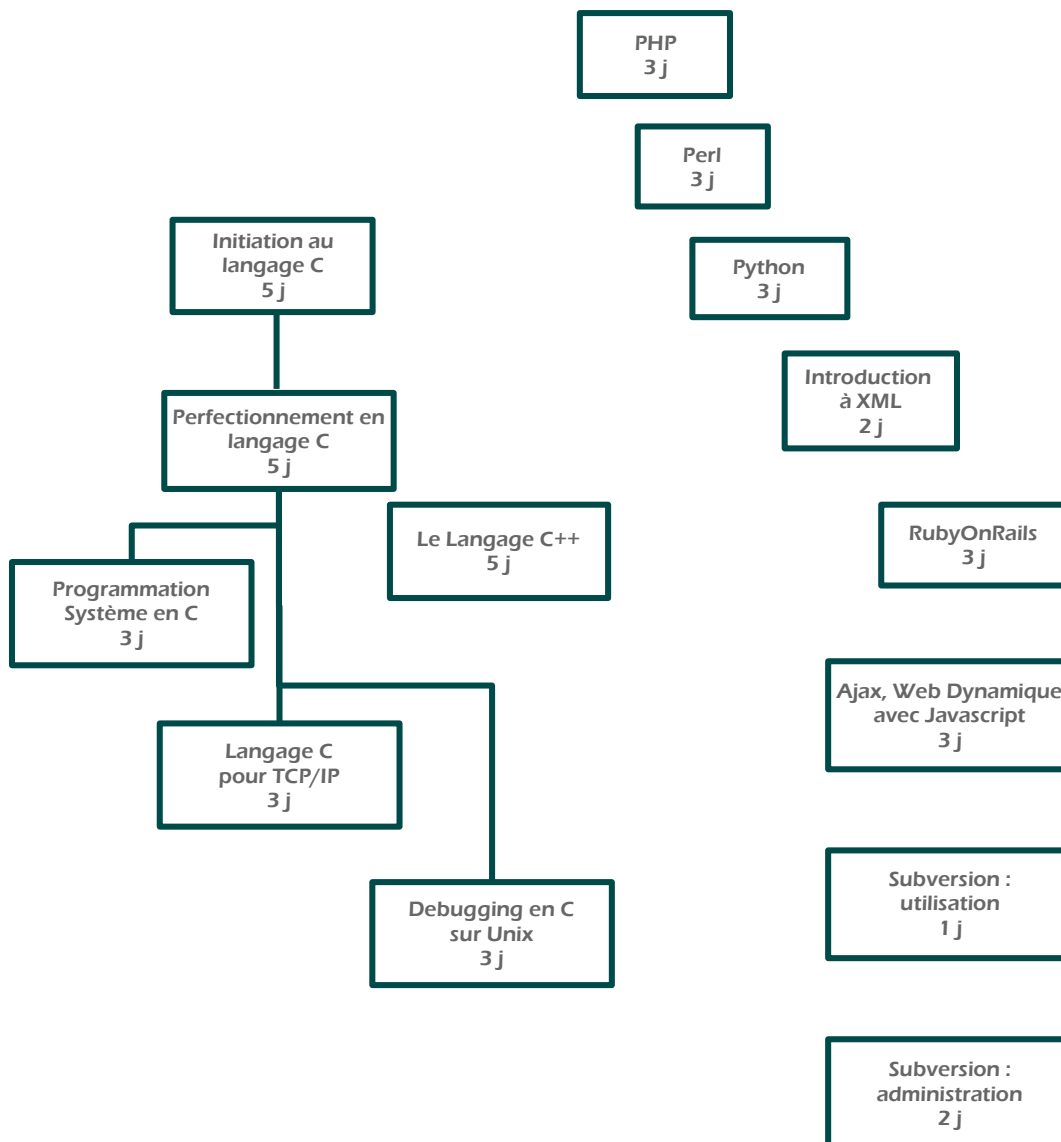
Connaissances préalables nécessaires:

Une bonne connaissance des principes et techniques d'internet.

Programme:

- Introduction** : Historique : le Web 2.0 Summit
Aspects techniques : interfaces web riches, mécanismes collaboratifs.
- Techniques web 2.0** : format HTML, feuilles de style CSS,
XML, javascript,
syndication RSS
Ajax avec XMLHttpRequest
- Les services Web 2.0** : Présentation wiki, galeries, blogs, syndication,...
- Wiki** : Fonctionnalités
Produits disponibles
Utilisation, mise en œuvre, choix techniques à réaliser :
hébergement, logicle, administration du wiki, ..
- Galeries, blogs** : Services disponibles et outils
- Flux RSS** : Principe de la syndication
format RSS, lecteurs de flux, notion d'abonnement,
Navigateurs compatibles
- Google Apps** : Services en ligne : courrier, messagerie instantanée, agenda, ...
Principe. Exemples d'applications.

Filières Langages



Initiation au langage C

Durée: 5 jours
1900 €

du 8 au 12 Février
du 12 au 16 Avril
du 14 au 18 Juin

du 30 Août au 3 Septembre
du 18 au 22 Octobre
du 6 au 10 Décembre

Public:

Tout programmeur en langage C.

Objectifs:

Connaître et maîtriser les concepts de base du langage C. Savoir écrire des programmes simples et acquérir des méthodes de programmation.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire pour suivre ce cours

Programme:

- Le C** : Présentation du langage C; Architecture, syntaxe. Structure d'un programme C. Compilation.
- Contrôle de programme** : Instructions de contrôle. Boucles. Les tests. Branchements.
- Variables** : Déclarations de variables; utilisation des types: variables entières, réelles, structurées. Type statique, registres. Notions sur les variables externes.
- Fonctions** : Fonctions de base: affichage et lecture des données. Entrées/sorties formatées. Fonctions personnalisées: définitions, règles de fonctionnement, récursivité.
- Manipulation de données** : Tableaux, chaînes de caractères. Présentation. Structures. Champs binaires. typedef. Conversions, règles de portée. Opérateurs. Affichage et saisie avancés.
- Librairies** : Introduction à la notion de librairie. La librairie standard. les fichiers inclus. Introduction aux différentes phases de compilation, édition de liens.
- Pointeurs** : Présentation. Utilisation.

Perfectionnement en langage C

Durée: 5 jours
1960 €

du 8 au 12 Mars
du 17 au 21 Mai

du 20 au 24 Septembre
du 22 au 26 Novembre

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C.

Objectifs:

Compléter des connaissances en langage C par une formation approfondie sur les mécanismes fondamentaux de fonctionnement.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les structures et fonctions de base du langage C.

Programme:

- | | |
|------------------|--|
| Rappels | : Architecture d'un programme écrit en C; fonctions de base; variables
Phases de compilation. |
| Variables | : Structures avancées, unions, tableaux, champs binaires, drapeaux.
Particularités des variables statiques, des variables externes.
Rappels et compléments sur l'édition de liens. |
| Fonctions | : Appels récursifs, transmission de liste variable d'arguments.
Pointeurs sur les fonctions.
Applications aux interpréteurs. |
| Mémoire | : Principe d'allocation mémoire; mise en œuvre. |
| Pratique | : Les listes chaînées, les arbres binaires.
Applications à l'organisation des données. |

Le langage C++

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C++ et liés au système d'exploitation.

Objectifs:

Apprendre la programmation objet au travers du langage C++. Maîtriser l'analyse et la conception de programmes orientés objets.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases d'un langage structuré (Pascal, C, ou autre).

Programme:

- POO** : Définitions: besoin, cadre d'utilisation, termes utilisés, organisation, historique.
- Base** : Architecture d'un programme écrit en C++; variables simples; affichage: cout et cin; fonctions; surcharges.
Structure d'un programme: boucles, tests, branchements
- Approche objets** : Structures, classes, constructeurs.
Fonctions membres, fonctions inline, surcharges des membres, accès: public, private, protected. Fonctions amies.
Imbrication d'objets, dérivation, héritage.
Fonctions virtuelles, polymorphisme. Classes abstraites.
- Données** : Les différents types de variables et attributs (tableaux, données statiques, constantes), règles de portées. Autoréférence.
Allocation dynamique: new et delete. Tableaux d'objets
- Compléments sur les opérateurs** : Opérateurs de calcul, comparaison, binaires.
Surcharge d'opérateurs. Surcharge de << et >>.
Fonction-opérateur amie.
- Système** : Gestion des erreurs. Les exceptions.
Contrôle des formats de données, redirections, traitement des fichiers, stream. Les manipulateurs, analyses de chaînes.
- Compilation** : Préprocesseur, contrôle de compilation, macros instructions.
Patrons de fonctions, patrons de classes. Fichiers inclus.

Programmation système en langage C

Durée: 3 jours

Prix et dates: nous consulter

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C et liés au système d'exploitation.

Objectifs:

Compléter des connaissances en langage C par une formation approfondie sur les mécanismes d'accès au système d'exploitation. L'accent sera particulièrement sur les fichiers, pointeurs, allocations de mémoire, communications et les bibliothèques systèmes.

Connaissances préalables nécessaires:

Il est demandé aux participants de bien connaître les structures et fonctions de base du langage C.

Programme:

Rappels	: Architecture d'un programme écrit en C; fonctions de base. Variables. Phases de compilation. Rappels sur les systèmes d'exploitation des ordinateurs.
Variables	: Structures, unions, tableaux. Pointeurs.
Fonctions	: Rappels sur les pointeurs de fonctions. Applications à la structure des fichiers.
Mémoire	: Les fonctions d'allocation et les appels systèmes: sbrk, malloc, realloc. Ecriture d'un allocateur de mémoire.
Communications	: Signaux et interruptions. Sémaphores et appels concurrents. Segments de mémoires partagées.
Parallélisation	: Création de processus fork(),clone(),setsid(). Threads. NPTL. Création, destruction. Synchronisation entre threads, détachement. Attributs. Gestion de la pile. Mutex. Conditions variables. Signaux, interruptions. Ordonnancement.
Systèmes	: Organisation. Gestion de processus. Pilotes de périphériques. Application au système Unix

Langage C pour TCP/IP

Durée: 2 jours

Prix et dates: nous consulter

Public:

Toute personne désireuse d'approfondir ses connaissances sur les réseaux TCP/IP.

Objectifs:

Comprendre les mécanismes fondamentaux de TCP/IP au travers d'une programmation d'outils en langage C: traceroute, ping et analyseur de trames.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP et du langage C.

Programme:

- | | |
|--------------------|---|
| Rappels | : Rappels sur le fonctionnement de TCP/IP et de son utilisation par le système d'exploitation |
| Lecture IP | : Programmation d'une interface d'accès à la couche réseau. Lecture de trames réseau. Décomposition et étude d'un analyseur de trames. Modification du programme. |
| Ecriture IP | : Envoi d'une trame ICMP sur le réseau. Ecriture d'un ping simplifié. Ecriture de l'utilitaire traceroute. |

Debugging en C sur Unix

Durée: 1 jour

Prix et dates: nous consulter

Public:

Développeurs en langage C. Exploitants et administrateurs Unix.

Objectifs:

Maîtriser les différentes phases de la compilation d'un programme écrit en C sur une machine Unix. Connaître les outils d'exploitation permettant de suivre une exécution, d'en analyser les performances.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du système Unix et du langage C.

Programme:

- | | |
|----------------------|--|
| Compilation | : Les différents langages de programmation. Comment un programme utilise les appels systèmes.
Les différentes phases de compilation
Reconnaissance des types de fichiers. Extraction du langage depuis un exécutable binaire .
Exploitation des fichiers objets: objdump, objcopy, readelf, size, strings |
| Bibliothèques | : Bibliothèques standards.
Fonctions et appels systèmes.
Commandes d'exploitation des bibliothèques: ar, ranlib.
Gestion des noms et symboles: nm, strip |
| Projet | : Outils de développement d'un projet: lint, make, cb.
Découpage d'un programme en modules. Installation par make. Structure d'un Makefile. |
| Debbuging | : Les debuggers: dbx, dbg, adb. Exploitation des fichiers core.
Création d'un dump mémoire. Informations nécessaires aux équipes de développement pour corriger les erreurs de programmation. |

Subversion utilisateur

Durée: 1 jour
390 € HT

le 15 Février
le 19 Avril
le 21 Juin

le 25 Août
le 11 Octobre
le 1er Décembre

Public:

Chefs de projet, développeurs, ingénieurs logiciels, architectes d'application, équipes de qualification

Objectifs:

S'approprier les principes de travail sur un référentiel partagé pour la gestion de version

Connaissances préalables nécessaires:

Connaissance en gestion de projet, développement.

