

Devenir Testeur Logiciels

— Avec LOUVERS - TEST ACADEMY —

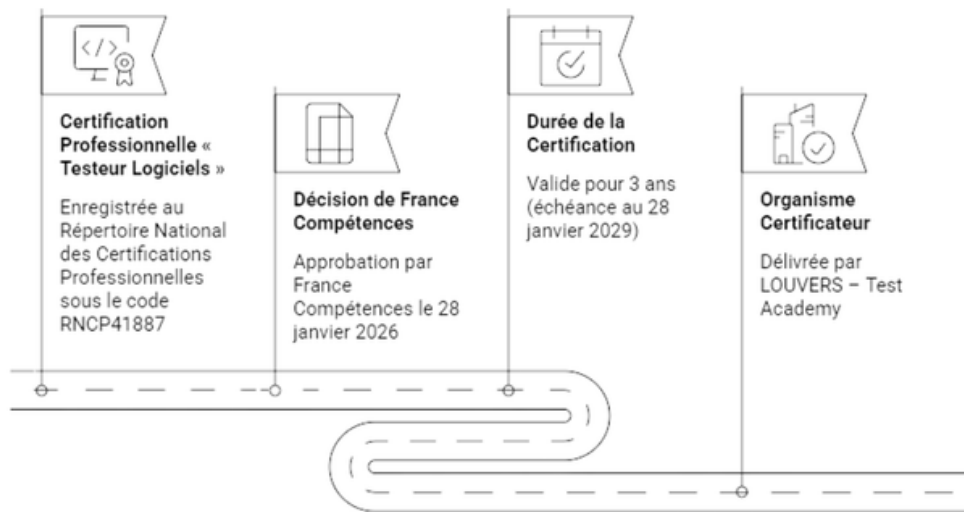
The logo consists of a dark blue square with the words "TEST ACADEMY" in white, uppercase, serif font, centered within the square.

TEST
ACADEMY

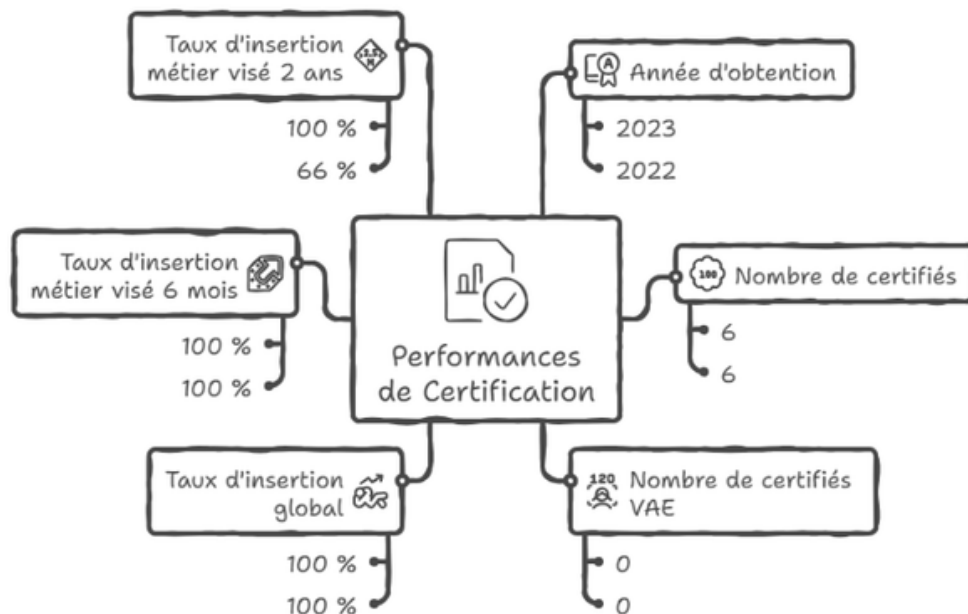
Programme de Formation

Certification

La certification RNCP41887 valide les compétences professionnelles du métier de testeur logiciels.



Le tableau ci-dessous présente les performances de la certification, incluant les taux d'insertion professionnelle et le nombre de certifiés.



Objectifs de la certification:

Bloc	Bloc 1	Bloc 2	Bloc 3
Description	Spécifier les exigences du logiciel et sélectionner les cas de test.	Implémenter et exécuter les cas de test, et restituer les résultats manuels.	Automatiser des scénarios de test et restituer un compte-rendu d'exécution.

