

# Formation - Clean Code

Cette formation a pour objectif de rendre le code des tests automatisés plus maintenable et plus fiable. Vous apprendrez les idées et techniques qui permettent d'écrire des ensembles de tests plus rapides, plus faciles à comprendre et qui n'échouent plus pour de mauvaises raisons. Vos tests vous permettront enfin de réellement gagner du temps, d'avoir une meilleure qualité logicielle et surtout un meilleur retour sur investissement.

## Objectifs :

### Cette formation a pour objectif de :

- Clarifier les objectifs de la formation pour aligner les attentes.
- Définir et expliquer les concepts fondamentaux du Clean Code et leur importance dans le développement logiciel.
- Discuter des concepts de couplage et de cohésion comme indicateurs de la structure logicielle.
- Introduire et expliquer les principaux acronymes utilisés dans les pratiques de développement de logiciels, tels que KISS, YAGNI, DRY/WET, et d'autres.
- Fournir des stratégies pour détecter et corriger ces odeurs de code dans le cadre du processus de développement.
- Souligner l'importance de ces pratiques pour le développement de logiciels de haute qualité, maintenables et extensibles.
- Expliquer l'importance de ces principes pour la création de systèmes logiciels flexibles, maintenables et évolutifs.
- Examiner les méthodes pour exprimer clairement l'intention métier à travers les tests, et comment ces tests contribuent à une meilleure compréhension du code.

### À l'issue de cette formation chaque apprenant sera capable de :

- Comprendre le fonctionnement de la formation, y compris comment le contenu sera présenté et évalué.
- Appliquer les principes fondamentaux du Clean Code dans la rédaction et la révision de code au sein du Robot Framework.
- Utiliser des métriques spécifiques pour évaluer la qualité et la maintenabilité du code.
- Appliquer ces principes dans la rédaction de code pour améliorer sa lisibilité, sa maintenabilité et son efficacité.
- Appliquer des stratégies de refactoring pour résoudre les problèmes identifiés par les odeurs de code, améliorant ainsi la maintenabilité et la clarté du code.
- Gérer correctement les exceptions et rédiger des commentaires qui ajoutent de la valeur au code sans l'encombrer.
- Reconnaître et éviter les pièges communs en programmation orientée objet, en se concentrant sur la création de code propre et maintenable.
- Utiliser les tests pour documenter clairement les fonctionnalités et les intentions métier du code, facilitant la maintenance et l'évolution du logiciel.

## Description :

### Cette formation vous permettra de :

- D'adopter des méthodes agiles a optimisé la construction de logiciels en adéquation avec les besoins réels, en améliorant simultanément le processus de livraison. Le Clean Code constitue un ensemble de pratiques techniques qui complète ces méthodes, assurant ainsi la longévité du logiciel construit.
- Un code propre est en effet le critère essentiel pour un logiciel robuste (avec un coût d'entretien réduit) et évolutif (qui peut être adapté aux nouveaux besoins).

## Méthode et moyens pédagogiques utilisés :

- Cours vidéos
- QCM
- Travaux pratiques
- Connexion internet
- Adresse email
- PC permettant de travailler à distance / domicile
- Casque audio ou écouteurs
- Son

## Prérequis :

Cette formation ne nécessite pas de prérequis.

## Evaluation:

En amont : Questionnaire d'évaluation à l'entrée en formation

En cours : Quiz d'activation des connaissances

En fin de formation : Quiz d'évaluation finale accompagné d'une étude de cas

## Certification :

Remise de :

- Une attestation de suivi de formation
- Certificat de réussite UpSkill4IT (si les conditions sont remplies)

## Délais d'accès :

Théorie en VOD (video on demand) : accès immédiat après la signature du contrat

## Durée de formation :

La partie théorique sur Clean Code en e-learning dure 4h sous format VOD et peut être réalisée en 1 journée (**3 premières sections**)

## Accès et format :

Théorie (sections de 1 à 3) :

Formation est proposée sous format VOD

Pratique (sections de 4 à 9) : en VC (visioconférence) : disponible via les ateliers et non incluse dans le tarif, date proposée sous 48h

## Public concerné :

- Cette formation sur Clean Code s'adresse à tous les développeurs, quel que soit leur niveau ou les technologies qu'ils utilisent pour développer des sites ou des logiciels

## Actualisation :

Date 16/06/2023 – V:001

## Présentation du formateur Clean Code Edouard Mangel

Formateur, coach Craftsman et développeur

J'ai travaillé dans différents contextes, pour le compte de sociétés de service, ou pour des clients finaux. J'ai pu intervenir comme développeur sur de nombreux projets, dans des domaines variés comme l'industrie, la recherche médicale, les collectivités territoriales, la banque ou encore la gestion de trésorerie.

