

## Table des matières

AGILE ET SCRUM .....	2
Formation Agile : Product Owner.....	2
Formation Agile Scrum : Initiation.....	5
Formation Agile : Scrum Developer .....	6
Formation Agile Scrum : Approfondissement .....	8
Formation Préparation de la certification PSD (scrum.org) .....	10
Formation SAFe Scrum Master (SAFe SM) .....	11
BIG DATA.....	13
Formation Apache Spark.....	13
Formation Data Science (R et Hadoop).....	15
Formation Elasticsearch : Rechercher et analyser en temps réel .....	17
Formation Hadoop : Déployer du Big Data .....	19
JAVA SE.....	21
Formation Java SE Initiation .....	21
Formation Java SE Approfondissement.....	23
Formation Linux Administration Avancé LPI 201: Maintenance, Customisation, Réparation.....	24
Formation Linux Administration Avancé LPI 202: Services Web, Mails, DNS.....	26
Formation Linux Avancé :.....	28
Serveur d'entreprises avec Samba/LDAP/Netfilter/Squid.....	28
Formation Linux Expert: LDAP.....	30
PYTHON.....	32
Formation Python Initiation .....	32
Formation Python: Approfondissement.....	34
Formation Python Avancé : Fonctionnalités avancées, domaines d'applications.....	35
DEVOPS.....	38
Formation DevOps – Foundation - Certification.....	38

# AGILE ET SCRUM

## Formation Agile : Product Owner

Durée : 2 jours

Public : Toute personne devant jouer le rôle de Product Owner dans un projet agile Scrum

Pré-requis : Connaissances de la méthode agile Scrum

Objectifs : Clarifier le travail et les responsabilités du Product Owner - Maîtriser les techniques de rédaction et de priorisation des stories du product backlog

### Comprendre le rôle du Product Owner dans un projet Scrum

Le cadre Scrum / Guide

Cycle de vie d'un projet Scrum

Rôles définis par Scrum : Product Owner, Scrum Master, Team

Time boxes : Sprint planning, Sprint Review, Sprint Retrospective, Daily Scrum

Artéfacts : Product Backlog, Sprint Backlog, Burndown chart

Règles et principes clé de Scrum

Responsabilités et rôle du Product Owner

Activités du Product Owner dans le cycle de vie Scrum

Atelier : **Analyse du guide Scrum et passage du Open Assessment PSPO I**

### Partager la vision du produit

Vision du produit : définition, intérêt

Forme d'une vision : Lean Canvas

Toolbox

Exemple complet de Product Vision Board

Atelier : **Construction et analyse d'une vision de produit**

### Recueillir le besoin

Faire émerger le besoin

Techniques de recueil

Organisation des rôles des acteurs (RACI)

Formaliser le besoin : IEEE vs Use Cases vs User Stories

Ateliers :

- **Construction d'une matrice RACI pour organiser les rôles d'un projet. - conduite d'un entretien de recueil (technique des 9 cases)**

- **Animation d'un atelier de brainstorming et construction de mind maps.**

## Construire le produit

Construction d'une Roadmap

Des objectifs aux fonctionnalités : Impact mapping Identifier et structurer les fonctionnalités  
: Story mapping

Déduire la roadmap de la story map

Démarche de construction du product backlog

Rédaction des user stories : forme, spécification INVEST

Compléments d'une user stories : contraintes opérationnelles, maquettes,..

Ecriture des tests d'acceptation (méthode Behaviour Driven Development)

Outils d'automatisation des tests

Comprendre la dette technique : identification, catégorisation, priorisation

Ateliers :

- Construction **d'une impact map à partir de la vision du produit.**
- **Construction d'une story map, identification des MMF (Minimum Marketable Features).** - **Construction de la roadmap.**
- **Rédaction de user stories et ateliers three amigos pour les tests d'acceptation.**
- **Organisation de la construction du produit (Kanban du Ready)**
- **Démarche d'enrichissement et d'optimisation du produit.**

## Estimer et prioriser les stories

Techniques d'estimation : ideal days vs story points

Mise en oeuvre de la méthode des story points Fiabilisation de l'estimation

Facteurs d'hiérarchisation des besoins

Méthodes de priorisation : Kano vs MOSCOW vs Poids relatifs Priorisation par thèmes :

themes screening / scoring Gestion des risques

Regroupement et ordre final des stories dans le product backlog

Ateliers :

- **Poker planning game pour l'estimation des stories.**
- **Calcul de la priorité avec la méthode des poids relatifs.**
- **Remaniement du product backlog.**

## Connaître les outils d'un environnement agile Scrum

Conception :

- Architecture évolutive
- Travaux d'étude technique

Codage :

- Intégration continue
- Pilotage par les tests
- Programmation en binôme - Refactoring
- Tests d'acceptation

Maintenance :

- Gestion des bugs

**Atelier : Panorama de la toolbox logiciel d'un projet agile.**

## Etablir un contrat agile

Différentes formes d'un contrat agile

Mise en place d'un Plan d'Assurance Qualité (PAQ)

Contractualisation par Sprint

# Formation Agile Scrum : Initiation

Durée : 1 jour

Public : Toute personne devant être impliquée dans un projet Scrum : ScrumMaster, Product Owner, Scrum Developer, DSI

Pré-requis : Notions de gestion de projets  
Objectifs : Découvrir la gestion de projets agiles avec la méthode Scrum

Adhérer à l'agilité

Gestion de projet : prédictive vs agile Pourquoi l'agilité ? Le manifeste agile Panorama des méthodes agiles Démarche d'adoption de l'agilité

Apprendre les pratiques agiles

Le Lean Management : objectif, principes

Kanban : principe, avantage, cycle de vie d'une étiquette Pratiques XP (eXtreme Programming)

Maîtriser la méthode agile Scrum

Le cadre Scrum / Guide

Cycle de vie d'un projet Scrum

Rôles définis par Scrum : Product Owner, Scrum Master, Team

Time boxes : Sprint planning, Sprint Review, Sprint Retrospective, Daily Scrum

Artéfacts : Product Backlog, Sprint Backlog, Burndown chart

Règles et principes clé de Scrum Responsabilités / rôle

Choix de la taille du sprint Constitution de l'équipe

Faiblesses de la méthode Ateliers :- **Analyse du guide de Scrum et passage du Open Assessment PSM I - Estimation des stories (charge, complexité) et**

**planning poker- Animation de rétrospective : Speed boat, Turn the tables, ...**

# Formation Agile : Scrum Developer

**Durée :** 2 jours

**Public :** Développeurs amenés à travailler dans une équipe agile Scrum

**Pré-requis :** Connaissances du cadre Scrum

**Objectifs :** Définir l'action d'un développeur dans un projet agile - Apprendre les bonnes pratiques de développement et de tests dans un environnement agile Scrum.