01

Méthode

Cette section vous accompagne étape par étape pour assimiler une méthode de test complète et structurée.



Analyse de test

- 1.1.1 Les spécifications
- 1.1.2 Condition de test
- 1.1.3 Analyse de test
- 1.1.4 Structure de test
- 1.1.5 **Examen: QCM**

Conception des tests

- 1.2.1 Technique des partitions d'équivalence
- 1.2.2 Technique des valeurs limites
- 1.2.3 Technique des tables de décision
- 1.2.4 Technique des transitions
- 1.2.5 Concevoir un cas de test
- 1.2.6 **Examen: QCM**

Implémentation des tests

- 1.3.1 Fonctionnement d'une campagne de test
- 1.3.2 Création d'une campagne de test
- 1.3.3 Implémentation d'un cas de test
- 1.3.4 Niveau de granularité des cas de test
- 1.3.5 **Examen: QCM**

Exécution des tests

- 1.4.1 Comment exécuter les tests
- 1.4.2 rédiger un rapport de test
- 1.4.3 Comment réaliser un rapport de ...
- 1.4.4 Télécharger un modèle de Rapport de test
- 1.4.5 Modifier le diagramme Périmètre (Treemap)
- 1.4.6 Modifier le diagramme de conformité
- 1.4.7 Modifier le diagramme global (Score Qualité)
- 1.4.8 Modifier le diagramme d'avancement ...
- 1.4.9 Présenter les résultats de test
- 1.4.10 **Examen: QCM**

Validation du module

- 1.5.1 Livrable
- 1.5.2 Séance d'accompagnement individuel avec formateur



02

L'outillage

Cette section vous accompagne à la maîtrise de l'outil de gestion des cas de test: JIRA



Analyse de test

- 2.1.1 JIRA 1- Créer et Configurer le projet
- 2.1.2 JIRA 2 - Créer le périmètre fonctionnel

Conception des tests

- 2.2.1 JIRA 1 - Concevoir les cas de test
- 2.2.2 JIRA 2 - Vérifier la structure de test
- 2.2.3 JIRA 3 - Définir le Patrimoine de test
- 2.2.4 **Travaux Pratiques**

Implémentation des tests

- 2.3.1 JIRA 1 - Créer une procédure de test
- 2.3.2 JIRA 2 - Créer des suites de test
- 2.3.3 JIRA 3- Créer une campagne de test
- 2.3.4 **Travaux Pratiques**

Exécution des tests

- 2.4.1 JIRA 1 - Créer une exécution de test
- 2.4.2 JIRA 2 - Exécution d'une campagne de test
- 2.4.3 **Travaux Pratiques**
- 2.4.4 JIRA 3 - Exécuter un cas de test
- 2.4.5 JIRA 4 - Créer un rapport de défaut
- 2.4.6 **Travaux Pratiques**
- 2.4.7 JIRA 5 -Remplir les valeurs du rapport de test

Validation du module

- 2.5.1 Examen: QCM
- 2.5.2 Livrable
- 2.5.3 Séance d'accompagnement individuel avec formateur





Analyse de test

- 3.1.1** Déterminer la structure du périmètre.
- 3.1.2** Réaliser une revue d'une histoire utilisateur
- 3.1.3** Vérifier que l'histoire utilisateur est Ready
- 3.1.4** Mettre à jour la structure des tests
- 3.1.5** JIRA : Créer les EPICS
- 3.1.6** JIRA : Créer la User Story
- 3.1.7** JIRA : Ajouter les conditions de test
- 3.1.8** Correction

Conception des tests

- 3.2.1** Lister les cas de test de l'histoire
- 3.2.2** Concevoir les cas de test de l'histoire
- 3.2.3** Créer les cas de test
- 3.2.4** Correction

Implémentation des tests

- 3.3.1** Implémenter les cas de test de l'histoire utilisateur avec un niveau de granularité large
- 3.3.2** Implémenter les cas de test de l'histoire utilisateur avec un niveau de granularité fine
- 3.3.3** Implémenter les cas de test de l'histoire utilisateur avec un niveau de granularité moyenne
- 3.3.4** Préparer une campagne de test