J'ai pu affiner ma pédagogie depuis 10 ans en tant que coach sportif, et je suis également formateur en développement informatique depuis 2016. J'interviens auprès d'entreprises et de particuliers, mais également au sein de différentes formations parmi lesquelles :

- La formation ingénieur en alternance au CNAM de Strasbourg,
- Le titre d'ingénieur en cours du soir du CNAM Strasbourg
- La formation bac + 5 en développement informatique du CESI.

## Accessibilité :

Si vous avez des difficultés ou êtes en situation de handicap veuillez nous contacter à l'adresse mail suivante : [contact@upskill4it.com](mailto:contact@upskill4it.com)



# Sommaire 1/1

## Section 1 – Introduction au Clean Code

1. Leçon 1 – Présentation du formateur
2. Leçon 2 – Présentation des objectifs de la formation
3. Leçon 3 – Présentation des modalités de la formation
4. Leçon 4 – Présentation des sources
5. Leçon 5 – Présentation du sommaire

## Section 2 – Principe du Clean Code de Robot Framework

1. Leçon 1 – Définitions : Axes de qualité d'un logiciel
2. Leçon 2 – Définitions : Le code sale, difficile à lire
3. Leçon 3 – Définitions : Le code sale, dur à comprendre
4. Leçon 4 – Définitions : Le code sale, difficile à modifier
5. Leçon 5 – Définitions : L'avis des experts
6. Leçon 6 – Définitions : Une définition de clean code
7. Leçon 7 – Définitions : Un code facile à lire
8. Leçon 8 – Définitions : Un code facile à comprendre
9. Leçon 9 – Définitions : Un code facile à modifier
10. Leçon 10 – Pour quoi faire du Clean Code : Le coût du code sale
11. Leçon 11 – Pour quoi faire du Clean Code : Les types de mémoire
12. Leçon 12 – Pour quoi faire du Clean Code : Les types de charge
13. Leçon 13 – Pour quoi faire du Clean Code : Convaincre les collègues
14. Leçon 14 – Pour quoi faire du Clean Code : Convaincre son manager
15. Leçon 15 – Pour quoi faire du Clean Code : Conclusion

## Section 3 – Métriques de qualité logicielle, maintenabilité

1. Leçon 1 – Couplage
2. Leçon 2 – Cohésion
3. Leçon 3 – Duplication
4. Leçon 4 – Complexité cyclomatique
5. Leçon 5 – Couverture de code

À partir de la Section 4 la pratique est en VC (visioconférence) : *disponible via les ateliers et non incluse dans le tarif*

## Section 4 – Acronymes

1. Leçon 1 – KISS
2. Leçon 2 – YAGNI
3. Leçon 3 – DRY/WET
4. Leçon 4 – Loi de Déméter
5. Leçon 5 – Tell don't ask (Hollywood Principle)
6. Leçon 6 – SLAP (Single Level of Abstraction Principle)
7. Leçon 7 – SoC Separation of Concerns

## Section 5 – Détecter les problèmes avec les Code smells

1. Leçon 1 – Bloaters
2. Leçon 2 – Couplers
3. Leçon 3 – Dispensables
4. Leçon 4 – Object orientation abusers
5. Leçon 5 – Change preventers

## Section 6 – Bonnes pratiques de design du code

1. Leçon 1 – Généralités
2. Leçon 2 – Variables
3. Leçon 3 – Les expressions conditionnelles
4. Leçon 4 – Comment écrire des fonctions "propres" ?
5. Leçon 5 – Comment écrire des classes "propres" ?
6. Leçon 6 – Comment gérer proprement les exceptions ?
7. Leçon 7 – Comment écrire un commentaire a forte valeur ajoutée ?

## Section 7 – Bonnes pratiques POO

1. Leçon 1 – SOLID
2. Leçon 2 – CUPID
3. Leçon 3 – Object Calisthenics

## Section 8 – Bonnes pratiques tests unitaires & intégration (Intro)

- Leçon 1 – Pas de clean code sans tests automatisés
- Leçon 2 – Comment véhiculer l'intention métier par les tests ?
- Leçon 3 – Principes FIRST
- Leçon 4 – Test Desiderata par Kent Beck

## Section 9 – Intro au TDD avec démo par kata : 1h de live coding

- Leçon 1 – Qu'est-ce que le TDD ?
- Leçon 2 – Pourquoi faire du TDD ?
- Leçon 3 – Quelques astuces de TDD
- Leçon 4 – Le TPP et live coding

Quiz