

# Faire de l'intégration continue avec JENKINS

**Programme** (Mis à jour le 28/06/2024)

## La théorie des tests

- Les différents niveaux de tests (Unitaires, intégration, vérification et qualification)
- La non régression
- Pourquoi mettre en œuvre des procédures de tests automatiques ?
- Présentation des outils utiles pour les tests

## Mise en œuvre de tests unitaires en Python

- Présentation des outils d'automatisation de jeux de tests en Python
- Installation et mise en œuvre de DocTest
- Installation et mise en œuvre de UnitTest
- Intégration dans l'IDE (Eclipse ou PyCharm)

## Code Coverage en Python

- Pourquoi faire du « code coverage » ?
- Présentation de l'outil Coverage et intégration dans l'IDE
- Analyse des rapports produits

## Introduction à l'intégration continue

- Qu'est ce que l'intégration continue ?
- Historique de Jenkins
- Installation de Jenkins
- Les plugins de Jenkins

## Configuration de Jenkins

- Prise en main de l'interface
- Configuration de la sécurité
- Configuration des divers outils
- Configuration pour l'envoi automatique de mails

## Création de projets Jenkins

- Qu'est ce qu'un « job » (un projet) ?
- Choix du SCM (Source Code Management)
- Quand lancer le build ?
- Gestion des builds

## Automatisation des tests dans Jenkins

- Configurer l'automatisation des tests
- Intégration des rapports sur l'exécution des jeux de tests
- Profilage des batteries de tests

## Outils d'analyse de qualité des codes

- Introduction
- Intégration des rapports dans Jenkins

## Produire la documentation

- Présentation des outils PyDOc et Doxygen

### Référence

THIL1983

### Durée

2 jours / 14 heures

### Prix HT / stagiaire

1450€

### Objectifs pédagogiques

- Rappporter la théorie et l'importance des tests
- Mettre en œuvre des tests unitaires en Python
- Analyser le « code coverage » enPython.
- S'initier à l'intégration continue avec Jenkins
- Configurer Jenkins pour la gestion des projets
- Créer et gérer des projets Jenkins.
- Automatiser les tests avec Jenkins
- Intégrer l'analyse de qualité de code dans Jenkins
- Générer et intégrer la documentation avec Jenkins.
- Maîtriser les aspects avancés de Jenkins.

### Niveau requis

- Connaissance préalable des technologies Python est très vivement conseillée.

### Public concerné

- Développeurs, architectes, chefs de projets et ingénieurs qualité

### Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

### Conditions d'accès à la formation

Délai : 3 mois à 1 semaine avant le démarrage de la formation dans la limite des effectifs indiqués

### Moyens pédagogiques et techniques

Salles de formation (les personnes en situation de handicap peuvent avoir des besoins spécifiques pour suivre la formation. N'hésitez pas à nous contacter pour en discuter) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD

Documents supports de formation projetés  
Apports théoriques, étude de cas concrets et exercices

Mise à disposition en ligne de documents supports à la suite de la formation

### Dispositif de suivi de l'exécution de l'évaluation des résultats de la formation

Feuilles d'emargement (signature électronique privilégiée)

- Ajout de Doxygen dans le processus build
- Intégration de la documentation dans Jenkins

## **Aspects complémentaires**

- Gestion des pipelines
- Gestion des noeuds (master, slaves)
- La console script de Jenkins
- L'outil Jenkins-CLI