Programme:

- Introduction** : Le besoin : gestion de version, développement projet.
Historique de Subversion.
Caractéristiques.
- Les fonctionnalités** : Contrôle des versions
Gestion des fichiers, répertoires et des modifications
Historique des changements
Fonctionnement en réseau, partage des fichiers
- Les concepts** : Définitions :
référentiels, branches, étiquettes, révision
- Les outils clients** : Plugin Eclipse,
ant
le mode commande : svn

Subversion Administrateur

Durée: 2 jours
790 € HT

les 16 et 17 Février
les 20 et 21 Avril
les 22 et 23 Juin

les 26 et 27 Août
les 12 et 13 Octobre
les 2 et 3 Décembre

Public:

Administrateurs de l'outil subversion

Objectifs:

S'appropriier les principes de travail sur un référentiel partagé pour la gestion de version.

Connaissances préalables nécessaires:

Connaissance en gestion de projet, bonne connaissance de l'administration du système Linux

Programme:

- | | |
|------------------------------------|---|
| Introduction | : La Gestion de Configuration Logicielle
- Gestion de version, développement projet
- Historique de Subversion |
| Les caractéristiques de Subversion | : Architecture client/serveur
- Le Référentiel
- Interfaces d'accès: mode commande, interfaces graphiques |
| Fonctionnalités | : - Contrôle des versions
- Gestion des fichiers, répertoires et des modifications
- Historique des changements
- Fonctionnement en réseau, partage des fichiers |
| Concepts et terminologie | : Référentiels
- Branches
- Etiquettes
- Copies
- Révision |

Subversion Administrateur

- Travail en équipe avec Subversion** : Les outils client: Plugin Eclipse, ant, le mode commande, Tortoise
- Récupérer les sources
 - Création d'une branche
 - Les fusions
 - Gestion des branches
 - Valider/Annuler des modifications
 - Résolution des conflits
 - Gestion des verrous
 - Création d'une étiquette
 - Analyse de l'historique
 - Bonnes pratiques
- Configuration du serveur** : Choix du serveur
- Serveur svnserv
 - svnserv avec ssh
 - Serveur http Apache
 - Configuration réseau
 - Gérer plusieurs méthodes d'accès
 - Gestion de la sécurité : utilisateurs, groupes, droits d'accès
- Administration du référentiel** : Stratégie de déploiement
- Mode de stockage : système de fichier ou base de données
 - Création et configuration du référentiel
 - Sauvegarde et restauration
 - Surveillance et optimisation
 - Les outils svnlook et svn admin
 - Ajout de projets
 - Migration
 - Gestion de l'historique

Programmation PHP

Durée: 3 jours

Public:

Tout développeur souhaitant acquérir les bases de la programmation PHP 5.

Objectifs:

Comprendre les principes de base, connaître la syntaxe, les fonctions et le développement objet en PHP.

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

- | | |
|---------------------|---|
| Introduction | : Présentation, historique, caractéristiques, domaines d'utilisation
Les apports de PHP 5.
Principe de fonctionnement avec apache
Mise en œuvre , configuration php.ini |
| Syntaxe de base | : Déclarations de variables
Types de données
Les expressions
Les opérateurs
Structures de contrôle
Les tableaux, les variables persistantes
Quelques exemples simples |
| Les fonctions | : Bibliothèques de fonctions :
graphiques, fichiers, http, imap, mathématiques, pdf ... |
| Les données | : Accès aux bases de données, la gestion des connexions.
Travaux Pratiques avec MySQL, les différentes bases de données accessibles. |
| Programmation Objet | : La notion d'Objet, les constructeurs, l'héritage, les interfaces, les méthodes et classes abstraites |
| Exploitation | : La gestion des exceptions |
| XML et webServices | : Utilisation de libxml2 (SAX, DOM, XSLT, Soap) |
| Composants PHP | : Pear : PHP Extension and Application Repository |

Python

Durée: 3 jours

Public:

Tout développeur souhaitant acquérir les bases de la programmation en Python.

Objectifs:

Connaître les possibilités du langage Python, et savoir écrire les programmes de base.

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

- | | |
|---------------------|--|
| Introduction | : Présentation Python
Positionnement par rapport à d'autres langages |
| Les bases | : les données : les numériques, chaînes de caractères, les listes
la structure des programmes : boucles, ... les fonctions
les entrées/sorties, accès aux fichiers
la programmation objet : les classes |
| Exécution | : différentes méthodes d'exécution (interpréteur, ligne de commande, fichier, ...)
écriture de scripts |
| Exploitation | : la gestion des erreurs
les exceptions
interfaces vers les langages C et C++ |

Perl

Durée: 3 jours
1050 € HT

du 15 au 17 Mars
du 10 au 12 Mai

du 6 au 8 Septembre
du 22 au 24 Novembre

Public:

Tout développeur souhaitant acquérir les bases de la programmation en Perl.

Objectifs:

Comprendre les principes de base de Perl, connaître la syntaxe de base.

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

- Introduction** : Présentation de Perl : caractéristiques, positionnement par rapport à d'autres langages (C, Java, Shell, PHP)
Plate-formes d'utilisation
- Les bases** : Syntaxe, structure des programmes,
variables, différents types de données, constantes particulières
Les opérateurs : de liaison, de décalage, ...
Manipulation de chaînes
Expressions régulières, recherche de motif, métacaractères,
Structures de contrôles,
Entrées-sorties standards
- Les fonctions :** : Principe, appel d'une fonction, passage des arguments,
attributs
- Références** : Définition, création de références, références symboliques
- Tableaux** : Manipulation de tableaux, hachage,
sauvegarde des tableaux
- Exécution** : différentes méthodes d'exécution
compilation , exécution
- L'environnement** : opérations sur les fichiers
communication inter-processus
- Les modules** : Création de modules Perl, utilisation
CPAN : les modules archivés
- Scripts pour administrateurs :** Exemples de procédures d'administration

XML Introduction

Durée: 1 jour

Prix et dates: nous consulter

Public:

Toute personne souhaitant acquérir des notions de base sur le langage XML.

Objectifs:

Connaître les fonctionnalités du langage XML.

Connaissances préalables nécessaires:

Une connaissance de base des concepts de l'internet et du langage HTML sont nécessaires pour suivre ce cours.

Programme:

- Introduction** : Besoin, historique
Intégration avec l'environnement existant.
Définitions: HTML, SGML, DTD, XML (eXtensible Markup Language)
- Notions de base** : Concepts, nommage des éléments, importations d'espaces de nommage, répartition des définitions
Les entités d'un DTD: les entités de caractères, les attributs (CDATA, ID, ENTITY, ENTITIES, NMTOKEN, NMTOKENS)
- Développements** : Les types d'éléments: types simples, complexes.
Définitions des éléments.
Instanciation d'un schéma.
Liens: simples, étendus
Présentation des API de programmation : DOM, SAX
- Exploitation** : Feuilles de style XSL, CCC. Les sous éléments de XSL: XSLT et XSL-FO.
Utilisation de XML pour les sites Web
Utilisation en base de données. Présentation de JDO.
Présentation de Hibernate et Castor.

Ruby On Rails

Durée: 3 jours
1 180 € HT

du 1er au 3 Février
du 6 au 8 Avril
du 31 Mai au 2 Juin

du 25 au 27 Août
du 4 au 6 Octobre
du 1er au 10 Décembre

Public:

Les développeurs d'application Web souhaitant programmer avec RoR.

Objectifs:

Savoir programmer en Ruby avec le framework Rails.

Connaissances préalables nécessaires:

Connaître un langage de programmation, de préférence objet.

Programme:

- Présentation** : Ruby, Rails, historique et caractéristiques
Les apports de Ruby On Rails
Architecture des applications : Modele, Vue, Controleur
- Mise en œuvre** : Installation
Développement rapide d'une première application.
- Ruby** : Présentation
Spécificité des noms Ruby
Les objets : classes, méthodes, modules, tableaux
Structure d'un programme,
La documentation RDoc
- Rails** : Configuration, architecture
Active Support
Active Record :
accès aux tables, connexion, relations, associations, gestion des transactions
Action Controller :
requetes, routage, filtrage, ...
Action View :
les différents formats, les assistants
- Extensions** : Développer des applications Ajax avec Rails
Action Mailer:
émission, réception de mails
WebServices en SOAP ou XML-RPC
- Exploitation** : Sécurité : les risques et les solutions
Techniques de déploiement
Maintenance, analyse des logs

Ruby On Rails

AS201

Développement Web avec Ajax

Durée: 2 jours

Prix et dates: nous consulter

Public:

Développeurs web.

Objectifs:

Savoir programmer avec les technologies Ajax.

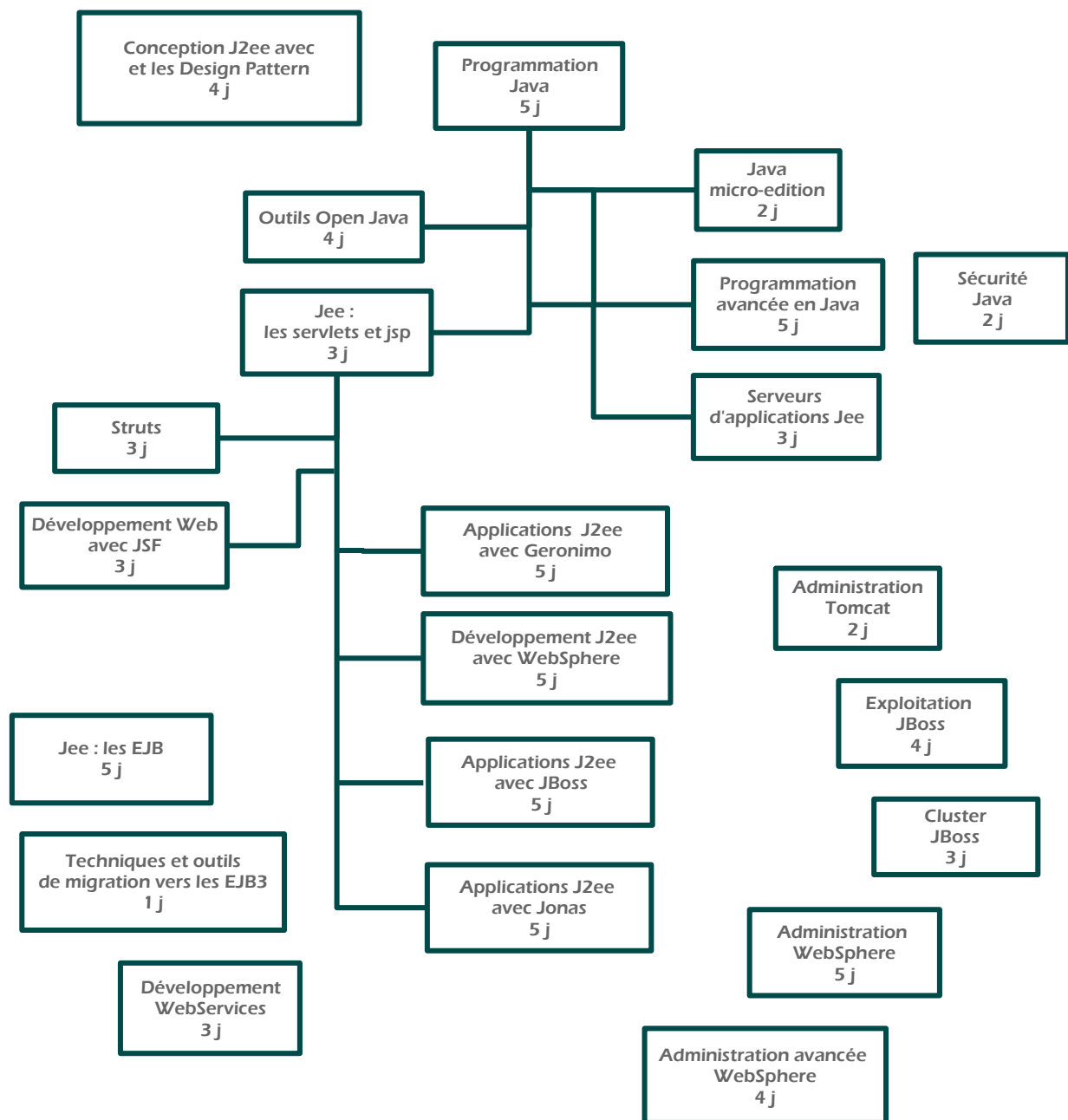
Connaissances préalables nécessaires:

Il est demandé aux participants de connaître le principe de fonctionnement d'Internet et du Web, le langage HTML. La connaissance d'un langage de programmation est utile.