## Maîtriser la méthode agile Scrum

Le cadre Scrum / Guide

Cycle de vie d'un projet Scrum

Rôles définis par Scrum : Product Owner, Scrum Master, Team

Time boxes : Sprint planning, Sprint Review, Sprint Retrospective, Daily Scrum Artéfacts :

Product Backlog, Sprint Backlog, Burndown chart

Règles et principes clé de Scrum

Responsabilités / rôle

Atelier : analyse du guide de Scrum et pratiques XP

## Analyser le produit

Analyse de la vision du produit

Compréhension de l'organisation des rôles (RACI)

Démarche de construction du product backlog

Rédaction des user stories : forme, spécification INVEST

Compléments d'une user stories : contraintes opérationnelles, maquettes,..

Ecriture des tests d'acceptation (méthode Behaviour Driven Development)

Outils d'automatisation des tests

Ateliers :

- **Construction d'une story map, identification des MMF (Minimum Marketable Features).**
- **Construction de la roadmap.**
- **Rédaction de user stories et atelier streamig pour les tests d'acceptation.**

## Estimer et prioriser les stories

Techniques d'estimation : ideal days vs story points

Mise en oeuvre de la méthode des story points Fiabilisation de l'estimation

Facteurs d'hiérarchisation des besoins

Méthodes de priorisation : Kano vs MOSCOW vs Poids relatifs Priorisation par thèmes :

themes screening / scoring Gestion des risques

Regroupement et ordre final des stories dans le product backlog

Ateliers :

- **Poker planning game pour l'estimation des stories.**
- **Calcul de la priorité avec la méthode des poids relatifs.**
- **Remaniement du product backlog.**

## Développer au sein d'une équipe agile

Pratiques XP (eXtreme Programming)

Daily Scrum meeting : démarche, intérêt Standards de développement Dette technique : classification, priorisation, planification

Architecture, refactoring, design patterns et Crosscutting concerns

Tests dans un environnement agile : unitaires, fonctionnels, intégration, ...

Techniques de tests : TDD, BDD, ATDD

Traitement des anomalies

Gestion du cycle de vie d'une application (ALM)

Comprendre les indicateurs de performances : Burn Down et Burn Up chart

Présentation du suivi de l'avancement : Kanban board, Parking Lots

Revue de sprint Rétrospective de sprint Ateliers :- **Animation d'une réunion de sprint planning,**

**découpage de stories en tâches. - Rédaction de tests et automatisation- Design patterns-**

**Revue de sprint : acteurs, démonstration, validation des stories.- L'art de la rétrospective. -**

**Utilisation d'outils de gestion de projets agiles.**

## Comprendre l'environnement de développement

Maîtriser un SCM (Git ou SVN) : fonctionnement, principales commandes

Intégration continue : principes, utilisation : Jenkins, GitLab-CI

Panorama des outils de gestion de projet agile Scrum

# Formation Agile Scrum : Approfondissement

Durée : 2 jours

Public : Toute personne devant être impliquée dans un projet Scrum : ScrumMaster, Product Owner, Scrum Developer, DSI

Pré-requis : Notions de gestion de projets Agile

Objectifs : Maîtriser la gestion de projets agiles avec la méthode Scrum

Rappel du cadre Scrum

Cycle de vie d'un projet Scrum

Rôles définis par Scrum : Product Owner, Scrum Master, Team

Time boxes : Sprint planning, Sprint Review, Sprint Retrospective, Daily Scrum

**Serious Game : Répartition des tâches par rôle**

Partager la vision du produit

Vision du produit : définition, intérêt

Forme d'une vision : Lean Canvas

Toolbox

Exemple complet de Product Vision Board

**Atelier : Construction et analyse d'une vision de produit**

Recueillir le besoin

Faire émerger le besoin

Techniques de recueil

Organisation des rôles des acteurs (RACI)

Formaliser le besoin : IEEE vs Use Cases vs User Stories

Ateliers :

- Construction d'une matrice RACI pour organiser les rôles d'un projet. - conduite d'un entretien de recueil (technique des 9 cases)

- Animation **d'un atelier de brainstorming et construction de mind maps.**

Construire le produit

Construction d'une Roadmap

Des objectifs aux fonctionnalités : Impact mapping Identifier et structurer les fonctionnalités :

Story mapping Déduire la roadmap de la story map Démarche de construction du product

backlog Rédaction des user stories : forme, spécification INVEST Compléments d'une user

stories : contraintes opérationnelles, maquettes, .. Ecriture des tests d'acceptation (méthode

Behaviour Driven Development) Outils d'automatisation des tests Ateliers :- **Construction**

**d'une impact map à partir de la vision du produit.- Construction d'une story map,**

**identification des MMF (Minimum Marketable Features). - Construction de la roadmap.-**

**Rédaction de user stories et atelier streamig pour les tests d'acceptation.**

Estimer et prioriser les stories

Techniques d'estimation : ideal days vs story points

Mise en oeuvre de la méthode des story points Fiabilisation de l'estimation Facteurs



d'hiérarchisation des besoins Méthodes de priorisation : Kano vs MOSCOW vs Poids relatifs  
Priorisation par thèmes : themes screening / scoring Gestion des risques

Regroupement et ordre final des stories dans le product backlog

**Ateliers :**

- **Poker planning game pour l'estimation des stories.**
- **Calcul de la priorité avec la méthode des poids relatifs. - Remaniement du product backlog.**