Exécution des tests

- 3.4.1** Initialiser la première exécution de la campagne de test
- 3.4.2** Créer le rapport de test de l'Exécution 1
- 3.4.3** Initialiser l'exécution 2
- 3.4.4** Présenter les résultats de l'Exécution 2
- 3.4.5** Initialiser l'Exécution 3
- 3.4.6** Présenter les résultats de l'Exécution 3

Validation du module

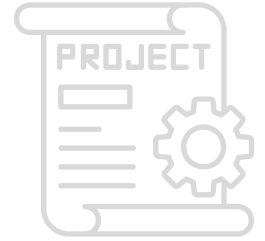
- 3.5.1** Examen: QCM
- 3.5.2** Livrable
- 3.5.3** Séance d'accompagnement individuel avec formateur



04

Le mode Projet

Présentation des modèles de gestion de projet : cascade, cycle en V et Scrum.



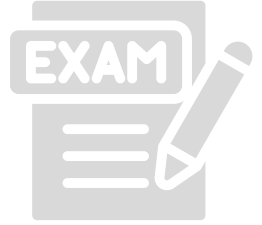
Processus de tests

- 7.1.1** L'indispensable rôle du Testeur
- 7.1.2** Le modèle en cascade
- 7.1.3** Le modèle en V
- 7.1.4** Le modèle SCRUM
- 7.1.5** Le processus de test

Validation du module

- 7.2.1** Examen: QCM
- 7.2.2** Livrable
- 7.2.3** Séance d'accompagnement individuel avec formateur





Préparation ISTQB

- 6.1.1** Fondamentaux des tests
- 6.1.2** Tester pendant le cycle de vie du développement logiciel
- 6.1.3** Tests statiques
- 6.1.4** Techniques de test
- 6.1.5** Gestion des tests
- 6.1.6** Outils de support aux tests

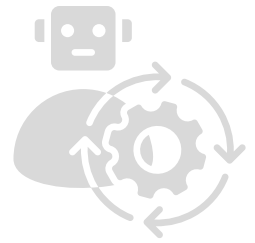
Entraînement

- 5.2.1** Examen Blanc A
- 5.2.2** Examen Blanc B
- 5.2.3** Examen Blanc C

Validation du module

- 5.3.1** Séance d'accompagnement individuel avec formateur
- 5.3.2** Session passage certification





Installation de l'environnement

- 4.1.1 Installer Python
- 4.1.2 Mettre à jour la variable d'environne...
- 4.1.3 Installer **RobotFramework**
- 4.1.4 Installer une librairie
- 4.1.5 Astuce - Acquérir les pilotes grâce à Webdrivermanager
- 4.1.6 Astuce - Installer **Selenium IDE**
- 4.1.7 Installer un éditeur - Visual St...
- 4.1.8 Installer une extension RobotFramework

Réalisation des tests

- | | |
|--|--|
| 4.2.1 Commencer par le test manuel | 4.2.8 Le Teardown |
| 4.2.2 Astuce - Enregistrer un test avec Selenium | 4.2.9 Les arguments |
| 4.2.3 Les cas de test | 4.2.10 Les tests négatifs |
| 4.2.4 Les librairies | 4.2.11 Les fichiers de ressources |
| 4.2.5 Les mots-clés | 4.2.12 Les templates |
| 4.2.6 Les variables | 4.2.13 Utiliser l'IA pour enrichir une suite |
| 4.2.7 Le Setup | |

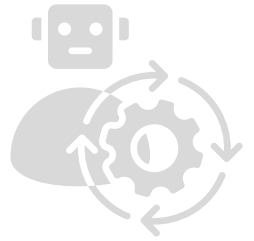
Pratique: Projet

- 4.2.1 Travaux pratiques
- 4.2.2 Corrections

Validation du module

- 4.3.1 Examen: QCM
- 4.3.2 Livrable
- 4.3.3 Séance d'accompagnement individuel avec formateur





API

- 5.1.1** Qu'est-ce qu'une API ?
- 5.1.2** Les principaux types d'API
- 5.1.3** API REST
- 5.1.4** Les requêtes : GET, POST, PUT, DELETE
- 5.1.5** Les codes retour
- 5.1.6** Découvrir JSON
- 5.1.7** Examen: QCM