Programme:

- Introduction** : Besoin, historique, cadre d'utilisation. Les apports d'AJAX, rôle et limitations. Les technologies mises en œuvre: XML, HTML, CSS. Architecture applicative.
- Bases XML :** : Nommage des éléments, importations d'espaces de nommage, répartition des définitions.
Les entités d'un DTD: les entités de caractères, les attributs (CDATA, ID, ENTITY, ENTITIES, NMTOKEN, NMTOKENS).
XML "bien formé", XML "valide". Accès aux éléments et attributs.
- Pages HTML** : Mode d'appel des scripts.
Formulaires HTML: comment créer une zone de saisie? Les différents modes de saisie HTML
Les feuilles de style CSS et XML
- Développement avec AJAX** : Le Modèle AJAX : principe Asynchronous Javascript Xml.
Principe de XMLHttpRequest. Requêtes Http/XML vers le serveur. Traitement du résultat avec XML.
Exemple avec l'API Google
- Mise en œuvre** : Instanciation. Gestion des exceptions: try et catch.
REQUETE : méthodes et propriétés. Exploitation des requetes GET et des requetes POST
- Exploitation des données** : Propriétés ".onreadystatechange", ".readyState", ".status" et ".statustext"
Utilisation des données de ".responseText" et ".responseXML"
Gestion des événements.

Filières Langages Java



Programmation Java

Durée: 5 jours
1860 €

du 1er au 5 Février
du 12 au 16 Avril
du 7 au 11 Juin

du 6 au 10 Septembre
du 25 au 29 Octobre
du 6 au 10 Décembre

Public:

Développeurs Java.

Objectifs:

Connaître les outils nécessaires à la programmation et l'exécution de programmes en Java, et acquérir les notions de base de programmation objet en Java.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation.

Programme:

- Présentation** : Utilisations de Java: sur le Web, en application lourde, en application distribuée, dans une architecture JEE, dans les cartes à puce, systèmes embarqués.
Définitions: JDK, JRE. Présentation du JDK 1.5
- Conception de base** : Programmation objet, les réutilisables. Syntaxe d'une classe. Principe de l'encapsulation. Constructeurs, attributs et méthodes. Accesseurs.
Différence entre objet et classe. Instanciation. Conventions de nommage.
- Notions de base** : Syntaxe du langage: Variables, références;
Instructions et opérateurs, Tableaux, Structure d'un programme.
Contrôle: tests, branchements, boucles
- Classes de base** : La classe System.
Les chaînes de caractères. Date et heure. Les classes Integer, Double.
- Exécution** : Exceptions de base. La structure try/catch.
Multi-tâches (threads). Les groupes de Thread. Les horloges.
Synchronisation entre tâches.
- Fichiers** : Les classes principales d'entrées/sorties.
Accès à un fichier, copie d'un fichier. Méthodes d'accès aux fichiers

Programmation Java

- Graphique** : Les classes graphiques, gestion d'images, présentation, animation.
Gestion des évènements. Gestion de la souris. Applets.
- Développement** : Abstraction et interfaces. Création et utilisation d'interface.
Exceptions, pile d'exécution. Création et utilisation d'exceptions.
- Outil de développement** : Eclipse : environnement de développement extensible par plugins.
Principes de fonctionnement, installation et premiers pas avec Eclipse.
JDT : Java Development Tools, eclipse Europa

Programmation avancée en Java

Durée: 5 jours
1970 € HT

du 18 au 22 Janvier
du 1er au 5 Mars
du 3 au 7 Mai

du 19 au 23 Juillet
du 20 au 24 Septembre
du 15 au 19 Novembre

Public:

Développeurs Java.

Objectifs:

Compléter des connaissances de base, Acquérir un niveau d'expertise en Java.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base du langage Java.

Programme:

- Le langage** : Travaux pratiques de rappels. Présentation du JDK utilisé. Nouveautés du JDK 1.5/1.6
- Bibliothèques** : Rappels sur la compilation. Principe des librairies, JFC, Java Beans, Swing. Gestion des paquets. Fichiers jar. CLASSPATH. Présentation de ant.
- Collections** : Manipulation d'ensembles d'objets. Listes, collections, ensembles, dictionnaires.
- JDBC** : Mécanismes de connexion aux bases de données. Principes de JDBC. Utilisation des méta données. Exploitation du ResultSet. Gestion du transactionnel. Exemple d'application transactionnelle de Java à SQL
- XML** : Présentation des API de programmation : DOM, SAX. TP: écriture d'un mini parseur XML.
- Réseau** : Les classes principales d'accès au réseau
Programmation par socket, sérialisation. Transferts d'objets au travers du réseau. Création d'une application client/serveur réseau.
Accès aux ressources partagées, synchronisation, verrous. Utilisation des threads.
- Accès concurrents** : Accès aux ressources partagées, synchronisation, verrous
Utilisation des threads
- Java Management eXtension** : Supervision avec JMX
Principe des MBeans, et exemples de MBeans standards fournis à partir du jdk 1.5

Programmation avancée en Java

- Sécurité** : Introduction à la sécurité Java. Protection du système vis à vis des applications tierces: SecurityManager, ClassLoader. JCE. Chiffrement, chiffrement asymétrique, hachage
- Intégration** : exécution de commandes système depuis Java.
appel de programme en C depuis Java: classes natives.
appel de Java depuis le C. Instanciation d'une JVM.

Sécurité Java

Durée: 2 jours
750 € HT

les 18 et 19 Février
les 22 et 23 Avril
les 24 et 25 Juin

les 30 et 31 Août
les 14 et 15 Octobre
les 29 et 30 Novembre

Public:

Toute personne désirant approfondir ses connaissances en langage Java au niveau de la sécurité des applications.

Objectifs:

Connaître les risques potentiels dans l'utilisation de Java, et les parades à mettre en œuvre.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base du langage Java.

Programme:

- Besoins** : Les risques : en fonction des différents modes d'utilisation de Java (applets, application, servlets)
- Sécurisation de la JVM** : Limites naturelles imposées par Java : gestion mémoire.
Contrôle du bytecode par la machine virtuelle.
Mise en œuvre du SecurityClassLoader
- Protection de l'exécution** : Exécution protégée : SecurityManager, ClassLoader.
Surcharge des méthodes d'accès: lecture, écriture, exécution, ouverture de socket, autorisation de connexions...
TP: Protection des accès sur le disque local d'une application.
- Chiffrement** : mécanisme de signature. Création de clés publiques et privées.
Les clés RSA, DSA.
Signature d'un document. Les algorithmes SHA1withDSA, MD5withRSA.
Les MessageDigest. Les algorithmes MD2, MD5, SHA-1, SHA-512
TP: Vérification de l'authenticité d'un document
- Certificats** : Cycle de vie d'un certificat. La fabrique de certificats Java.
Les certificats de modification X509.
- Contrôle** : Rappel sur les ACL. Le paquetage java.security.acl. Ajout d'entrée, vérification d'accès.

Outils Open Source Java

Durée: 4 jours

Prix et dates: nous consulter

Public:

Développeurs Java.

Objectifs:

Connaître les outils de développement Java en Open source, en comprendre les mécanismes de base, savoir trouver de la documentation. Cette formation est accompagnée de nombreux travaux pratiques.

Connaissances préalables nécessaires:

Programmation de base en Java.

Programme:

- Open source** : Présentation, principes de base.
Les principales licences (ASF, GPL, BSD, MPL, ...)
- Editeurs** : Eclipse
Le projet Eclipse. Fonctionnalités.
Les notions de Perspectives, Vues, Editeurs. Les plugins.
Autres outils: kdevelopper, Anjuta, netbeans,
- Ant** : Présentation
Construction d'applications Java : différentes phases
(compilation, déploiement, archivage, documentation, ...)
Automatisation, construction d'une nouvelle tâche Ant
- Maven** : POM : Project Object Model
Normalisation du processus de développement.
Maven et Ant
- Struts** : L'architecture MVC : Model View Controller.
Technologies Java, XML
Le principe, les outils disponibles, les tag library, l'automatisation,
les classes de gestion d'erreur
- JSF** : Les apports de JSF par rapport à Struts
Les différentes implémentations: la référence Sun, Apache
MyFaces, Smile, ECruiser
Travaux pratiques avec Apache MyFaces et Tomcat
Utilisation des pages JSF avec AJAX

Outils Open Source Java

- ORM** : Object Relational Mapping
Le principe, les caractéristiques des différents outils :
Hibernate, OJB, Castor, iBatis, EJB3
- Commons** : Sous projet de jakarta : Commons Proper et Commons
SandBox.
Présentation de quelques composants : BeanUtils, CLI,
Collections, Digester, IO, ...
- Tests** : Junit pour les tests unitaires
OpenSTA : tests applications web (stress et performance)
- Documentation** : Le projet Apache Jakarta, Source forge, OSTG

Java Micro Edition

Durée: 2 jours

Prix et dates: nous consulter

Public:

Développeurs Java souhaitant programmer avec Java 2 Micro Edition.

Objectifs:

Comprendre le fonctionnement de J2ME, savoir élaborer l'architecture et choisir les outils à mettre en œuvre en fonction du matériel concerné.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du langage Java.

Programme:

- Présentation** : Le besoin : les applications embarquées, les objets disposant de ressources restreintes (téléphones mobiles , PDA, ...)
- Fonctionnement de J2ME** : optimisations java par rapport à un jre.
Les machines virtuelles, configurations, et profils adaptés à chaque type de matériel.
Les deux types de terminaux :
- téléphones mobiles et PDA (configuration CLDC)
- les terminaux connectés (configuration CDC)
- CLDC** : La notion de KVM. Les limites. Les profils MIDP.
- CDC** : La machine virtuelle CDM. Les profils CDC.
- Programmation J2ME** : Organisation d'une midlet.
Les midlets, exemples:
- application graphique de base avec menu
- application client/serveur entre un téléphone portable et un serveur.
- J2ME Wireless Toolkit** : Les API:
Bluetooth, J2ME Web Services, Security and Trust Services, Mobile Media.

JEE : les servlets et JSP

Durée: 3 jours
1070 € HT

du 11 au 13 Janvier
du 15 au 17 Mars
du 10 au 12 Mai

du 5 au 7 Juillet
du 13 au 15 Septembre
du 8 au 10 Novembre

Public:

Les développeurs de servlets et de pages JSP.

Objectifs:

Comprendre les mécanismes des servlets et JSP. Savoir créer des pages Web dynamiques.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques de base Internet (HTML, serveur HTTP).

Programme:

- Définitions** : Application Java, applet, servlet, JSP
- Besoins** : pages HTML dynamiques, communication avec des bases de données, applications Java, et inter-servlets.
Rappels sur le modèle JEE, les composants d'un serveur d'application Java.
- servlets** : Cycle de vie: Instanciation, initialisation, traitement des requêtes et arrêt des servlets, persistance, sessions
contexte de session HTTP, cookies, réécriture d'URL.
méthodes d'accès, objet de requête et objet de réponse
chaînage de servlets, utilisation en filtres, communication entre servlets.
Travaux pratiques : modification des images d'un site
- JSP** : Java Server Pages : positionnement par rapport aux servlets.
Balises JSP. Redirections et enchainements. Cycle de vie.
- Architecture** : Découplage Java/HTML à l'aide de JavaBean. Exemple avec le pattern commande.
- XML** : Extensible Markup Langage : présentation, utilisation dans les fichiers de configuration
- Gestion des données** : Connexion aux bases de données : JDBC, pool de connexions.
Accès aux données depuis les servlets. Transactions. Exécution d'instructions SQL, utilisation de blob. Mapping XML.

JEE : les servlets et JSP

- Mise en œuvre : échange de données entre un client HTML et une base de données sur un serveur Web.
Exemple d'application transactionnelle de Java à SQL.
Dialogue Applet/servlet
- Ajax JSP Tag Library : Utilisation d'Ajax dans les pages JSP : les balises disponibles

JEE : les EJB

Durée: 5 jours
2250 € HT

du 18 au 22 Janvier
du 22 au 26 Mars
du 17 au 21 Mai

du 19 au 23 Juillet
du 20 au 24 Septembre
du 15 au 19 Novembre

Public:

Développeurs d'applications JEE.