Planifier et piloter un projet Scrum

Roadmap et plan de release

Réunion de sprint planning, construction du sprint backlog

Daily Scrum meeting Indicateurs de performances : Burn Down et Burn Up chart Suivi de

l'avancement : Kanban board, Parking Lots Outil de checkpoint des équipes : ROTI, One

Word, ... Revue de sprint Rétrospective de sprint Ateliers :- **Animation d'une réunion de**

**sprint planning, découpage de stories en tâches. - Revue de sprint : acteurs,**

**démonstration, validation des stories. - L'art de la rétrospective : speed car/boat, Turn the**

**tables, Jeopardy- Utilisation d'outils de gestion de projets agiles.**

Etablir un contrat agile

Différentes formes d'un contrat agile Mise en place d'un Plan d'Assurance Qualité (PAQ)

Contractualisation par Sprint

## Formation Préparation de la certification PSD (scrum.org)

Durée : 1 jour

Public : Développeurs amenés à travailler dans une équipe agile Scrum

Pré-requis : Connaissances du cadre Scrum

Objectifs : Définir l'action d'un développeur dans un projet agile - Apprendre les bonnes pratiques de développement et de tests dans un environnement agile Scrum.

### Maîtriser la méthode agile Scrum

Le cadre Scrum / Guide

Cycle de vie d'un projet Scrum

Rôles définis par Scrum : Product Owner, Scrum Master, Team  
Time boxes : Sprint planning, Sprint Review, Sprint Retrospective, Daily Scrum

Artéfacts : Product Backlog, Sprint Backlog, Burndown chart  
Règles et principes clé de Scrum  
Responsabilités / rôle  
Atelier : **analyse du guide de Scrum et pratiques XP**

### Analyser le produit

Analyse de la vision du produit

Compréhension de l'organisation des rôles (RACI)  
Démarche de construction du product backlog  
Rédaction des user stories : forme, spécification INVEST  
Compléments d'une user stories : contraintes opérationnelles, maquettes,...  
Ecriture des tests d'acceptation (méthode Behaviour Driven Development)  
Outils d'automatisation des tests  
Ateliers :- **Construction d'une story map, identification des MMF (Minimum Marketable Features). - Construction de la roadmap.- Rédaction de user stories et atelier three amigospour le testsd'acceptation.**

### Estimer et prioriser les stories

Techniques d'estimation : ideal days vs story points

Mise en oeuvre de la méthode des story points  
Fiabilisation de l'estimation  
Facteurs d'hierarchisation des besoins  
Méthodes de priorisation : Kano vs MOSCOW vs Poids relatifs

Priorisation par thèmes : themes screening / scoring

Gestion des risques

Regroupement et ordre final des stories dans le product backlog

# Formation SAFe Scrum Master (SAFe SM)

**Durée :** 2 jours

**Public :** Scrum Master, futur Scrum Master, Coachs agiles, Toute personne impliqué dans la mise en place de l'agilité à grande échelle

**Pré-requis :** Avoir des connaissances sur les concepts de l'agilité (Scrum, XP, Kanban, Lean), Disposer d'une expérience significative au sein d'une équipe Agile, Niveau correct d'anglais écrit

**Objectifs :** Comprendre les principes du Scaled Agile Framework (version 4.5) - Caractériser le rôle de Scrum Master - Comprendre et être capable de mettre en œuvre un Program Increment - Appréhender les notions de coaching d'une équipe agile - Préparer l'examen « SAFe SSM »

## Scrum dans le contexte SAFe

- Les concepts de l'agilité, Le manifeste Agile et la mise à l'échelle,
- Les éléments fondamentaux de Scrum,
- Les principes et les rôles SAFe,
- Agile Release Train (ART).

## Le Scrum Master

- Ses responsabilités vis-à-vis de l'équipe,
- Ses responsabilités dans un contexte SAFe,
- Les compétences attendues.

## Planification d'un Program Increment (PI planning)

- Préparation de l'événement,
- Features et types de stories,
- Concevoir la planification du PI,
- Déterminer les objectifs du PI planning,
- Maîtriser les risques
- Améliorer le processus de planification de l'incrément

## Faciliter l'exécution de l'itération

- Le rôle du Scrum Master dans la planification,
- Planifier et suivre l'avancement de l'itération, Affiner le Backlog,
- Faciliter la revue de l'itération et la démo du système,
- Le rôle du Scrum Master dans la qualité au quotidien,
- Déploiement continu avec DevOps,
- Livrer à la demande,

Les mesures à utiliser.

Terminer un Program Increment

Mesure de la performance du Program Increment,

Faciliter le workshop d'inspection et d'adaptation.

Coacher une équipe agile

Servant leader,

Les étapes de formation d'une équipe performante selon Tuckmann,

Faciliter les évènements de l'équipe SAFe, Les questions d'un coach, Développer l'esprit d'équipe et résoudre les conflits.

Préparation à l'examen « SAFe SSM »

Examen :

Le **passage de l'examen** s'effectue **en différé** dans le mois qui suit la formation, en **ligne et en anglais**.

Cela consiste en un QCM de 90 minutes, comportant 45 questions. Un score minimum de 73% est requis pour réussir l'examen.

Méthodes et moyens pédagogiques Exposé théorique, pratique et participatif, dispensé en FR ou en EN, alternant :

Présentation théorique, Discussion autour des cas organisationnels des participants, Simulation d'un « PI planning », Ex

**Sont fournis aux stagiaires :**

**Support en EN présenté par un instructeur certifié, format papier prise de note (noir & blanc) ou en accès en ligne, QCM** et examens blancs en EN, Requier une salle équipée de PC (formateur et stagiaires), de connexions internet, d'un vidéoprojecteur et d'un tableau blanc ou paperboard. Mode de dispense

En présentiel.

