L'AKADÉMIE DU DIGITAL · TRAINING PROGRAMME Qualiopi certification being prepared

Programmation — Fondamentaux multi-langages

FOR-DEV-M1-A · Development, DevOps & Data

Acquérir les fondamentaux de la programmation, transposables à tout langage

DURATION
35 heures — 5 jours

PRICE
2?450 € HT

AUDIENCE
Schools & training centres

ACCESS
Access within 2 working days after validation of needs and prerequisites.

Tarif établi sur la base de 35h à 70 € HT/h. Parcours dédié aux écoles et centres de formation, établi sur devis ; tarif indicatif et négociable selon le volume et le contexte.

Le module alterne séquences théoriques et ateliers pratiques tout au long de chaque journée. Il part de la logique algorithmique pour aller jusqu'à la réalisation d'un projet structuré, en s'appuyant sur deux langages (Python et JavaScript) et les outils du développeur.

Learning objectives

- Comprendre la logique algorithmique et les structures fondamentales (variables, conditions, boucles, fonctions)
- Manipuler les structures de données courantes (tableaux, listes, dictionnaires, objets)
- Lire, écrire et déboguer du code dans au moins deux langages (Python et JavaScript)
- Comprendre les paradigmes (procédural, orienté objet, fonctionnel) et savoir les distinguer
- Utiliser les outils du développeur : éditeur/IDE, terminal, gestion de versions (Git)
- Concevoir et réaliser un petit programme structuré de bout en bout

Target audience

- Étudiants et apprenants des écoles et centres de formation partenaires
- Personnes en reconversion vers les métiers du développement
- Profils techniques débutants souhaitant des bases solides et transposables

Prerequisites

Aucun prérequis de programmation. Une aisance informatique solide est requise (manipulation de fichiers, installation de logiciels, logique) ainsi qu'une appétence pour la résolution de problèmes. Ces éléments sont validés par un formulaire en ligne et / ou un entretien préalable.

- Savoir utiliser un ordinateur de façon avancée (fichiers, installation de logiciels, navigation système).

- Disposer d'une logique de résolution de problèmes.
- Être à l'aise avec l'anglais technique de base (un atout pour la documentation).
- Aucune compétence préalable en programmation n'est exigée.

Programme

Jour 1 — Logique algorithmique & premiers pas

Programmer : c'est quoi ?	1h00	Algorithme, pseudo-code, variables et types de données. Exposé interactif, support visuel, exemples de la vie courante traduits en algorithmes.	Questions orales de compréhension.
Premiers scripts	1h30	Manipuler variables, types et entrées/sorties. TP guidé pas-à-pas, environnement préparé, exercices progressifs.	Exercices pratiques validés.
Conditions & logique booléenne	1h00	Tests, opérateurs, branchements conditionnels. Exposé, schémas de logigrammes, exemples commentés.	QCM court sur la logique.
Exercices de conditions	2h30	Cas pratiques mobilisant les conditions. TP individuels et en binôme, correction collective.	Exercices corrigés ; auto-évaluation.

Jour 2 — Boucles, fonctions & structures de données

Boucles & fonctions	1h15	Itérations (for, while), définition et réutilisation de fonctions. Exposé, exemples, démonstration en direct.	Questions orales ; QCM.
Exercices algorithmiques	2h15	Itérations et fonctions réutilisables. TP guidé, défis algorithmiques progressifs.	Exercices validés.
Structures de données	1h00	Tableaux, listes, dictionnaires : usages et choix. Exposé, tableau comparatif, exemples.	QCM sur les structures.
Manipulation de collections	2h30	Parcourir, filtrer et transformer des collections. TP, jeux de données fournis, correction collective.	Exercices validés.

Jour 3 — Deux langages en parallèle (Python & JavaScript)

Python vs JavaScript	1h15	Similitudes et différences syntaxiques ; quand utiliser quoi. Exposé comparatif, exemples côte à côte.	Questions orales ; QCM.
Porter un programme	2h30	Transposer un même programme de Python vers JavaScript. TP guidé, tableau de correspondance, accompagnement.	Programme porté et fonctionnel.
Lecture de code & débogage	1h00	Lire du code inconnu, comprendre les erreurs. Exposé, typologie des erreurs, méthode de débogage.	QCM sur les types d'erreurs.
Chasse aux bugs	2h15	Identifier et corriger des bugs sur du code fourni. TP de débogage, code volontairement bugué, revue collective.	Bugs corrigés validés.

Jour 4 — Paradigmes & outils du développeur

Paradigmes de programmation	1h15	Procédural, orienté objet, fonctionnel (notions et usages). Exposé, exemples illustrant chaque paradigme.	Questions orales ; QCM.
Refactoring	2h15	Refactoriser un script en fonctions puis en objets. TP guidé, code de départ fourni, accompagnement.	Refactoring validé.
Outils du développeur & Git	1h00	IDE, terminal, gestion de versions avec Git. Exposé, démonstration, schéma du workflow Git.	QCM sur Git et les outils.
Initiation Git	2h30	Initier un dépôt, commits, branches. TP guidé, dépôt d'exemple, scénarios de versionnage.	Manipulations Git validées.

Jour 5 — Projet de synthèse

Concevoir un programme	1h00	Méthode de conception : découpage, choix, tests. Exposé, méthode de cadrage, présentation du projet fil rouge.	Cadrage du projet validé.
Réalisation du projet	3h00	Réaliser un projet fil rouge structuré. TP projet, accompagnement individualisé, points d'étape.	Avancement du projet.
Finalisation, tests & débogage	1h30	Finaliser, tester et déboguer le projet. TP, tests, revue de code entre pairs.	Projet fonctionnel.
Restitution & évaluation finale	1h00	Présenter le projet ; évaluation sommative. Présentations, revue collective, QCM final, enquête de satisfaction.	Évaluation sommative : projet + QCM final.

Teaching methods and resources

In-person training at our premises or yours, in small groups. A mix of theory, hands-on workshops and practical scenarios.

Required equipment :

8 Go de Ram (16 Go recommandés)

- Les stagiaires doivent disposer d'un PC avec le Wifi et 8 Go de RAM (16 Go recommandés).
- Effectif adapté au format école / centre de formation.
- Parcours dédié aux écoles et centres de formation, établi sur devis.

Assessment methods

Learning outcomes are assessed during and at the end of the training; an attendance certificate is given to every learner.

Accessibility

Training accessible to people with disabilities. Contact us before registering so we can study the adjustments needed.

Funding

Quote or training agreement on request. Qualiopi certification is being prepared: eligibility for OPCO funding will open once it is obtained.

Certification audit planned for T1 2027. No OPCO eligibility before it is obtained.