Réalisation des tests

- 5.2.1** Réaliser une première requête simple
- 5.2.2** Consulter les spécifications d'une API
- 5.2.3** L'API ReqRes
- 5.2.4** GET : Consulter les spécifications
- 5.2.5** GET : Créer le cas de test
- 5.2.6** POST : Consulter les spécifications
- 5.2.7** POST : Créer le cas de test
- 5.2.8** DELETE : Consulter les spécifications
- 5.2.9** DELETE : Créer le cas de test
- 5.2.10** PUT : Consulter les spécifications
- 5.2.11** PUT : Créer le cas de test

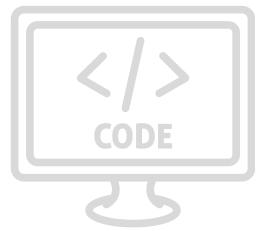
Optimiser avec l'IA

- 5.3.1** Télécharger et installer Antigravity
- 5.3.2** Formater le code des tests existants
- 5.3.3** Exécuter avec succès les tests API
- 5.3.4** Générer de nouveaux tests
- 5.3.5** Exécuter les nouveaux tests jusqu'au succès

Validation du module

- 5.4.1** Examen: QCM
- 5.4.2** Livrable
- 5.4.3** Séance d'accompagnement individuel avec formateur





Installation

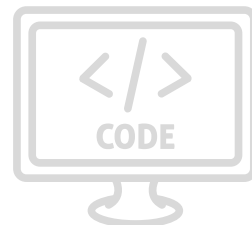
- 9.1.1 C'est quoi un programme ?
- 9.1.2 Installation de Python
- 9.1.3 Installation de Visual Studio Code (VSC)
- 9.1.4 Examen: QCM

Programmation déclarative

- 9.2.1 Leçon 1: Afficher un texte
- 9.2.2 Exercices + Correction
- 9.2.3 Leçon 2: Afficher un texte avec variable
- 9.2.4 Exercices + Correction
- 9.2.5 Leçon 3: Demander une donnée à l'utilisateur
- 9.2.6 Exercices + Correction
- 9.2.7 Leçon 4: Utiliser les commentaires
- 9.2.8 Exercices + Correction
- 9.2.9 Leçon 5: Les variables numériques
- 9.2.10 Exercices + Correction
- 9.2.11 Leçon 6: Convertir une chaîne de caractère
- 9.2.12 Exercices + Correction
- 9.2.13 Leçon 7, 8 et 9: Les conditions
- 9.2.14 Exercices + Correction
- 9.2.15 Leçon 10: Les exceptions
- 9.2.16 Exercices + Correction
- 9.2.17 Leçon 11: La boucle while
- 9.2.18 Exercices + Correction
- 9.2.19 Leçon 12: Utiliser le débogueur
- 9.2.20 Exercices + Correction
- 9.2.21 Leçon 13: Les fonctions
- 9.2.22 Exercices + Correction
- 9.2.23 Leçon 14: Variables globales et locales
- 9.2.24 Exercices + Correction
- 9.2.25 Leçon 15: La boucle For
- 9.2.26 Exercices + Correction
- 9.2.27 Leçon 16: Travaux pratiques
- 9.2.28 Exercices + Correction
- 9.2.29 Leçon 17: Les chaînes formatées
- 9.2.30 Exercices + Correction

Examen: QCM

Examen: QCM



Programmation orientée objet

- 9.3.1** Leçon 1 : Les Classes et les objets
- 9.3.2** Leçon 1 : Exercice
- 9.3.3** Leçon 1 : Corrigé
- 9.3.4** Leçon 2 : Les méthodes
- 9.3.5** Leçon 2 : Les méthodes - Exercice
- 9.3.6** Leçon 2 : Les méthodes - Corrigé
- 9.3.7** Leçon 3 : Les type de méthodes
- 9.3.8** Leçon 3 : Les type de méthodes - Exercice
- 9.3.9** Leçon 3 : Les type de méthodes - Corrigé
- 9.3.10** Leçon 4 : Visibilité & Encapsulation
- 9.3.11** Leçon 4 : Visibilité & Encapsulation - Exercice
- 9.3.12** Leçon 4 : Visibilité & Encapsulation - Corrigé

Examen: QCM

- 9.3.13** Leçon 5 : Héritage
- 9.3.14** Leçon 5 : Héritage - Exercices
- 9.3.15** Leçon 5 : Héritage - Corrigé
- 9.3.16** Leçon 6 : Héritage multiple
- 9.3.17** Héritage multiple - Corrigé
- 9.3.18** Leçon 7 - Composition
- 9.3.19** Composition - Corrigé
- 9.3.20** Leçon 8 - Polymorphisme
- 9.3.21** Polymorphisme _ Corrigé
- 9.3.22** Surcharge
- 9.3.23** Surcharge - Corrigé