Objectifs:

Comprendre les principes de fonctionnement et maîtriser le développement et la mise en œuvre des EJB. Nos cours sont fondés sur de nombreux travaux pratiques permettant d'assimiler les technologies EJB 3.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation en Java, ainsi que quelques notions de bases de données.

Programme:

Le modèle JEE

: Spécifications du JEE : les composants d'un serveur d'application Java, positionnement des EJB, architectures d'applications.
Positionnement des différents éléments: container, serveur, services associés: JDBC, JNDI, JMS, JTA, JIDL, JMS
Pool d'EJB. Enregistrement des EJB, identification.
Déploiement sur différents serveurs d'application du marché : Geronimo, Websphere, JBoss, GlassFish, ...

Développement

: Composants d'une application d'entreprise: .ear, .jar, .rar, .war
Types d'EJB : EJB entité, EJB session, EJB Message-driven.
Persistance et transactions.
Les annotations spécifiques EJB 3
Méthodes de base: `ejbCreate()`, `ejbActivate()`, `setSessionContext()`, `ejbStore()`,...

Mise en œuvre

: Apport du jdk1.5. Annotations et génériques. Principes de mise en œuvre des EJB3.
Simplification de la définition des interfaces. Utilisation des annotations en remplacement du descripteur de déploiement.
Simplification de la correspondance objet/relationnel.
Les serveurs d'EJB3: `ejb3deploy` sous JBoss, WebSphere 6.1.

JEE : les EJB

- EJB Session** : EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions. Dépendance d'injection. Intercepteurs de méthodes métier. Application: création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"
- EJB Entité** : Persistance avec les EJB. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes de recherche (query). Gestion des jointures en EJB3. Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY. Mapping par défaut. Utilisation de l'EntityManager. Relations entre EJB (one-one,one-many,many-many). Le langage EJB-QL. Héritage. Synchronisation avec la base de données, détachement/attachement. Application: création d'une application utilisant des jointures Many/Many
- Gestion des données** : Pool de connexions. Correspondance objet/relationnel. Mapping XML.
- EJB MessageDriven** : Développement de composants orientés message. Principe des MOM (Middlewares Orientés Message). Présentation de JMS. Développement d'un EJB Message Driven (MDB). L'interface JMS MessageListener. Traitement asynchrone. La méthode onMessage(). Fichiers de déploiement. Cycle de vie des MDB. Gestion des erreurs.
- Interopérabilité** : Compatibilité ascendante des EJB3. Utilisation d'EJB en version 2 avec des EJB 3. Utilisation des fichiers de déploiement xml avec les EJB 3. Mise en œuvre d'une application faisant cohabiter les deux versions d'EJB.
- Transactions** : Les transactions distribuées: commit à deux phases. Présentation JTA et JTS. Annotations de transaction. Gestion des exceptions dans un contexte transactionnel. Mise en œuvre d'une application EJB3 transactionnelle. Mise en évidence des retours arrières (rollback) et de la propagation du contexte transactionnel.
- Sécurité** : Présentation de JAAS. Mise en œuvre dans le conteneur d'EJB. Rôles et permissions sur les méthodes. Intégration de la sécurité dans le serveur d'applications.

Développement Web Services

Durée: 3 jours
1490 € HT

du 25 au 27 Janvier
du 29 au 31 Mars
du 25 au 27 Mai

du 26 au 28 Juillet
du 27 au 29 Septembre
du 22 au 24 Novembre

Public:

Les chefs de projets et développeurs souhaitant concevoir et développer des web services.

Objectifs:

Savoir développer des applications utilisant les techniques des Web services.

Connaissances préalables nécessaires:

Ce cours présente les WebServices en environnement Java.

Programme:

- Introduction** : Historique. Définitions.
Les différents éditeurs, et leurs produits.
Architecture technique.
Rappels sur XML.
- Le protocole SOAP** : Simple Object Access Protocol pour l'échange de messages XML. Structure d'un message.
- Les annuaires UDDI** : Universal Description, discovery and Integration pour la recherche des services web disponibles.
Les annuaires publics.
Structure des données.
Travaux pratiques avec jUDDI.
- Les API Java XML** : JAXP, JAXB, SAAJ : pour le traitement des données XML
JAX RPC, lancement de procédures distantes
JAXM, messages XML; JAXR, identification de services web
- Le Langage WSDL** : Web Service Description Langage
- Mise en œuvre** : Travaux Pratiques : écriture de services web avec Axis, et Xfire.
- Sécurité** : Le besoin. Les solutions actuelles : authentification http, ws-security.

Techniques et outils de migration vers les **AS007** EJB3

Durée: 1 jour
450 € HT

le 28 Janvier
le 1er Avril
le 28 Mai

le 29 Juillet
le 30 Septembre
le 25 Novembre

Public:

Les développeurs d'applications JEE

Objectifs:

Connaître les principales différences entre les EJB3 et les EJB2, et les difficultés et solutions lors de la migration vers les EJB3.

Connaissances préalables nécessaires:

Bonnes connaissances de Java/JEE et des EJB

Programme:

Rappels sur les standards EJB 2.1 : Les interfaces, méthodes d'accès, descripteurs de déploiement

Nouveautés EJB 3.0 : Utilisation des annotations.
Exemples de Bean Session, Entité et MD en v2.1 et leur équivalent en v3

Interopérabilité : Compatibilité ascendante des EJB3. Utilisation d'EJB en version 2 avec des EJB 3.
Utilisation des fichiers de déploiement xml avec les EJB 3
Mise en œuvre d'une application faisant cohabiter les deux versions d'EJB.

Compatibilité et migration : Organisation des migrations, les limites de la cohabitation des EJB2 et EJB3
Les jointures entre EJB 2.1 et EJB3, le service ejb3-deployer dans JBoss, ...
Les solutions techniques

Automatisation : Migration en masse des composants EJB : les outils du marché

Serveurs d'application JEE

Durée: 3 jours
1 120 € HT

du 1er au 3 Février
du 6 au 8 Avril
du 31 Mai au 2 Juin

du 25 au 27 Août
du 4 au 6 Octobre
du 1er au 3 Décembre

Public:

Les chefs de projets et toute personne souhaitant comprendre le fonctionnement de l'architecture JEE.

Objectifs:

Connaître l'architecture et les principes des serveurs d'applications. Savoir concevoir une application avec le modèle JEE.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

- Serveurs d'application** : Introduction. Transactions. Architecture des applications web. Les différents éléments et leurs rôles
- Le modèle JEE** : L'architecture JEE. Le modèle JEE. servlets, Java Server Pages, EJB. Spécifications. Les composants d'un serveur d'application Java
- Les produits** : Présentation de différents serveurs d'application du marché : Geronimo, JBoss, Jonas, WebLogic, WebSphere
Comparatif produits: version de JDK, type d'administration, automatisation, industrialisation, besoin en ressources, support des EJB3, ...
Portabilité des applications JEE.
- Administration** : Définition des différents objets à gérer : serveurs Web, serveur d'application, moteur de servlets, container, EJB, hôtes virtuels, connecteurs JDBC.
Le service de nommage JNDI.
- Conteneur Web** : Servlets, pages JSP:pages HTML dynamiques, communication avec bases de données et applications Java.
Les frameworks de développement : objectifs et techniques mise en œuvre
Struts, JSF, Apache MyFaces, Spring
- Développement avec les EJB** : Les EJB : spécifications (état actuel et limites).
EJB entité, EJB session, EJB Message-driven. Apports des EJB3.
Jointures. Transactions. Présentation de EJB-OL. Relations entre le développeur d'EJB et le DBA.

Serveurs d'application JEE

- Intégration** : Transactions. Utilisation de JTA, JTS. Transactions explicites. Transactions gérées par container.
Sécurité: JAAS, rôles, groupes de permissions.
Mécanismes de connexion aux bases de données : JDBC, pool de connexions.
Connexion interapplicatives avec JCA.
- Déploiement et exploitation** : Cycle de vie d'une application. Industrialisation. Les fonctions à assurer: code, assemblage, nommage, création des fichiers de description en XML, des fichiers jar, mise en production.
Les outils de développement: eclipse, WSAD.
Les outils de mise en production.

Développement JEE avec WebSphere

Durée: 5 jours
2250 € HT

du 18 au 22 Janvier
du 22 au 26 Mars
du 17 au 21 Mai

du 19 au 23 Juillet
du 20 au 24 Septembre
du 15 au 19 Novembre

Public:

Développeurs et intégrateurs d'applications JEE avec le serveur d'application WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere 6.1 et 7.0. Savoir concevoir, développer, et déployer une application..

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques Internet (serveur HTTP, HTML,...) et quelques notions de bases de données.

Programme:

- Le modèle JEE** : les composants d'un serveur d'application Java : serveur HTTP, moteur de servlets, serveur d'EJB
- WebSphere** : Présentation des fonctionnalités du produit WebSphere
La terminologie et les objets WebSphere Application Server : application d'entreprise, serveur d'application, application web, sources de données, pilote JDBC, hôtes virtuels.
Présentation des différents métiers autour de WebSphere : développeur, intégrateur, administrateur.
- Servlets, pages JSP** : pages HTML dynamiques, communication avec bases de données et applications Java, communication avec d'autres servlets
- Développement** : Composants d'une application d'entreprise: .ear, .jar, .rar, .war
Déploiement dans WebSphere.
Utilisation des outils d'assemblage d'applications: wrd, rad, ...
Regroupement de composants unitaires en une application.
Création d'une application sur-mesure (créer un .ear, un .war, etc.).
- EJB** : Types d'EJB : EJB entité, EJB session, EJB Message-driven.
Persistance et transactions.
Méthodes de base: ejbCreate(), ejbActivate(), setSessionContext(), ejbStore(),...
Fichiers de description en XML. Les fichiers spécifiques de WebSphere: ibm-application-bnd/ext/ext-pme.xml, ibm-ejb-jar-*.xml, ibm-web-*.xml.

Développement JEE avec WebSphere

- EJB Session** : EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions.
Application: création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"
- EJB Entity** : Persistance avec les EJB BMP, CMP, CMP2. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes findBy. Gestion des jointures.
Application: création d'un EJB BMP persistant, d'un CMP de type "Compte Bancaire" et de CMP avec jointure.
Relations entre EJB (one-one,one-many,many-many). Le langage EJB-QL
Correspondance objet/relationnel. Mapping XML.
- Intégration** : Transactions. Utilisation de JTA, JTS. Transactions explicites. Transactions gérées par container. Mise en évidence du retour arrière déclenché par le conteneur WebSphere.
Sécurité. Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.
- Introduction aux EJB3** : Apport du jdk1.5. Annotations et génériques. Principes de mise en œuvre des EJB3.
Simplification de la définition des interfaces. Utilisation des annotations en remplacement du descripteur de déploiement.
Simplification de la correspondance objet/relationnel.
Compatibilité ascendante. Les EJB3 sous WebSphere 6.1.
- EJB Session en version 3** : Reprise de l'EJB Caddie. Migration en version 3. Dépendance d'injection. Intercepteurs de méthodes métier.
- EJB Entité en version 3** : Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY. Mapping par défaut. Relations en EJB3.
Héritage. Synchronisation avec la base de données, détachement/attachement.

Développement JEE avec JOnAS

Durée: 5 jours

Prix et dates: nous consulter

Public:

Développeurs, chefs de projet et intégrateurs d'applications en modèle JEE.

Objectifs:

Connaître l'architecture et les principes du serveur d'EJB JOnAS. Savoir concevoir une application avec le modèle JEE. Savoir programmer et utiliser les EJB.

Connaissances préalables nécessaires:

Il est demandé aux participants de maîtriser la programmation Java.