# BIG DATA

## Formation Apache Spark

**Durée :** 4 jours

**Public :** Développeurs, architectes système et responsables techniques qui veulent déployer des solutions Spark dans leur entreprise

**Pré-requis :** Avoir une expérience professionnelle de programmation du niveau de la formation 471, Programmation Java : Les fondamentaux, ou de la formation 419, Programmation en C# : Les fondamentaux. Avoir entre 3 et 6 mois d'expérience de programmation avec un langage orienté objet

### Objectifs :

- Développer des applications avec Spark - Utiliser les bibliothèques pour SQL, les flux de données et l'apprentissage automatique - Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles - Développer des applications métier qui s'intègrent à Spark

### Introduction

Définition du Big Data et des calculs À quoi sert Spark Quels sont les avantages de Spark

### Applications évolutives

Identifier les limites de performances des CPU modernes Développer les modèles de traitement en parallèle traditionnels

### Créer des algorithmes parallèles

Utiliser la programmation fonctionnelle pour l'exécution des programmes en parallèles Retranscrire des difficultés rencontrées sur le terrain dans des algorithmes parallèles

### Structures de données parallèles

Répartir les données dans le cluster avec les RDD (Resilient Distributed Datasets) et les DataFrames Répartir l'exécution des tâches entre plusieurs nœuds Lancer les applications avec le modèle d'exécution de Spark

### Structure des clusters Spark

Créer des clusters résilients et résistants aux pannes Mettre en place un système de stockage distribué évolutif

### Gestion du cluster

Surveillance et administration des applications Spark Afficher les plans d'exécution et les résultats

### Choisir l'environnement de développement

Réaliser une analyse exploratoire avec le shell Spark Créer des applications Spark autonomes

## Utiliser les API Spark

Programmation avec Scala et d'autres langages compatibles  
Créer des applications avec les API de base  
Enrichir les applications avec les bibliothèques intégrées

## Interroger des données structurées

Traiter les requêtes avec les DataFrames et le code SQL embarqué  
Développer SQL avec les fonctions définies par l'utilisateur (UDF)

Utiliser les ensembles de données aux formats JSON et Parquet

## Intégration à des systèmes externes

Connexion aux bases de données avec JDBC  
Lancer des requêtes Hive sur des applications externes

## Qu'appelle-t-on flux de données ?

Utiliser des fenêtres glissantes  
Déterminer l'état d'un flux de données continu  
Traiter des flux de données simultanés  
Améliorer les performances et la fiabilité

## Traiter les flux des sources de données

Traiter les flux des sources intégrées (fichiers journaux, sockets Twitter, Kinesis, Kafka)  
Développer des récepteurs personnalisés  
Traiter les données avec l'API Streaming et Spark SQL

## Classer les observations

Prévoir les résultats avec l'apprentissage supervisé  
Créer un élément de classification pour l'arbre de décision

## Identifier les schémas récurrents

Regrouper les données avec l'apprentissage non supervisé  
Créer un cluster avec la méthode k-means

## Développer des applications métier avec Spark

Mise à disposition de Spark via un service Web RESTful  
Générer des tableaux de bord avec Spark

## Utiliser Spark sous forme de service

Service cloud vs. sur site

Choisir un fournisseur de services (AWS, Azure, Databricks, etc.)

Développer Spark pour les clusters de grande taille  
Améliorer la sécurité des clusters

multifournisseurs  
Suivi du développement continu de produits Spark sur le marché

Projet Tungsten : repousser les performances à la limite des capacités des équipements modernes

Utiliser les projets développés avec Spark  
Revoir l'architecture de Spark pour les plateformes mobiles

## Formation Data Science (R et Hadoop)

Durée : 5 jours

**Public :** Professionnels des bases de données, managers, analystes de données, data scientists et assistants à maîtrise d'ouvrage. Cette formation est très utile pour les professionnels chargés de gérer les prévisions et les tendances

**Pré-requis :** Connaissances en matière de programmation et de statistiques sont utiles sans toutefois être obligatoires

**Objectifs :**

- Appliquer des techniques d'exploration des données pour améliorer la prise de décisions métier à partir de sources de données internes et externes - Prendre une longueur d'avance sur vos concurrents avec l'analyse des données structurées et non structurées - Prédire un résultat en utilisant des techniques d'apprentissage automatique supervisé

Exploration et analyse des données avec R

Charger, interroger et manipuler des données avec R  
Nettoyer les données brutes avant la modélisation  
Réduire les dimensions avec l'analyse en composantes principales (ACP)  
Développer les fonctionnalités de R avec les packages définis par l'utilisateur

Faciliter la pensée analytique avec la visualisation des données

Explorer les caractéristiques d'un ensemble de données à travers la visualisation  
Représenter graphiquement la distribution des données avec des boîtes à moustaches, des histogrammes et des diagrammes de densité  
Identifier les valeurs hors normes

Explorer les données non structurées pour les applications métier

Traitement préliminaire et préparation des données non structurées pour une analyse plus approfondie  
Décrire un ensemble de documents avec une matrice termes-documents

Difficultés supplémentaires liées au Big Data

Examiner les architectures de MapReduce et Hadoop  
Intégrer R et Hadoop à RHadoop

Estimer les valeurs avec les règles de régression linéaire et logistique

Modéliser la relation entre une variable de sortie et plusieurs variables d'entrée  
Interpréter correctement les coefficients des données continues et qualitatives

Techniques de régression pour manipuler le Big Data

Traiter les ensembles de données volumineux avec RHadoop  
Créer des modules de régression pour RHadoop

Identification automatique de chaque nouvel élément de données

Utiliser des arbres de décision pour prédire les valeurs cible  
Appliquer des règles de probabilité pour prédire les résultats avec le modèle Naive Bayes  
Combiner les variables de prédiction des arbres et les forêts aléatoires dans RHadoop

Évaluer les performances des modèles

Visualiser les performances des modèles avec une courbe ROC  
Évaluer les modèles de classification avec des matrices de confusion

Identifier des groupes encore inconnus dans un ensemble de données

Segmenter le marché client avec l'algorithme K-Means  
Trouver des similarités avec les mesures des distances  
Créer des clusters en forme d'arbres et des mises en cluster hiérarchiques  
Mettre en cluster les tweets et les fichiers texte pour mieux les comprendre