Examen: QCM

Validation du module

- 9.4.1** Livrable
- 9.4.2** Séance d'accompagnement individuel avec formateur





Jenkins

- 8.1.1** Présentation de Jenkins
- 8.1.2** Pourquoi Installer Jenkins
- 8.1.3** Installer la bonne version de Java
- 8.1.4** Illustration - Installer Java
- 8.1.5** Installer le war Jenkins
- 8.1.6** Illustration - Installer Jenkins par WAR
- 8.1.7** Se familiariser avec le Tableau de...
- 8.1.8** Illustration - Découvrir le tableau ...
- 8.1.9** Créer le premier job de test
- 8.1.10** Illustration - Créer son premier job
- 8.1.11** Ajouter des paramètres au job
- 8.1.12** Illustration - Paramétrer un job
- 8.1.13** Lancer un job de test sur GitHub
- 8.1.14** Illustration - Lancer des tests sur Git

Validation du module

- 8.2.1** Livrable
- 8.2.2** Séance d'accompagnement individuel avec formateur





Introduction et installation

- 10.1.1** Introduction aux bases de données
- 10.1.2** Mise en place de l'environnement
- 10.1.3** Déploiement d'une base de donnée
- 10.1.4** Présentation du langage SQL
- 10.1.5** Examen QCM

Les jointures

- 10.3.1** Le principe de jointure
- 10.3.2** Les types de jointures
- 10.3.3** Les unions
- 10.3.4** Examen QCM

Les bases et les tables

- 10.5.1** Introduction partie V
- 10.5.2** Les types de données
- 10.5.3** Clé primaire clé étrangère
- 10.5.4** Création de table
- 10.5.5** Manipulation de table
- 10.5.6** Examen QCM

Validation du module

- 10.7.1** Examen: QCM
- 10.7.2** Livrable
- 10.7.3** Séance d'accompagnement individuel avec formateur

Les bases du langage SQL

- 10.2.1** Récupération des données - Clause
- 10.2.2** Les options de la Clause SELECT
- 10.2.3** Les fonctions d'agrégations
- 10.2.4** Les clauses avancées du SELECT
- 10.2.5** Examen QCM

Les fonctions et les sous-requêtes

- 10.4.1** Les fonctions mathématiques
- 10.4.2** Manipulation des dates et heures
- 10.4.3** Fonctions de chaînes de caractères
- 10.4.4** Les sous-requêtes
- 10.4.5** Examen QCM

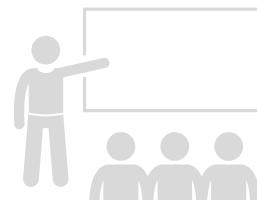
Les vues

- 10.6.1** Concept et création de vues
- 10.6.2** Modification et suppression de vues
- 10.6.3** Examen QCM



Le projet de fin de formation

Ce projet vous donne l'opportunité de mettre en pratique toutes vos connaissances en test logiciel.



Introduction

Intégrez **notre projet de fin de formation** spécialement conçu pour les personnes souhaitant se reconverter en tant que testeurs logiciels. Ce projet unique vous offre l'opportunité de mettre en pratique vos acquis dans **un environnement professionnel simulé** dans le cadre de l'application web Homey, une application de location saisonnière.

Simulation

Vous rejoindrez une équipe agile travaillant selon le modèle Scrum, une méthodologie moderne et flexible qui favorise la collaboration et l'adaptabilité. Cette immersion vous permettra de comprendre et de participer activement à toutes les étapes du développement logiciel, de la planification à la livraison continue de valeur.

Application pratique

Cette simulation vous offre une opportunité précieuse de mettre en pratique vos connaissances acquises pendant les modules précédents de la formation en test logiciel. En participant à ce projet, vous allez :

- Appliquer des techniques de test statique, des tests dynamiques manuels et des tests automatisés pour garantir la qualité du logiciel.
- Développer votre capacité à travailler en autonomie ou en binôme, en fonction des exigences du projet.

Validation du module

Soutenance de fin de formation où vous présenterez vos travaux, justifierez vos choix de test, et démontrerez vos compétences en matière de conception et d'exécution de cas de test manuels et automatisés.