Programme:

- Le modèle JEE** : Les composants d'un serveur d'application Java : serveur HTTP, moteur de servlets, serveur d'EJB.
Terminologie: application web, sources de données, pilote JDBC.
- JOnAS** : Présentation des fonctionnalités du produit JOnAS
Support des spécifications JEE : EJB, JTA, JDBC, JMS, JMX, JNDI
Installation, configuration standard.
Composition d'une version personnalisée avec l'outil Ant, vérification avec JOnAS Check.
Les fichiers de configuration : jonas.properties, carol.properties, jonas-web.xml, jonas-ejb-jar.xml.
Architecture d'une application JEE avec JOnAS : intégration avec Apache, Tomcat, Jetty.
- Gestion des ressources** : Fournisseurs d'URL, fournisseurs JMS (JORAM), sessionMail.
Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Présentation de JDO. Mapping XML.
Connexion interapplicatives avec JCA.
- Servlets, pages JSP** : pages HTML dynamiques, communication avec bases de données et applications Java, communication avec d'autres servlets

Développement JEE avec JOnAS

- Développement avec les EJB** : Les EJB : spécifications (état actuel et limites).
EJB entité, EJB session, EJB Message-driven
fichiers de description en XML
Le générateur de classes d'interposition: GenIC
Jointures en CMP2. Relations entre EJB, présentation de EJB-OL.
JOnAS JTM: Gestion des transactions.
Sécurité: rôles, groupes de permissions.
Intégration dans Eclipse avec le plugin Jonas.
- Application répartie** : Le service de nommage
L'accès depuis Java : l'interface JNDI
- Déploiement** : Les différents composants d'une application : les fichiers application d'entreprise (.ear), les fichiers EJB (.jar), les fichiers application web (.war).
Déploiement dans JOnAS.
- Intégration** : Regroupement de composants unitaires en une application.
Création d'une application sur-mesure (créer un .ear, un .war, etc.).
- Exploitation** : La console d'administration JMX : jonas admin
Gestion des performances (surveillance mémoire, threads)
Analyse des fichiers de logs dans JOnAS et dans Tomcat pour les requêtes http.

Applications JEE avec JBoss

Durée: 5 jours
2250 € HT

du 25 au 29 Janvier
du 29 Mars au 2 Avril
du 7 au 11 Juin

du 30 Août au 3 Septembre
du 25 au 29 Octobre
du 6 au 10 Décembre

Public:

Développeurs et intégrateurs d'applications en modèle JEE.

Objectifs:

Connaître l'architecture et les principes du serveur JBoss. Savoir concevoir une application avec le modèle JEE, et la mettre en œuvre avec Jboss et Tomcat.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques Internet (serveur HTTP, HTML,...) et quelques notions de bases de données.

Programme:

- JBoss** : Présentation. L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Installation, configuration.
- Le modèle JEE** : Composants d'un serveur d'application JEE: serveur HTTP, moteur de servlets, conteneur d'EJB.
Positionnement des outils : Jboss, Tomcat, Apache, ...
Terminologie: application web, conteneur, sources de données, pilote JDBC, hôtes virtuels.
- Servlets, pages JSP** : pages HTML dynamiques, communication avec des bases de données, applications Java, inter-servlets
- Développement** : Composants d'une application d'entreprise: .ear, .jar, .rar, .war
Outils d'assemblage d'applications: ant, plugins, ...
Regroupement de composants unitaires en une application.
Création d'une application sur-mesure (créer un .ear, .war, ...).
- EJB** : Types d'EJB : EJB entité, EJB session, EJB Message-driven.
Persistance et transactions.
Annotations spécifiques EJB 3. Annotations et génériques.
Principes de mise en œuvre des EJB3.
Fichiers de description en XML. Les fichiers spécifiques: jboss.xml, jboss-web.xml, ...
Annotations en remplacement du descripteur de déploiement.
Simplification de la correspondance objet/relationnel.
Compatibilité ascendante. Les EJB3 sous JBoss.

Applications JEE avec JBoss

- EJB Session** : EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions. Dépendance d'injection. Intercepteurs de méthodes métier. Application: création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"
- EJB Entité** : Persistance avec les EJB. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes de recherche (query). Gestion des jointures en EJB3. Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY. Mapping par défaut. Utilisation de l'EntityManager. Relations entre EJB (one-one,one-many,many-many). Le langage EJB-QL. Héritage. Synchronisation avec la base de données, détachement/attachement. Correspondance objet/relationnel sous JBoss: jbosscmp-jdbc.xml. Application: création d'une application utilisant des jointures Many/Many
- EJB MessageDriven** : Développement de composants orientés message. Principe des MOM (Middlewares Orientés Message). Présentation de JMS. Développement d'un EJB Message Driven (MDB). L'interface JMS MessageListener. Traitement asynchrone. La méthode onMessage(). Fichiers de déploiement. Cycle de vie des MDB. Gestion des erreurs.
- Transactions** : Les transactions distribuées: commit à deux phases. Utilisation de JTA, JTS. Transactions explicites ou gérées par container. Annotations de transaction. Gestion des exceptions dans un contexte transactionnel. Mise en œuvre d'une application EJB3 transactionnelle. Mise en évidence du retour arrière déclenché par le conteneur JBoss et de la propagation du contexte transactionnel.
- Sécurité** : JAAS: rôles, groupes de permissions. Mise en œuvre dans le conteneur d'EJB. Rôles et permissions sur les méthodes. Intégration de la sécurité dans le serveur d'applications. Mise en place d'une politique de sécurité lors du déploiement de l'application. Mise en œuvre d'une application en cluster.

Applications JEE avec Geronimo

Durée: 5 jours
2250 € HT

du 25 au 29 Janvier
du 29 Mars au 2 Avril
du 7 au 11 Juin

du 30 Août au 3 Septembre
du 25 au 29 Octobre
du 6 au 10 Décembre

Public:

Développeurs et intégrateurs d'applications en modèle JEE.

Objectifs:

Connaître l'architecture et les principes du serveur Geronimo. Savoir concevoir une application avec le modèle JEE, et la mettre en œuvre avec Geronimo et Tomcat.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques Internet (serveur HTTP, HTML,...) et quelques notions de bases de données.

Programme:

- Présentation** : Geronimo :le positionnement, la certification JEE, la licence Apache. La version Little-G
Architecture :
les liens avec les autres projets openSource : Jetty, Tomcat, Derby, Axis, ...
les GBeans, gestion du cycle de vie des GBeans : le noyau Geronimo
- Installation** : Vérification des prérequis, installation. Variables d'environnement. Modes de démarrage du serveur.
- Rappels sur le modèle JEE** : les composants d'un serveur d'application JEE : serveur HTTP, moteur de servlets, serveur d'EJB.
Terminologie: application web, container, sources de données, pilote JDBC, hôtes virtuels.
Visualisation dans la console Geronimo.
Déploiement d'une application simple. Recherche des contextes racines.
- Outils** : Les outils d'administration en mode commande et par la console.
Configuration de Services,
Mécanismes de connexion aux bases de données : JDBC, pool de connexions.
Configuration du service JMS

Applications JEE avec Geronimo

- Servlets, pages JSP** : pages HTML dynamiques, communication avec bases de données et applications Java, communication avec d'autres servlets
- Liens avec Tomcat et/ou Jetty** : Installation et modification des paramètres de base
Activation du service.
- Développement avec les EJB** : Types d'EJB : EJB entité, EJB session, EJB Message-driven.
Persistance et transactions.
Les annotations spécifiques EJB 3. Annotations et génériques.
Principes de mise en œuvre des EJB3.
Fichiers de description en XML. Les fichiers spécifiques: geronimo-application.xml, geronimo-web.xml, ...
Utilisation des annotations en remplacement du descripteur de déploiement. Simplification de la correspondance objet/relationnel.
Compatibilité ascendante. Déploiement des EJB3 sous Geronimo.
- EJB Session** : EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions.
Dépendance d'injection. Intercepteurs de méthodes métier.
Application: création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"
- EJB Entité** : Persistance avec les EJB. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes de recherche (query).
Gestion des jointures en EJB3.
Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY.
Mapping par défaut. Utilisation de l'EntityManager.
Relations entre EJB (one-one,one-many,many-many). Le langage EJB-OL
Héritage. Synchronisation avec la base de données, détachement/attachement.
Correspondance objet/relationnel sous JBoss: jbossCMP-jdbc.xml.
Application: création d'une application utilisant des jointures Many/Many
- EJB MessageDriven** : Développement de composants orientés message
Principe des MOM (Middlewares Orientés Message).
Présentation de JMS. Développement d'un EJB Message Driven (MDB).
L'interface JMS MessageListener. Traitement asynchrone. La méthode onMessage().
Fichiers de déploiement.
Cycle de vie des MDB. Gestion des erreurs.

Applications JEE avec Geronimo

- Transactions**
- : Les transactions distribuées: commit à deux phases.
Utilisation de JTA, JTS. Transactions explicites. Transactions gérées par container.
Présentation JTA et JTS. Annotations de transaction.
Gestion des exceptions dans un contexte transactionnel.
Mise en œuvre d'une application EJB3 transactionnelle. Mise en évidence du retour arrière déclenché par le conteneur Geronimo et de la propagation du contexte transactionnel.
- Sécurité**
- : Présentation de JAAS. Mise en œuvre dans le conteneur d'EJB.
Rôles et permissions sur les méthodes.
Intégration de la sécurité dans le serveur d'applications.
Sécurité. Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.

Conception JEE avec les Design Patterns

Durée: 4 jours
1680 € HT

du 8 au 11 Février
du 12 au 15 Avril
du 14 au 17 Juin

du 30 Août au 2 Septembre
du 25 au 28 Octobre
du 6 au 9 Décembre

Public:

Chefs de projets, architectes et ingénieurs concepteurs.

Objectifs:

Comprendre les patrons de conception (Design Patterns) et les apports de UML2. Savoir les mettre en œuvre dans une architecture technique de conception.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de programmation objets, et des architectures JEE et EJB

Programme:

- | | |
|--------------------------------------|---|
| Définitions | : Principes des solutions de conception cataloguées.
Méthodologie: définition des besoins techniques, des classes "types" du pattern, des collaborations entre classes. |
| Présentation | : Rappels d'architecture.
Le modèle MVC, les outils de mise en œuvre (framework, ...).
Les design patterns.
Description formelle. Choix et utilisation d'un patron.
Présentation des patrons de conception: origine, les 3 familles (création, structuration et comportement), autres patrons |
| Patrons de création | : Rendre un système indépendant de la façon dont les objets sont créés.
Etude des patrons: fabrique abstraite, constructeur, prototype, singleton |
| Patrons de structuration | : Composer des objets pour obtenir de nouvelles fonctionnalités
Identifier des structures de classes abstraites et évolutives
Etude de: adaptateur, pont, composite, décorateur, façade, proxy |
| Patrons de gestion des comportements | : Identifier des coopérations évolutives.
Etude de: chaîne de responsabilités, commande, interpréteur, itérateur, médiateur, memento, observateur, état, stratégie, patron de méthode, visiteur. |
| Communication | : Coopérations découplées utilisables dans un système distribué
Etude de: publication/souscription, proxy |

Conception JEE avec les Design Patterns

Mise en œuvre

: Modélisation des patrons de conception et intégration à un diagramme de classes UML.
Création d'une application JEE en utilisant les patterns de Sun : SessionFacade, BusinessDelegate, serviceLocator, ..

Administration WebSphere

Durée: 5 jours
2250 € HT

du 8 au 12 Février
du 12 au 16 Avril
du 14 au 18 Juin

du 30 Août au 3 Septembre
du 25 au 29 Octobre
du 13 au 17 Décembre

Public:

Techniciens d'exploitation, administrateurs WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere. Savoir installer, configurer et exploiter des applications sous WebSphere. Les travaux pratiques sont réalisés avec la version 6 et 7.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et quelques notions de base sur les bases de données.

Programme:

- WebSphere** : Présentation des fonctionnalités du produit WebSphere. Rappels sur la terminologie et les objets WebSphere Application Server. Architecture. Topologie WebSphere.
- Installation** : Le produit WebSphere Application Server, Base de données, IBM-HTTP server, jdk. Installation manuelle. Installation automatique.
- Administration du système** : Fonctionnement de la console
Les objets à administrer : arborescence, groupes de serveurs, nœuds, applications d'entreprises.
Paramètres de configuration : la base de données de configuration, les fichiers XML.
- Intégration** : Utilisation de l'outil d'assemblage d'applications (WRD).
Le service de nommage: l'interface JNDI.
Gestion des hôtes virtuels
Variables WebSphere.
Domaines de réplication
Création de profils serveurs.
- Ressources** : Sources de données. Exemple: mise en œuvre avec PostgreSQL. Modification de la base de données cible.
Fournisseurs d'URL, fournisseurs JMS, les sessions JavaMail.
Connexion interapplicatives avec JCA.
Gestion du pool de connexions sous WebSphere Application Server.

