

Test Driven Development (TDD) en Python

Programme (Mis à jour le 08/04/2026)

Introduction aux tests unitaires automatisés (1h30)

- Pourquoi tester?
- Dette technique, régressions, maintenance
- Position des tests unitaires dans la pyramide des tests
- Différence entre tests unitaires, tests d'intégration, tests fonctionnels
- Tests en Python : unittest, pytest (choix privilégié)

Meilleures pratiques en matière de tests unitaires (2h)

- Caractéristiques d'un bon test : rapide, indépendant, déterministe, lisible
- Convention de nommage
- Structure Arrange / Act / Assert
- Tests paramétrés
- Cas nominaux vs cas limites
- Tests et exceptions
- Ce qu'il ne faut pas tester

Écrire des tests pour le TDD (2h30)

- Philosophie du TDD
- Différence entre tester après coup et concevoir par le test
- Écriture d'un test expressif
- Gérer les faux départs et les tests rouges

Le cycle Rouge – Vert – Réfacteur (1h)

- Rouge : écrire un test qui échoue
- Vert : coder le minimum fonctionnel
- Réfacteur : améliorer sans casser

Approfondissement du cycle Rouge-Vert-Réfacteur (2h)

- Identifier les signaux de refactoring
- Tests comme filet de sécurité
- Duplication vs abstraction
- Refactoring orienté design

La puissance des mocks (3h)

- Pourquoi utiliser des mocks?
- Différence entre Mock, Stub, Fake
- Outils Python : unittest.mock, pytest-mock
- Bonnes pratiques : ce qu'il est pertinent de mocker, les dangers du sur-mocking

TDD dans un projet réel (1h30)

- Intégration des tests dans un projet existant
- TDD et dette technique
- TDD en équipe
- Limites et écueils du TDD
- Quand ne pas faire du TDD

Référence

THIL3665

Durée

2 jours / 14 heures

Prix HT / stagiaire

1450€

Objectifs pédagogiques

- Comprendre les principes fondamentaux des tests unitaires automatisés
- Appliquer les bonnes pratiques de tests en Python
- Mettre en œuvre le TDD au quotidien
- Utiliser efficacement les mocks et stubs
- Refactorer du code en toute sécurité grâce aux tests

Niveau requis

- Bonne maîtrise des bases de Python
- Notions de programmation orientée objet

Public concerné

- Développeurs Python souhaitant maîtriser le Test Driven Development

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'émargement (signature électronique privilégiée)

Evaluations formatives et des acquis sous forme de questions orales et/ou écrites (QCM) et/ou mises en situation

Questionnaires de satisfaction (enquête électronique privilégiée)