Mettre à jour les connexions avec l'analyse des associations

Identifier les connexions importantes avec l'analyse des réseaux sociaux  
Comprendre l'utilisation des résultats de l'analyse des réseaux sociaux à des fins marketing

Définir et évaluer des règles d'association

Identifier les préférences réelles des clients à partir d'un ensemble de données transactionnelles  
pour améliorer l'expérience utilisateur  
Calculer les indices de support et de confiance et le lift pour différencier les bonnes règles des mauvaises



# Formation Elasticsearch : Rechercher et analyser en temps réel

Durée : 3 jours

Public : Administrateurs systèmes

Pré-requis : Connaissances en administration système, préférablement Java

Objectifs :

- Mettre en place Elasticsearch pour indexer des documents - Comprendre l'écosystème Elasticsearch

Introduction

Présentation du projet Elasticsearch

Pré-requis, Présentation de Lucene

Installation

Atelier pratique : installation d'**Elasticsearch**

Découvrir Elasticsearch

Configuration

Notion de cluster ElasticSearch

Un peu de JSON

Nœuds, Index et Type, Documents

Insérer, indexer, extraire, supprimer des documents **Atelier pratique : ajout de document simple dans l'index, récupération**

Mapping de données

Correspondance de données dans Elasticsearch

Types : valeurs exactes et texte complet (full text, exact values)

Indexes inversés Objets et types complexes

L'analyse, et les analystes

Extraction de texte, analyse de texte

Filtres de caractères, jetons, filtres de jeton Analyses intégrées : standard, analyse des espaces, analyse linguistique **Atelier pratique : utilisation du mapping, et des analystes standards pour rechercher des documents**

La recherche effective

La recherche lite, et vide

Query DSL : le langage de recherche Elastisearch

Filtres et requêtes

Combinaisons

Recherche full text, multi-champs

Comparaison partielle, par proximité Pertinence des résultats **Atelier pratique : écrire des requêtes plus compliquées, les passer en JSON à Elasticsearch**

Pour aller plus loin

Elasticsearch et les plugins

Utilisation de River pour aller chercher des données externes (wikipedia, twitter,

...)Maintenance du cluster et des nœudsIntégration Java (et autres)**Atelier pratique :**

**utilisation de River avec Wikipedia, administrer un nœud, piloter une recherche depuis**

**Java**

## Formation Hadoop : Déployer du Big Data

**Durée :** 4 jours **Public :** Administrateurs systèmes **Pré-requis :** Connaissances en administration système, préférablement Java

### Objectifs :

Comprendre le Big Data et ses enjeux - Savoir déployer Hadoop et son écosystème - Comprendre HDFS, MapReduce - Structurer les données avec HBase - Écrire des requêtes avec HiveQL - Lancer une analyse avec Pig

### Référence :

#### Introduction au Big Data

Qu'est-ce que le Big Data ?

Source des données : l'homme, la machine

La problématique détaillée Position de Hadoop dans le paysage

#### Introduction à Hadoop

L'origine du projet

Le système de fichiers HDFS Comprendre l'algorithme MapReduce L'environnement

d'Hadoop : HBase, ZooKeeper, Hive, Pig... L'API YARN

#### Mettre Hadoop en place : HDFS

Du mode autonome au mode complètement distribué en cluster Pré-requis, distributions Hadoop Cluster Hadoop : NameNode, ResourceManager, DataNode, NodeManager Les fichiers de configuration Opérations de base sur le cluster HDFS : formatage, démarrage, arrêt

**Atelier pratique : installer Hadoop sur 2 nœuds, formater et manipuler HDFS**

#### Travailler avec MapReduce

L'intérêt de MapReduce Mappers, reducers, parallélisme et indépendance des

traitements Entrées, sorties Soumission d'un job à Hadoop **Atelier pratique : exécuter une**

**tâche via MapReduce, avec sortie dans HDFS**

#### Une base de données distribuée : HBase

L'accès aléatoire, temps réel, lecture-écriture au Big Data Fonctionnalités de HBase, NoSQL

Pré-requis, configuration Manipulation via le shell HBase **Atelier pratique : mettre en place**

**HBase sur Hadoop, créer et manipuler une table**

#### Et pourquoi pas un peu de SQL avec Hive ?

Présentation de Hive

Gérer le schéma : bases, tables, vues, partitions Manipulation des données, requêtes et map-

reduce avec HiveQL Audits et journal d'erreurs **Atelier pratique : chargement de données**

**massives dans Hive, requêtes**

## Analyser les données avec Pig

Présentation, installation du projet Apache Pig Exécutions de Pig en local, en mode map-reduce Scriptor pour Pig Le langage Pig Latin

Manipulations de données et stockage avec Pig

**Atelier pratique : écrire un script contenant du Pig Latin pour une tâche simple, et l'exécuter en local, puis en mode map reduce.**

## Aller plus loin avec Hadoop

Gérer les logs et l'audit de tâches Hadoop Découvrir MRUnit pour les tests unitaires dans Hadoop Débogage en local Surveillance des performances Atelier pratique : mise en place d'un job MapReduce plus complexe avec traces et tests unitaires

# JAVA SE

## Formation Java SE Initiation

Durée : 3 jours

Public : Tous

Pré-requis : Notions de programmation

Objectifs : Pouvoir réaliser des applications en Java – Savoir choisir les technologies adaptées et mettre en place des interfaces efficaces

### Découvrir la plateforme Java

Historique, versions

Editions Java : Java SE, Java EE, Java ME

Compilation et interprétation par la JVM (Java Virtual Machine) Technologies/frameworks

Java et positionnement Environnement de développement Empaquetage et déploiement

d'une application Java **Atelier : Installation du JDK (Java Development Kit) et d'un IDE**

**(Eclipse/NetBeans) - Structure d'un projet, buildPath,...**

### Maîtriser les bases

Utilisation de variables, constantes, opérateurs Types simples et types références Transtypage, Wrappers Expression de conditions : if/else, switch, opérateur ternaire Utilisation de boucles : for, while, do while Manipulation de tableaux

Factorisation de codes avec méthodes Surcharge, arguments variables, récursivité Commenter et documenter du code **Atelier : Multiples exemples de manipulation de structures de contrôles et de fonctions**

### Apprendre l'objet

Définition de classes, POJO vs JavaBean Déclaration des membres d'instance / de classe (static) Constructeurs et instanciation Cycle de vie d'un objet en mémoire Diagramme de classes (UML) Agrégation d'objets (association) Encapsulation : getters et setters Extension de classes (Héritage)

Comparaison d'objets

Classes abstraites

Interfaces et implémentation

Polymorphisme

**Atelier : Modélisation de problèmes en objet**

### Gérer les exceptions

Définition, types d'exceptions Capturer et traiter une exception (try/catch/finally)

Lever/Remonter une exception (throw/throws)

Création d'exceptions  
**Atelier : Gestion des exceptions susceptibles d'être déclenchées dans une application**

Utiliser des collections

Présentation de l'API disponible, generics

Comparatif, choix d'un type de collection

Classes essentielles : ArrayList, HashMap,...