Administration WebSphere

- Sécurité** : Les niveaux de sécurité dans WebSphere (Système, ressources, les rôles, etc ...). Définition des rôles. Authentification ldap. Configurer la sécurité du serveur
- Outils d'administration** : Console d'administration.
scripts prédéfinis
Outils de migrations, collector.
Administration à distance, automatisation.
wsadmin: présentation, objets supportés: \$AdminApp, \$AdminConfig, \$AdminControl, \$AdminTask
JACL: Syntaxe de base, contrôle de programme, exemples.
Jython: Syntaxe de base, contrôle de programme, exemples.
- Surveillance** : Informations émises par WebSphere, les exceptions, les messages de la console d'administration, les fichiers de traces.
Journalisation.Journal de la JVM.Vérification de la configuration.
Résolution des incidents
- Gestion des performances** : Etude du Ressource Analyser. Exploitation de l'advisor.
- Multi serveurs** : Déclaration de plusieurs serveurs sur la même machine dans le même profil.
Routeur HTTP IBM: Configuration
Mise en place d'un cluster avec affinités de Session sans gestionnaire de déploiement

Administration Tomcat

Durée: 2 jours
890 € HT

les 28 et 29 Janvier
les 18 et 19 Mars
les 10 et 11 Mai

les 5 et 6 Juillet
les 16 et 17 Septembre
les 22 et 23 Novembre

Public:

Toute personne souhaitant configurer, et administrer une application avec Tomcat.

Objectifs:

Comprendre le fonctionnement de Tomcat, et savoir le mettre en œuvre, l'installer, le configurer et l'administrer, optimiser le fonctionnement du serveur.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases tcp/ip, http, java (jsp, servlets)

Programme:

- Concepts de base : Les distributions Tomcat
Utilisation de Tomcat avec le serveur Web Apache
Servlets, JSP et composants,
Architecture d'un site Java
- Installation : Installation de Tomcat et modification des paramètres de base
- Configuration : Introduction
Le fichier server.xml
Fonctions avancées
Compilation des pages jsp avec jasper
- Déploiement d'applications web : Descripteurs XML
les fichiers .war
Déploiement à chaud
Tomcat Manager
- Connecteurs : La balise connector
Le connecteur HTTP, fonctionnement Tomcat en standalone
- Sécurité : Sécurisation et permissions. Les domaines, les rôles.
stockage informations de sécurité : JDBC, Datasource, JNDI,
JAAS, mémoire
Security Manager
- Performances et tests : gestion de la charge : load-balancing
Mise en place de clusters
- Exploitation : Intégration JMX

Exploitation JBoss

Durée: 4 jours
1860 € HT

du 11 au 14 Janvier
du 1er au 4 Mars
du 25 au 28 Mai
du 5 au 8 Juillet

du 6 au 9 Septembre
du 2 au 5 Novembre
du 13 au 16 Décembre

Public:

Exploitants, administrateurs d'applications JEE fonctionnant avec JBoss.

Objectifs:

Savoir installer et configurer JBoss. Savoir intégrer une application JEE, en assurer la mise en production, l'exploitation.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et quelques notions de base sur les bases de données.

Programme:

- JBoss**
 - : Présentation
 - L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Les outils JBoss: Hibernate, AOP, cache IDE.
 - Présentation du noyau JBoss. Présentation du micro-conteneur.
 - Installation, configuration, arborescence des fichiers.
 - Les services disponibles. Adaptation de la configuration : ajout/suppression de services
 - Visualisation dans la console JMX de JBoss.
- Applicatif**
 - : Terminologie: application web, container, sources de données, pilote JDBC, hôtes virtuels.
 - Configuration des services essentiels : JBossWeb, JNDI, JBossMQ, ...
 - Configuration du scanner de déploiement d'applications.
- Lien avec Tomcat**
 - : Installation de Tomcat et modification des paramètres de base
 - Activation du service Tomcat dans la configuration Jboss. Le fichier server.xml. Administration du service web.
 - Mise en place d'hôtes virtuels.
- Gestion des ressources**
 - : Fournisseurs d'URL, fournisseurs JMS, sessionMail.
 - Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Modification de la base de données cible. Exemple avec postgresql.
 - Connexion interapplicatives avec JCA.

Exploitation JBoss

- Sécurité avec JAAS et SecurityManager** : Intégration de la sécurité dans le serveur d'applications. Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.
- Automatisation** : Exploitation par scripts avec twiddle. Recherche de MBeans. Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, connexions JDBC. Scripts locaux avec bshDeployer.
- Intégration** : Configuration de JBoss en multi-instances sur un même serveur. Accès depuis un serveur apache, mise en place d'une liaison ajp à l'aide de mod_jk. Ecriture d'un service .sar JBoss permettant de simuler une application batch.
- Suivi** : Gestion des performances. Utilisation du MBean SNMP. Envoi de trap SNMP à une console de supervision. Fichiers de logs, résolution d'incidents. Configuration de Log4j. Modification des niveaux de journalisation. Automatisation d'alertes en cas d'incident.

Administration WebSphere avancée

Durée: 4 jours

Prix et dates: nous consulter

Public:

Techniciens d'exploitation, administrateurs WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere. Savoir administrer des applications sous WebSphere, et expertiser une application sous WebSphere 6.1 et 7.0 Network Deployment.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et de WebSphere Application Server.

Programme:

- Network Deployment** : Gestion des nœuds
Installation du gestion de nœuds. Installation d'un nœud piloté.
Transformation d'un serveur autonome en nœud piloté.
Fonctionnement des agents locaux. Gestion des serveurs.
- Cluster** : Gestion de cluster: ajout de nœud, visualisation.
Gestion des sessions répliquées. Mise en évidence des reprises de session. Déploiement d'applications sur la cellule. Déploiement séquentiel.
- Automatisation** : En ligne de commande, en mode script.
scripts avancés de gestion des nœuds
Utilisation de Jython. Positionnement par rapport à JAAS. Les apports de Python.
Les fonctions, modules, traitement des paramètres.
wsadmin: Présentation, objets supportés: AdminApp, AdminConfig, AdminControl, AdminTask
Appel des API Java , Jython et XML
- Redirecteur HTTP** : Routeur HTTP IBM: Configuration
Mise en place des affinités de Session.
Chiffrement SSL en amont et en aval du plugin. Liaison avec un serveur Apache interne en SSL.

Administration WebSphere avancée

- Sécurisation des accès** : Mise en place d'un serveur LDap en tant que référentiel utilisateurs. Chiffrage.
Mise en place d'un fournisseur externe d'authentification.
Utilisation d'un JACC Tivoli Access Manager
- Gestion des performances** : Suivi des performances à l'aide de scripts: récupération des compteurs WebSphere depuis un script Jython.
Relevés automatiques avec déclenchement d'alertes.
Mesure des consommations mémoires, des threads actifs. Mise en œuvre d'un système de surveillance d'un pool de connexion JDBC.
- Extensions** : Présentation de la version WebSphere 7.0 Network Deployment.
Présentation de la version Extended Deployment: gestion répartie de l'état des services, gestion des charges, planification des travaux.

Clusters JBoss

Durée: 3 jours
1430 € HT

du 15 au 17 Février
du 19 au 21 Avril
du 21 au 23 Juin

du 13 au 15 Septembre
du 8 au 10 Novembre

Public:

Architectes, administrateurs d'applications JEE souhaitant configurer, optimiser des clusters JBoss.

Objectifs:

Comprendre le fonctionnement d'une grappe de serveurs JBoss, savoir les configurer et les administrer. Comprendre les impacts sur le développement d'applications JEE en cluster. Les TP seront réalisés sur une version 4.2.3 et 5.1 de JBoss.

Connaissances préalables nécessaires:

Une bonne connaissance de Jboss est nécessaire et des environnements applicatifs JEE.

Programme:

- JBoss** : Rappels sur l'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Les outils JBoss: Hibernate, AOP, cache IDE, JGroups.
Installation, configuration.
- Cluster** : Fonctionnement en cluster ou en grappes de serveurs.
Configuration d'un cluster.
Définitions: nœud, partition, serveur, application.
- Configuration** : Présentation du projet JGroups intégré dans JBoss. Création d'une grappe autour d'une partition. Séparation de la grappe en multi-partitions.
Modifications des paramètres réseaux. Séparation de la grappe en plusieurs domaines multicast . Configuration du mode de lancement en ligne de commande et à l'aide de fichier de configuration.
Gestion de versions différentes de JBoss sur une même grappe.
Problème des cast.
- Applications** : Déploiement d'application dans une grappe. Modes de fonctionnement réparti ou singleton. Mise à jour des applications.
Réplication de sessions HTTP. Réplication d'EJB. HA-JNDI. Mise en évidence de la disponibilité du service de nommage.

Clusters JBoss

- Routage HTTP** : Distribution de sessions HTTP. Routage de niveau 7. Les outils. Mise en place de pound, mod_jk et haproxy. Mise en évidence de la répartition de charge.
- Architecture** : Mise en place d'un référentiel central d'applications. Architecture d'exploitation pour les applications. Intégration des fichiers journaux. Regroupement. Configuration de Log4j. Liaison d'une grappe avec une base de données répliquée.
- Automatisation** : Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, état des connexions JDBC. Ecriture d'un script donnant une vue globale des applications d'une grappe: état, nom, charge.

Struts

Durée: 3 jours
1360 € HT

du 15 au 17 Février
du 19 au 21 Avril
du 21 au 23 Juin

du 25 au 27 Août
du 11 au 13 Octobre
du 1er au 3 Décembre

Public:

Développeurs Java.

Objectifs:

Comprendre le fonctionnement de Struts, savoir développer une application en utilisant struts, connaître les possibilités d'interfaçage avec d'autres technologies Java.

Connaissances préalables nécessaires:

Une bonne connaissance de Java est nécessaire, et plus précisément des technologies jsp, et servlets.

Programme:

Le standard JEE

: L'architecture MVC : Model View Controller.
Technologies Java, XML
Rappel sur la méthode de construction d'une application avec les pages jsp et les servlets.

Présentation Struts

: Historique. Le projet Jakarta. Objectif de struts. Le principe, les outils disponibles.

Architecture

: Le modèle MVC: les objets métiers, les objets IHM.
Le rôle des différents composants: ActionServlet, pages JSP, objets ActionForm, Action, et ActionServlet.
Les javabeans de stockage. La persistance : différentes solutions.
La vue: HTML dynamique avec Struts.
les formulaires dynamiques;
affichage de beans avec struts, tests (empty, equal, greaterEqual).
Le contrôleur: principe, la méthode execute.
Exemple de servlet Action.
Les actions standards : ForwardAction, DispatchAction

Mise en œuvre

: Installation de struts.
Assemblage d'une application, les fichiers web.xml et struts-config.xml, les fichiers .TLD
Travaux pratiques d'intégration avec Tomcat, en environnement Eclipse.

Struts

- Extensions** : Struts Validator: intérêt, principe, les fichiers XML, la vue, l'assemblage. Mise en œuvre sur un exemple.
DynaActionForms: implémentation, validation.
Struts Templates: objectif, principe, la balise include, les pages templates, les balises tiles (tiles:insert, tiles:getAsString)
Développement d'un exemple: création d'une page générique avec les templates.
- Exploitation** : les classes de gestion d'erreur, traces : exemples avec log4J, performances : mesures avec Junit, les messages utilisateurs : les taglibs associées, automatisation : génération des fichiers avec Ant : présentation de l'outil, gestion des dépendances, exemples.
- Evolutions** : Présentation de la technologie : Java Server Faces
Le framework Struts Shale Framework
Le projet Apache MyFaces

Développement Web avec JSF

Durée: 3 jours
1360 € HT

du 15 au 17 Février
du 19 au 21 Avril
du 21 au 23 Juin

du 25 au 27 Août
du 11 au 13 Octobre
du 1er au 3 Décembre

Public:

Développeurs Java.

Objectifs:

Comprendre le mécanisme des Java Server Faces, savoir les mettre en œuvre pour développer une application Web.

Connaissances préalables nécessaires:

Une bonne connaissance de Java est nécessaire, et plus précisément des technologies jsp, et servlets.

Programme:

Le standard JEE

: L'architecture MVC : Model View Controller.
Rappel sur la méthode de construction d'une application avec les pages jsp et les servlets.