Parcours, opérations sur des collections et tris

**Atelier : Manipulation de collections d'objets**

Manipuler des fichiers

Flux binaires / caractères  
Lecture et écriture de fichiers  
Utilisation de buffers  
Manipulation de chemins, répertoires, surveillance  
Sérialisation d'objets : binaire, XML  
Externalisation de configuration dans des .properties  
Gestion des logs : java.util.logging, Log4j

**Atelier : Implémentation d'exports et imports depuis des fichiers**

# Formation Java SE Approfondissement

Durée : 2 jours

Public : Tous

Pré-requis : Avoir suivi le stage "Java initiation" ou posséder les connaissances équivalentes

Objectifs : Connaître et maîtriser les concepts avancés de la programmation Java - Réaliser et déployer des applications complètes, performantes et maintenables

## Construire des interfaces graphiques

Présentation de Swing : containers, widgets

Fenêtres modales/non modales, boîtes de messages

Positionnement des contrôles (Layout Managers)

Gestion des évènements : claviers, souris

WYSIWYG disponibles

Gestion du redimensionnement

**Atelier : Application complète de gestion**

## Accéder à des bases de données

Présentation de l'API JDBC  
Ecriture de requêtes et traitement des résultats  
Gestion des transactions  
Mapping relationnel objet (Pattern DAO)

Présentation de frameworks

ORM  
**Atelier : Organisation et implémentation d'une couche d'accès aux données**

## Notions avancées

Communications réseau  
Gestion des processus  
Options JVM  
Optimiser du code,

généricité  
Introspection (Reflection API)  
**Atelier : optimisation de code et déploiements**

# LINUX

## Formation Linux Administration Avancé LPI 201: Maintenance, Customisation, Réparation

Durée: 3 jours

Public: Tous

Pré-requis: Avoir la certification LPI 102 ou les connaissances équivalentes

Objectifs: Apprendre à superviser, réparer, customiser et maintenir des serveurs Linux

### Les commandes Unix

- Paramétrage avancé de l'environnement

- Paramétrage du shell (.bashrc et .bash\_profile)

- Programmer en shell : loop, test, SQL Automatiser les travaux courants d'administration

- Les commandes sed et awk

- Connaître l'architecture des comptes utilisateurs

- Gérer le PATH **Atelier: personnaliser son propre environnement**

### Les interfaces graphiques

- Configurer le serveur X11 et les cartes graphiques

- Gérer les font du serveur X11 Gérer les claviers avec le serveur X11 Les commandes de base

- du serveur graphique Gestion des interfaces LightDM, KDM et GDM **Atelier: installer et**

- personnaliser sa propre interface graphique**

### Les taches d'administration

- L'authentification

- Gestion des droits des utilisateurs

- Connaissance du fichier password et de PAM

- Gestion des tâches planifiées Configuration des paramètres locaux **Atelier: configurer**

- l'authentification du système ; planifier un script.**

### Les service de base

- Configuration de l'horloge et de la timezone

- Configuration d'un serveur NTP

- Configuration des demon syslog et logrotate

- Connaissance der-syslog

- Configuration de la messagerie et connaissance des différents MTA



Configuration et gestion du serveur d'impression

Atelier: **organiser la gestion des log de son propre systeme**

Les principes de base de la sécurité

Audit d'un système : repérer les droits accordés à tort

Gestion des mots de passe

Audit des ports ouverts

Définir les limites des ressources accordées aux utilisateurs

Auditer les logs de connexion

Gérer les droits de sudo

**Atelier: auditer les vulnérabilités de son propre système et activer une protection par pare-feu**

# Formation Linux Administration Avancé LPI 202: Services Web, Mails, DNS

Durée: 3 jours

Public: Administrateurs Systèmes Linux

Pré-requis: Avoir la certification LPI 201 ou les connaissances équivalentes

Objectifs: Apprendre à administrer les services Web, mails, Samba, DNS

## Le DNS

Configuration et gestion de BIND Définition de zone Recharger une nouvelles définition

Connaissance de dnsMasq et powerDNS

Le reverse DNS

BIND dans un environnement chroot Connaissance de dnsSec et ses outils

## Le WEB

Configuration et gestion d'Apache

Définition des accès Gestion des modules, notamment PERL et PHP Authentification Le virtual

hosting Configuration et gestion SSL Génération des certificats Installation des

certificats Configuration et gestion d'un proxy Gestion des ACL

## Le partage de fichiers

Configuration et gestion de samba

Les outils de samba

Monter un volume SMB

Mapper les utilisateurs windows et linux

Configuration et gestion de NFS

Les outils de NFS

Monter un volume NFS TCP Wrapper

Configuration et gestion de dhcpd

Les outils de dhcp

Configuration et gestion de PAM

Configuration et gestion de openLDAP

Schemas et whitepage dans LDAP Connaissance du demon SSSD

## Le service mail

Configuration et gestion de postfix

Connaissance du protocole SMTP

Connaissance d'exim et sendmail

Configuration et gestion de procmail

Configuration et gestion de IMAP et POP

## La sécurité

Configuration et gestion d'IPTable

Les outils et commandes d'IPTable

La redirection de port et IP forwarding

Créer des règles de filtrage  
Connaissance du filtrage IPV6  
Configuration et gestion de pureFTP  
et vs