Présentation JSF

: Les apports de JSF par rapport à Struts
Les différentes implémentations :
la référence Sun, Apache MyFaces, Smile, ECruiser
Architecture d'une application

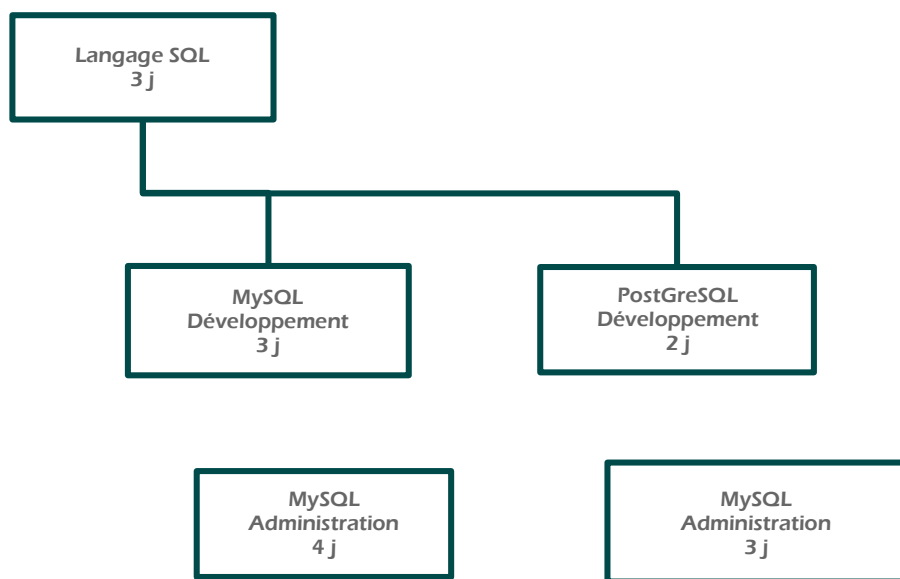
Développement avec les JSF:

Le cycle de vie d'une page JavaServer Faces
Les balises
Les composants JavaBeans : backing beans
Gestion des messages, des événements
Les fichiers de configuration, descripteurs de déploiement
Définition des règles de navigation
Travaux pratiques avec Apache MyFaces et Tomcat

Intégration

: Lien avec Struts Shale Framework.
Utilisation des pages JSF avec AJAX

Filières Bases de Données



Langage SQL

Durée: 3 jours

Public:

Analystes, développeurs, utilisateurs. Exploitants, administrateurs de bases de données.

Objectifs:

Apprendre à créer des applications avec une base de données MySQL 5.0

Connaissances préalables nécessaires:

Concepts relationnels

Programme:

- Introduction** : Rappel sur le modèle relationnel
Les composantes de SQL
Les tables
La norme SQL
- La requête SELECT** : Syntaxe générale. Projection de colonnes. Sélection de lignes.
Les différents types de prédicats.
Les expressions. Les fonctions. L'agrégation. Le tri.
Les tables temporaires .
- La jointure** : Syntaxe.
Inner join. Outer join .
- Les requêtes de mise à jour** : INSERT, UPDATE, DELETE.
Notion de transaction : COMMIT, ROLLBACK
- Les requêtes imbriquées** : Le Subselect simple
Le Subselect corrélé
Les opérateurs ANY, SOME, ALL, EXISTS
- Opérateur ensembliste** : Union de Select
- La définition des objets :
Data Definition Language** : CREATE TABLE
CREATE INDEX
ALTER et DROP
L'intégrité référentielle
- Les VUES** : Création et utilisation
Mises à jour

MySQL : Développement

Durée: 3 jours
1340 € HT

du 1^{er} au 3 Février
du 6 au 8 Avril
du 31 Mai au 2 Juin

du 6 au 8 Septembre
du 8 au 10 Novembre

Public:

Développeurs et toute personne souhaitant mettre en place une application avec une base de données MySQL

Objectifs:

Apprendre à créer des applications avec une base de données MySQL 5.0

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

- Introduction** : Présentation, historique, les versions MySQL (standard, Max, Pro, Classic), les licences (GPL et commerciale).
Les composants
Installation outils clients
- Clients MySQL** : Présentation de Query Browser
Construction de requêtes
Tests, comparaison et debugging et optimisation de requêtes
- Procédures de chargement de données** : Format des données à importer
Les commandes disponibles : load, insert,
Développement de scripts d'importation
- DDL et DML** : Les spécificités de MySQL dans le mise en œuvre des requêtes SQL.
- Programmation serveur** : L'intérêt de la programmation côté serveur, et les limites
Mise en œuvre de déclencheurs (triggers)
et programmation de procédures stockées
Déclaration et utilisation de variables
Gestion des curseurs
Evolutions prévues dans les futures versions de MySQL.

MySQL : Administration

Durée: 4 jours
1780 € HT

du 8 au 11 Février
du 12 au 15 Avril
du 7 au 10 Juin

du 30 Août au 2 Septembre
du 4 au 7 Octobre
du 15 au 18 Novembre

Public:

Toute personne souhaitant configurer, installer et administrer une base de données MySQL

Objectifs:

Comprendre le fonctionnement, et savoir installer, configurer et administrer une base de données MySQL (le cours est réalisé sur une version 5 de MySQL). (le cours est réalisé sur une version 5 de MySQL).

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

- Introduction** : Présentation, historique, les versions MySQL (standard, Max, Pro, Classic), les licences (GPL et commerciale).
Les composants du serveur MySQL
Caractéristiques: transactions, clusters.
- Installation** : Choix du produit à installer : les RPMs, le code compilé ou les sources. Installation, configuration.
Les scripts fournis avec MySQL : démarrage du serveur, création des tables de droits d'accès, démarrage de multi-serveurs, ...
Outils graphiques
- Sécurité** : Système des privilèges : principe de fonctionnement, authentification, contrôle des droits pour les requêtes
Gestion des comptes utilisateurs : création/suppression de comptes, limitation des ressources, sécurisation des accès
Mise en place de SSL.
- Les fichiers de logs** : Les erreurs, les modifications du fichier ISAM, les requêtes
- Optimisation** : Les requêtes en cache.
Les différents points à optimiser : les requêtes, les tables, la configuration du serveur, et de l'environnement.
- Sauvegardes** : Les tables MyISAM et InnoDB
Utilitaire myisamchk : contrôler, réparer, optimiser
Vérification sur base à l'arrêt. Réparation. Vérification/réparation en cours d'exploitation.
Méthode de sauvegarde des données MySQL, script mysqldump, ou mysqlhotcopy

MySQL : Administration

Haute Disponibilité : Mise en œuvre de la réplication

PostgreSQL : Développement

Durée: 2 jours
880 € HT

les 18 et 19 Janvier
les 8 et 9 Mars
les 3 et 4 Mai

les 5 et 6 Juillet
les 18 et 19 Octobre
les 6 et 7 Décembre

Public:

Développeurs souhaitant mettre en place une application avec PostgreSQL

Objectifs:

Apprendre à créer des applications avec une base de données PostgreSQL

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

- Présentation de PostgreSQL :** PostgreSQL : serveur de base de données relationnelle objet
Historique, licence, mode de développement
Les composants du serveur PostgreSQL: serveur, client, connecteurs jdbc, tcl, pl, python
Les apports de la version 8.3
- Installation et configuration :** Installation de PostgreSQL
Installation avec un système de paquetage
- Interagir avec le serveur :** Le concept du client serveur
Le client PostgreSQL
Utiliser les clients graphiques
- Spécificités SQL de PostgreSQL :** Spécificités DDL (Définition des objets)
Spécificités DML (Manipulation des données)
Optimisation des requêtes
- Programmation coté serveur :** Importation et exportation de données
L'intérêt de la programmation côté serveur et ses limites
Mise en œuvre de déclencheurs (triggers) et programmation de procédures stockées
Déclaration et utilisation de variables
Gestion des curseurs
Mise en œuvre avec PL/pgsql et présentation des autres outils disponibles

PostgreSQL : Administration

Durée: 3 jours
1300 € HT

du 20 au 22 Janvier
du 10 au 12 Mars
du 6 au 8 Mai

du 7 au 9 Juillet
du 20 au 22 Octobre
du 8 au 10 Décembre

Public:

Toute personne souhaitant configurer, installer et exploiter une base de données PostgreSQL

Objectifs:

Comprendre le fonctionnement, et savoir installer, configurer et exploitation une base de données PostgreSQL.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

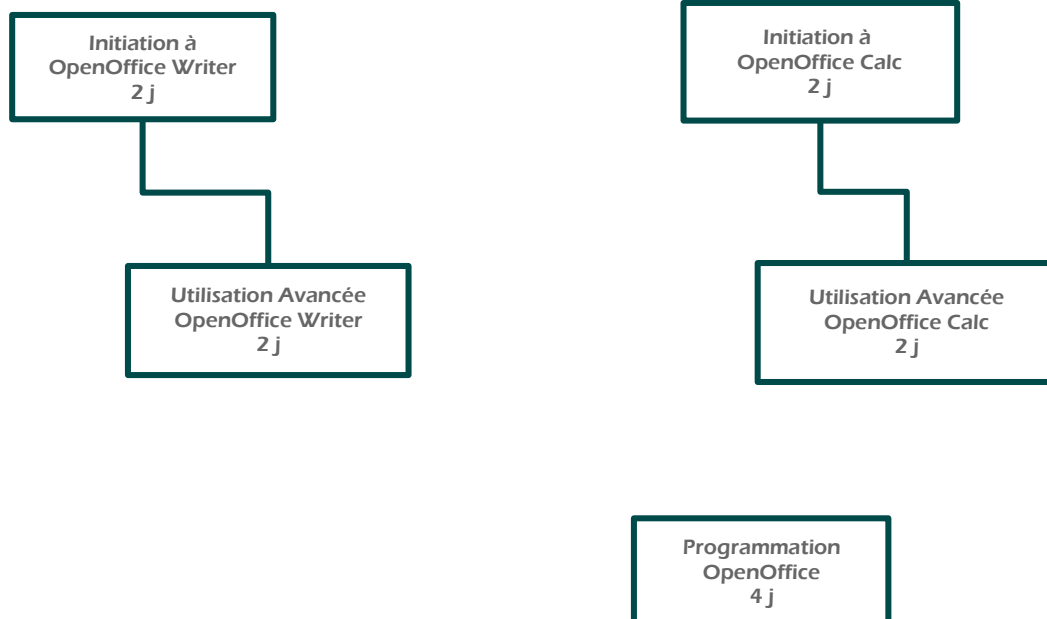
Programme:

- Introduction** : Présentation, historique, les versions PostgreSQL, les outils complémentaires et les licences.
Les composants du serveur PostgreSQL: serveur, client, connecteurs jdbc, tcl, pl, python
- Installation** : Choix du produit à installer : les RPMs, le code compilé ou les sources.
Installation. Configuration, organisation du répertoire /var/lib/pgsql, fichier postgresql.conf.
Les scripts fournis avec PostgreSQL : démarrage du serveur, création des tables de droits d'accès, démarrage de multi-serveurs, ...
- Sécurité** : Système des privilèges : principe de fonctionnement, authentification, contrôle des droits pour les requetes
Gestion des comptes utilisateurs : création/suppression de comptes, limitation des ressources, sécurisation des accès: fichier pg_hba.conf
Mapping avec les utilisateurs systèmes: pg_ident.conf
- Utilisation** : Commande psql. Accès aux tables. Les commandes en \ pgadmin: installation, configuration.

PostgreSQL : Administration

Exploitation : Sauvegardes/Sauvegardes
contrôler l'état de la table, réparer, optimiser:
pg_dump, pg_dumpall, vacuumdb
Méthode de sauvegarde des données PostgreSQL
Montée de niveaux:
Copie de serveur à serveur, réplication.
Tests de régression

Filières bureautiques



Utilisation Avancée OpenOffice Calc

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser le tableur OoCalc.

Objectifs:

Savoir maîtriser les fonctions du tableur OO Calc.

Connaissances préalables nécessaires:

Connaissance d'un tableur, de préférence OO Calc.

Programme:

- | | |
|--------------------------|---|
| Fonctions | : Présentation des différentes fonctions disponibles :
Classeur, Information, Mathématiques, Financières |
| Graphiques | : Définition d'un graphique :
zone de données, échelle, étiquettes, titre, légende
Graphique en 3D. |
| Listes de données | : Utilisation du filtre standard, AutoFiltre.
Pilotes de données. |
| Styles | : Créer, appliquer un style, modèle de classeur |
| Exploitation | : Protection d'un classeur, des cellules.
Gestion des versions.
Importations/Exportations:
format MS-Excel, ASCII, CVS, SYLK.
Options d'importations.
Les option spécifiques du tableur
Macros instructions simples |

Programmation OpenOffice

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant Programmer en OOBASIC, ou transformer des macros Visual Basic pour la suite OpenOffice.

Objectifs:

Maîtriser la programmation avec OO Basic.

Connaissances préalables nécessaires:

Les notions de programmation sont indispensables, ainsi qu'une bonne connaissance du tableur Calc et du traitement de texte Writer

Programme:

- OOBasic** : Présentation. Caractéristiques : OOBASIC Langage Objet , classification par collections.
Structures de données. Services : ouverture et enregistrement des fichiers, les filtres texte communs et les documents vierges
- Classeurs** : Ouverture, fermeture, sauvegarde de classeurs vierges et existants.
- Feuilles** : Accès nommé et indexé, ajout, recopie, déplacement et suppression de Feuilles. Activation, protection et renommage.
- Cellules** : Accès, Ajout, recopie, déplacement et suppression de pages de cellules nommées.
Accès aux cellules par position, adresses des cellules. Propriétés des cellules, contenu des cellules. Nombres et formules, Textes et Curseurs. Formatage, formats numériques
- Les tableaux** : Copie ou déplacement en bloc. Propriété "DataArray". Transfert d'une cellule dans un bloc (valeur et formule).
Tri et Descripteur du tri.
Filtrage et descripteur d'extraction. Totalisations et Descripteur de totalisation.
Recherche et remplacement de caractères : les descripteurs associés.
- Fonctions utilitaires** : Effacer le contenu des cellules. Insérer ou supprimer des cellules, des lignes ou des colonnes.
Autres Fonctions (mathématiques, financières).
Le texte et les curseurs.

Programmation OpenOffice

- Interface de saisie** : Boites de Dialogues et l'EDI, les contrôles.
Exemple de gestion des événements.
Configuration par programme. Imprimer, sélection multiple.
Configuration Calc
- Utiliser la documentation** : XRAY appliqué à Calc, recherche de propriétés et de méthodes
SDK : les informations disponibles

Initiation à OpenOffice Writer

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser le traitement de texte OOWriter.

Objectifs:

Savoir utiliser les fonctions de base du traitement de texte.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

- | | |
|------------------------------------|---|
| Bases | : Lancement, gestion des barres d'outils, les différents modes d'affichage.
Utilisation de l'aide.
Ouverture/création/enregistrement de fichiers.
Saisie et modifications de texte : insertion, copie, annulation, caractères spéciaux, recherche et remplacement de texte |
| Mise en forme | : caractères: polices, effets de caractères, ...
paragraphes: alignement, enchainement, retraits, écarts, bordures
pages : en-têtes, pieds de pages, ... |
| Styles | : Les styles prédéfinis, le styliste
Le catalogue
Création de nouveaux styles |
| Quelques fonctions avancées | : Les tableaux : insertion, manipulation des colonnes et lignes, scinder/fusionner les tableaux.
Vérificateur d'orthographe.
Insertion de champs particuliers: date, heure, numéros de page, caractères spéciaux |

Utilisation Avancée OpenOfficeWriter

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant approfondir ses connaissances sur le traitement de texte OpenOffice Writer.

Objectifs:

Savoir utiliser les fonctions avancées du traitement de texte.

Connaissances préalables nécessaires:

Savoir utiliser un traitement de texte, de préférence, OOWriter.

Programme:

- Gestion de documents** : Modèles : création, catalogue
Document maître : insertion, déplacer, actualiser
Numérotation des titres, des chapitres
Table des matières et Index
Les liens hypertextes
Notes de bas de pages, repères de texte.
Gestion des versions de document.
- Outils** : Vérification orthographique, Coupure de mots,
AutoCorrection, AutoFormat
Dictionnaire des synonymes
- Calculs et graphiques** : Valeurs numériques, formules
Insertion/modification de graphiques
- Dessins** : Outils de dessin, insertion images, sons
Gestion des cadres et des objets : ancrage, habillage,
contour, ...
- Mailing** : Principe, base de données, source de données, impression
- Exploitation** : Importations/Exportations: formats Word, HTML. Exportations
PDF.
Personnalisation. Macros instructions.

Initiation à OpenOffice Calc

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser un tableur et désirant connaître les grands principes du tableur OpenOffice pour Linux.

Objectifs:

Savoir utiliser le tableur OpenOfficeCalc pour les opérations courantes.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

- Bases** :
- Lancement, gestion des barres d'outils.
 - Ouverture/création/sauvegarde de fichiers.
 - Utilisation de l'aide.
 - Définitions: classeur, feuille de calcul, cellules.
 - Déplacement dans une feuille : recherche d'une cellule, ...
 - Manipulations sur différentes feuilles : déplacement,
 - Affichage : partiel, plein écran, masquer une barre d'outil, ...
 - Sélection : des cellules, des lignes, des colonnes
- Saisie** :
- Le contenu d'une cellule : texte, valeurs, formules
 - Création de séries de données, saisie semi-automatique
 - Effacer, supprimer, copier, trier des données.
 - Nommer les zones : sélection de plages de cellules, accès par le nom de la zone.
- Calculs simples** :
- Syntaxe d'une formule. Références relatives/absolues
 - Utilisation des assistants pour créer des formules.
 - Somme, moyenne, formules statistiques de base
- Mise en forme** :
- Adapter les largeurs de colonnes, hauteurs de lignes, masquer des lignes et colonnes,
 - Format des valeurs numériques, dates et heures, alignement, etc ...
- Impression** :
- Mise en page, options.

Bulletin d'inscription

Intitulé du stage :

Dates :

Coordonnées du responsable formation :

Nom :

Prénom :

Société :

Téléphone :

Adresse mail :

Adresse postale :

.....

Coordonnées des participants :

Nom :	Prénom :	Adresse mail :

Vous pouvez nous transmettre votre inscription :

Par mail : pdf@Pythagore-fd.fr

Par fax : au 01 55 33 52 11

Dès réception, nous vous enverrons une confirmation d'inscription, ainsi que les modalités pratiques concernant le stage.

Conditions de vente

Tous les prix indiqués dans ce catalogue sont hors taxes.

Toute inscription à une formation implique l'adhésion des présentes conditions de vente. L'inscription est ferme à partir de la signature de la convention. Pythagore F.D. se réserve la possibilité d'annuler une session si le nombre de stagiaires est insuffisant. Pythagore F.D. informera le client au minimum dix jours ouvrables avant le début du stage. Le client peut alors reporter ou annuler son inscription.

Annulations

Toute formation commencée est intégralement due. Si l'annulation d'une inscription à un stage inter-entreprise, est faite dans la période allant du 10^e jour ouvré au 2^e jour ouvré avant le début du stage, 50% du montant des frais de formation sont dus. Si l'annulation n'a pas été faite 48h avant le début de la formation, la totalité du montant des frais de formation reste due. Dans le cas de l'annulation d'une formation ou d'un cycle de formation en intra moins de dix jours ouvrables avant le début de la formation, 50% des frais de formation restent dus.

Déroulement du stage

Tout stagiaire doit se conformer au règlement du centre de formation de Pythagore F.D. A défaut, le stagiaire pourra être exclus de la formation. Les frais de formation sont alors intégralement dus. Pendant la formation, les stagiaires restent les employés du client qui est responsable de leurs faits et gestes en application de l'article 1384 du code civil.

Paielement

Les factures sont payables, sans escompte, à réception pour les stages en inter-entreprises. Les cycles de formation d'une durée supérieure à un mois font l'objet d'une facturation mensuelle. En cas de non-paiement à son échéance, toute somme due portera intérêt de plein droit au taux d'une fois et demie le taux d'intérêt légal.

Litiges

Tout litige qui ne pourra être réglé à l'amiable sera du ressort du tribunal de Commerce de Paris.

Pythagore F.D. - 11, rue du Faubourg Poissonnière 75009 PARIS
Tél : 01 55 33 52 10 - Fax : 01 55 33 52 11
S.A.S au capital de 40 000 Euros - RCS Paris B 398 145 474

**Pour toute information, appelez-nous au
01 55 33 52 10
www.Pythagore-fd.fr**

Table des matières

Catalogue des formations 2010		1
Pythagore F.D. : Apprendre à Apprendre		
Pythagore F.D. : Apprendre à Apprendre		3
Filières Systèmes		
Présentation des systèmes d'exploitation	SY001	5
Systèmes temps réel	SY002	6
Virtualisation avec Xen	SY004	7
Introduction à Unix	UX001	8
Le Shell	UX002	9
Shell avancé	UX004	10
Scripts Shell pour administrateurs	UX005	11
Administration UNIX	UX011	12
Administration AIX	UX018	14
Administration AIX avancée	UX019	16
Administration Solaris 9 et 10	UX040	18
Administration avancée Solaris 9 et 10	UX041	20
MAC OS X Server : Prise en main	MC010	21
MAC OS X Server : Installation et configuration	MC012	23
Linux utilisation	UX100	25
Linux, poste de travail	UX106	26
Introduction au logiciel libre	UX107	27
Administration Linux	UX111	28
Administration avancée Linux	UX140	30
Les services réseaux Linux	UX112	32
Linux système sécurisé	UX117	34
Linux sécurité des accès	UX118	36
Haute disponibilité Linux	UX115	37
Linux : optimisation performances métrologie	UX119	39
Virtualisation Linux	UX123	40
Gestion de Parc avec OCS et GLPI	UX124	41
Filières Systèmes embarqués et mobilité		
Linux temps réel et embarqué	UX122	44
RT Linux	UX121	46
openWrt	UX126	48
Linux mobile	UX127	50
Linux mobile avec Android	UX128	51
Developpement iPhone	MC020	53
Filières Réseaux et TCP/IP		
Introduction aux réseaux	RS001	56
Bus One-wire	RS125	58
TCP/IP : protocoles et mise en œuvre	RS003	59
Voix sur IP	RS023	61
Voip - Configuration Asterisk	RS024	62
Installation et Configuration WIFI	RS021	63
Administration réseaux IP	RS007	64
Routage IP avancé	RS009	66
IP: Migration vers IPv6	RS014	67

Performances et optimisation des réseaux	RS025	68
Sécurité TCP/IP	RS018	69
SNMP	RS022	70
Annuaire LDAP	RS122	71
Messagerie	RS006	72
Nagios Centreon : utilisation	RS129	73
Administration Nagios	RS130	74

Filières Internet et Web

Serveur WEB : apache	RS104	77
Serveur Proxy Squid	RS114	78
Sécurisation des services internet	RS115	79
HTML	RS102	80
Web Dynamique avec JavaScript	RS106	82
CMS : Présentation	RS126	83
CMS : mise en œuvre de Joomla	RS127	84
Présentation Web 2.0	RS128	85

Filières Langages

Initiation au langage C	LC001	87
Perfectionnement en langage C	LC002	88
Le langage C++	LC004	89
Programmation système en langage C	LC010	90
Langage C pour TCP/IP	RS016	91
Debugging en C sur Unix	UX015	92
Subversion utilisateur	LD002	93
Subversion Administrateur	LD003	94
Programmation PHP	LH001	96
Python	LY001	97
Perl	LP001	98
XML Introduction	AS120	99
Ruby On Rails	AS201	100
Développement Web avec Ajax	RS109	102

Filières Langages Java

Programmation Java	LJ001	104
Programmation avancée en Java	LJ002	106
Sécurité Java	LJ005	108
Outils Open Source Java	LJ010	109
Java Micro Edition	LJ100	111
JEE : les servlets et JSP	AS004	112
JEE : les EJB	AS005	114
Développement Web Services	AS006	116
Techniques et outils de migration vers les EJB3	AS007	117
Serveurs d'application JEE	AS011	118
Développement JEE avec WebSphere	AS101	120
Développement JEE avec JOnAS	AS103	122
Applications JEE avec JBoss	AS104	124
Applications JEE avec Geronimo	AS105	126
Conception JEE avec les Design Patterns	AS122	129
Administration WebSphere	AS131	131
Administration Tomcat	AS133	133
Exploitation JBoss	AS134	134

Administration WebSphere avancée	AS135	136
Clusters JBoss	AS144	138
Struts	AS150	140
Développement Web avec JSF	AS151	142

Filières Bases de Données

Langage SQL	BD001	145
MySQL : Développement	BD010	146
MySQL : Administration	BD011	147
PostgreSQL : Développement	BD020	149
PostgreSQL : Administration	BD021	150

Filières bureautiques

Utilisation Avancée OpenOffice Calc	OC002	153
Programmation OpenOffice	OP001	154
Initiation à OpenOffice Writer	OW001	156
Utilisation Avancée OpenOfficeWriter	OW002	157
Initiation à OpenOffice Calc	OC001	158

Bulletin d'inscription