Compréhension des différences entre connexion active et passive

Configuration et gestion d'openSSH

Les autorisations et restrictions pour le super utilisateur et les autres utilisateurs

Gestion des clés et de leurs échanges  
Les outils de test des ports  
Connaissance des référents  
vers lesquels reporter les failles et anomalies du system Outils d'

Connaissance d'openVAS et snort

Configuration et gestion d'openVPN

## Formation Linux Avancé :

### Serveur d'entreprises avec Samba/LDAP/Netfilter/Squid

Durée : 5 jours

Public: Administrateurs

Pré-requis: Linux Administration: les bases

Objectifs: Savoir installer, configurer et maintenir un serveur de fichier/d'authentification/routeur/firewall pour une entreprise

#### Introduction

Linux personnel et Linux en entreprise

Présentation de cas pratiques

Gestion des disques

Technologies RAID et LVM

Choix des systèmes de fichiers

Gestion des sauvegardes : les différentes solutions possibles

Atelier pratique : mise en place de sauvegardes

Gestion des quotas

Atelier pratique : mise en place de quotas

#### Premier pas avec LDAP

Introduction

La notion d'annuaire

Principe des annuaires distribués

Le standard LDAP  
Architecture de LDAP  
Les modèles de LDAP  
Modèle d'information  
Modèle de désignation  
Modèle de service  
Modèle de sécurité  
Gestion distribuée des annuaires

LDAP  
Atelier pratique : création théorique d'un annuaire LDAP  
OpenLDAP  
Installation sur une distribution GNU/Linux  
Mettre en place les schémas nécessaires

Mise en place d'une structure de données  
Utilisation de Ldapsearch  
Mise en place d

Atelier pratique : installation et mise en place d'un annuaire LDAP

#### L'authentification sous Linux

Principes généraux  
Introduction à l'architecture PAM  
Atelier pratique : utilisation de LDAP comme base de données d'a

Présentation du protocole NFS

Présentation du protocole

Les différents serveurs disponibles sous Linux  
Mise en place  
Pourquoi il nécessite LDAP ou NIS

Atelier pratique : mise en place de NFS

## L'interconnexion des systèmes Unix et Windows: Samba

Présentation du protocole SMB

Présentation de Samba Commandes Samba utiles Mettre en œuvre une configuration de base

Gestion des utilisateurs

Configuration des partages

Partage d'imprimantes

Atelier pratique : mise en place de partages Samba

## Le VPN

Comprendre les différents niveaux de tunnels

Les protocoles et leurs implémentations

Présentation de OpenVPN Atelier pratique : mise en place d'OpenVPN

## Le firewall

La sécurité au niveau du protocole IP

Fonctionnement du logiciel Netfilter

Les outils de configuration (Iptables et interfaces)

Atelier pratique : configuration de Netfilter

## Le proxy cache

Problématiques auxquelles répond un serveur de cache

Présentation de SQUID Présentation de Dansguardian Atelier pratique : installation & configuration de Squid

## Gestion des pannes

Anticiper pour éviter les pannes

Bons réflexes à avoir en cas de pannes

Méthodologie pour la résolution de pannes

# Formation Linux Expert: LDAP

Durée: 2 jours

Public: Administrateurs systèmes Linux

Pré-requis: Avoir suivi la formation "Linux Avancé : Serveur d'entreprises avec Samba/LDAP/Netfilter/Squid" ou posséder les connaissances équivalentes

Objectifs: Découvrir LDAP - implémenter OpenLDAP dans différents services réseau

Référence: LIN332-F

L'authentification distribuée en environnement mixte avec LDAP

## Introduction

- La notion d'annuaire

- Principe des annuaires distribués

- Le standard LDAP

## Architecture de LDAP

- Les modèles de LDAP

- Modèle d'information

- Modèle de désignation

- Modèle de service

- Modèle de sécurité

- Gestion distribuée des annuaires LDAP

## La syntaxe LDAP

- Les éléments du langage

- Les URLs LDAP

- Le format LDIF

- Le Directory Index Tree

- Atelier pratique : création théorique d'un annuaire LDAP**

## OpenLDAP

- Installation sur une distribution GNU/Linux

- Mettre en place les schémas nécessaires

- Mise en place d'une structure de données

Utilisation de Ldapsearch

Mise en place de la réplication

**Atelier pratique : installation et mise en place d'un annuaire LDAP**

### Les services utilisant LDAP

L'authentification sur un système GNU/Linux

L'authentification sur un serveur (HTTP, FTP, SMTP, POP, IMAP, ...)

Déployer un annuaire d'entreprise

**Atelier pratique : utilisation pratique de LDAP**

# PYTHON

## Formation Python Initiation

Durée: 3 jours

Public: Développeurs

Pré-requis: Aucun

Objectifs: Connaître les possibilités et limites de Python - Maîtriser la syntaxe essentielle de Python

### Introduction

Historique, alternatives

A propos de Python 2 et Python 3

Présentation et installation de Python dans un environnement Windows ou Linux

Mise en œuvre de Python : accès au terminal

Environnements de développement assisté ; association à des outils connus et éditeurs pratiques

### Syntaxe basique

Syntaxe de base, en interprétation directe et en script  
Déclarer des variables en Python, types internes et leur utilisation

Utiliser les opérateurs et expressions

Présentation et détail des structures de contrôle : les tests, les boucles  
Utiliser les boucles et tests pour réaliser des algorithmes

Se servir des tableaux, algorithmes divers de traitement de données massives

Types de données et opérations avancées (ensembles, dictionnaires, ordonnancement...)

Gestion des exceptions

Atelier pratique : une application interactive de gestion, petit à petit

### Structuration procédurale

Intérêt des structurations du code dans le projet et pour le groupe de projets

Créer et organiser les fonctions

Retour des fonctions, absence de retour, multiples retours et intérêt dans un algorithme complexe

Espaces de noms, manipulation

Créations de bibliothèques de fonctions, documentation

Atelier pratique : découpage de l'existant et quelques créations



## Bibliothèques essentielles

Lire, écrire, modifier les chaînes de caractères

Calculer avec Python

Gérer les dates, heures, intervalles de temps avec Python

Fonctions diverses nécessaires

Atelier pratique : application complète et interactive

# Formation Python: Approfondissement

Durée: 2 jours

Public: Tous

**Pré-requis:** Avoir suivi le stage Python initiation ou posséder les connaissances équivalentes

**Objectifs:** Être capable de prévoir, développer, tester, déployer une application complète et complexe en Python

## Introduction

- Présentation générale

- Rappels de Python

## Programmation Orientée Objet

- Théorie et utilisation ; pourquoi l'objet

- Connaître la syntaxe fondamentale des classes avec Python

- Utilisation et création de classes de services et de données

- Agrégation, héritage, autres points avancés de la POO

- Bilan sur l'utilisation de l'objet

- Atelier pratique : augmentation de l'application par des classes de données

## Bibliothèques additionnelles courantes

- Interfaces graphiques - TkInter, GTK, wxWidgets

- Sockets et CGI

- Manipulation de fichiers divers

- Utiliser des bases de données avec Python, DBAPI

- Présentation de Python et le Web; l'exemple de CherryPy, Django

- Atelier pratique : une application graphique complète**

# Formation Python Avancé : Fonctionnalités avancées, domaines d'applications

Durée: 2 jours

Public: Tous

Pré-requis: Avoir suivi le stage Python initiation ou posséder les connaissances équivalentes

Objectifs: Être capable de prévoir, développer, tester, déployer une application complète et complexe en Python

## Introduction

Installation d'un IDE

Les versions du langage :

- Exemples de différence- Solutions de contournement

Structure d'un projet

## Éléments avancés du langage

Les pièges courants

Les contextes

- Exemples de contexte connus
- Atelier : Création d'un contexte
- Les itérables- Exemples d'itérables connus- Fonctionnement de la boucle `for`
- Atelier : Création d'un itérable
- Les générateurs
- Fonctionnement
- Atelier : Création d'un générateur
- Les décorateurs
- Exemples de décorateurs connus
- Fonctionnement
- Atelier : Création d'un décorateur

Les métaclasses- Fonctionnement

- Atelier : Créat

## Fonctionnalités utiles dans tout projet

Les environnements virtuels

Les dépendances

Les packages redistribuables

- Créer un package redistribuable
- Installer son package dans un autre projet

- Définir des dépendances
- Inclure des fichiers de données

Les tests unitaires avec `unittest` - Principe

- Test de retour des fonctions
  - Court-circuiter les effets de bord avec `unit
  - Autres frameworks de test
- Gestion des logs  
Générer une documentation  
La compilation

Python dans différents domaines d'application

Les programmes en ligne de commande

- Accès aux arguments
- Analyse des arguments avec `argparse`
- Les alternatives à `argparse`
- Atelier : Calcul de la taille de répertoires

Utilisation d'un ORM (Object-Relational Mapping)

- Déclaration des entités
- Manipulation d'entités
- Atelier : Sauvegarde de la taille des répertoires en base de données

Créer une petite application web

- Les frameworks web
- Atelier : Application web avec `Flask` permettant de visualiser du code source à distance

Créer un client web

- Envoyer des requêtes HTTP
- Extraction d'information à partir des données reçues
- Parser de l'HTML
- Parser du JSON
- Communiquer avec une API Automatiser un navigateur web avec `selenium`
- Atelier : Créer un comparateur de technologie

Les interfaces graphiques- Liste des frameworks graphiques

- Installation de `PyQt5`
- Atelier : Application graphique de comparaison de fichier

Programmation concurrente

- Le multithreading
  - Présentation
  - Le GIL : Global Interpreter Lock
- Le multiprocessing
  - Présentation
  - Communication entre les processus

#### Création d'une API REST

- Exemple d'API REST
- Création manuelle
- Création avec `Flask-RESTful`
- Mise en production

# DEVOPS

## Formation DevOps – Foundation - Certification

3 jours Tous

**Durée :** Développeurs, architectes, administrateurs systèmes

**Public :**

- Objectifs :**
- Comprendre l'utilité du cadre méthodologique de DevOps pour l'entreprise et pour les services informatiques
  - Connaître les principes de DevOps
  - Comprendre les relations avec les méthodes agiles, le Lean et l'ITSM
  - Comprendre comment améliorer les flux de travail, la communication et les boucles de feedback entre les équipes
  - Connaître les facteurs critiques de succès et les indicateurs clés de performance
  - Préparer et passer l'examen "DevOps Foundation" du Devops Institute

## ORIGINES DE DEVOPS

- Contexte
- Réalité des entreprises
- Le mouvement DevOps

## DEVOPS – PRÉSENTATION GÉNÉRALE

- Définition
- Ce qui est DevOps...
- Ce que n'est pas DevOps...
- DevOps et la performance IT
- Les indicateurs de mesure de la performance IT
- DevOps et l'automatisation

## DEVOPS ET LES AUTRES RÉFÉRENTIELS

- DevOps et l'Agilité
- DevOps et le Lean
- DevOps et ITIL
- LA CULTURE DEVOPS
- Compétences
- Structures organisationnelles
- Rôles
- Collaboration et communication

## LES PRATIQUES DEVOPS

- Intégration continue
- Livraison continue
- Déploiement continu
- VSM (Carte de flux de valeur)
- Kanban
- Théorie des contraintes
- Retours d'expérience
- La roue de Deming (PDCA)
- "Improvement Kata"
- Préparation à l'échec
- Amélioration des processus ITSM
- DEVOPS ET LES OUTILS
- Les pratiques d'automatisation
- Préoccupations et catégories

## MISE EN PLACE DE LA CULTURE DEVOPS

- Par où commencer...
- Les risques, les facteurs de succès

## PRÉPARATION ET PASSAGE DE L'EXAMEN DE CERTIFICATION "DEVOPS FOUNDATION" DU DEVOPS INSTITUTE

- Révisions
- Passage de l'examen en fin de session
- Précisions : examen en anglais sous forme d'un QCM (40 questions)