FONDAMENTAUX

Algorithmique et logique de programmation

Apprenez à raisonner comme un développeur avant d'apprendre un langage : décomposer un problème, variables, conditions, boucles, fonctions et tableaux, jusqu'à un premier script Python.

DURÉE	TARIF	NIVEAU	LANGUE	GROUPE	FORMAT
2j (14h)	**650 € / participant** Inter-entreprise. Intra : sur devis, au plus 650 € / participant	**Débutant**	**Français**	**2-6**	**Formation** À distance

1 PUBLIC VISÉ

- Grands débutants et personnes en reconversion qui veulent apprendre la logique de programmation avant un premier langage
- Étudiants qui veulent consolider leur logique avant d'apprendre un langage

2 PRÉREQUIS

- Utiliser un ordinateur : clavier, navigateur web, créer et enregistrer un fichier
- Aucune connaissance en programmation n'est requise

3 OBJECTIFS PÉDAGOGIQUES

- Décomposer un problème en entrées, traitement et sorties, et l'écrire en pseudocode avec des variables typées et des opérateurs
- Écrire en pseudocode des conditions et des boucles, y compris imbriquées, qui couvrent tous les cas prévus
- Découper un algorithme en fonctions avec paramètres et valeur de retour
- Parcourir un tableau pour y rechercher un élément, compter des valeurs et le trier par une méthode simple
- Traduire un algorithme écrit en pseudocode en un script Python qui s'exécute

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) : Décomposer un problème, conditions et boucles

Module 1 : Décomposer un problème et manipuler des variables (3h30)

Qu'est-ce qu'un algorithme (1h30)

- Un algorithme dans la vie courante : recette, itinéraire, mode d'emploi
- Algorithme, programme et langage : ce qui les distingue
- Les étapes de la résolution d'un problème : entrée, traitement, sortie
- Le pseudocode et le schéma (logigramme) : deux façons d'écrire un algorithme
- Exercice d'entraînement : décrire en étapes ordonnées une tâche du quotidien

Variables, types et opérateurs (2h)

- Variables : déclaration, affectation, nommage
- Types de données : entiers, décimaux, chaînes, booléens
- Opérateurs arithmétiques, de comparaison et logiques ; priorité des opérations
- Suivre l'exécution pas à pas avec un tableau de valeurs
- TP évalué : écrire en pseudocode le calcul d'un prix TTC avec remise, en partant de l'énoncé : entrées, traitement, sorties, variables typées

Module 2 : Écrire des conditions et des boucles (3h30)

Conditions (1h30)

- Si, sinon, sinon si
- Conditions combinées et conditions imbriquées
- Les cas limites : la valeur juste au bord, la saisie vide
- Erreurs de logique fréquentes
- Exercice d'entraînement : valider une saisie et choisir un tarif selon l'âge

Boucles et boucles imbriquées (2h)

- Boucle tant que : répéter tant qu'une condition est vraie
- Boucle pour : répéter un nombre connu de fois
- Compter, accumuler, chercher : les trois usages courants
- Boucles imbriquées : une boucle dans une boucle
- TP évalué : afficher une table de multiplication complète, puis compter les valeurs qui dépassent un seuil saisi

Jour 2 (7h) : Fonctions, tableaux et premier script Python

Module 3 : Fonctions et tableaux (3h30)

Fonctions (1h30)

- Définir et appeler une fonction
- Paramètres et valeur de retour
- Découper un problème en fonctions : éviter les répétitions
- Exercice d'entraînement : regrouper en fonctions un algorithme fourni qui répète trois fois le même calcul

Tableaux : parcourir, rechercher, trier (2h)

- Notion de tableau : indices, taille, parcours
- Rechercher un élément, compter, trouver le minimum et le maximum
- Trier un tableau par une méthode simple (tri par sélection)
- TP évalué : à partir d'un tableau de notes, écrire les fonctions qui calculent la moyenne, trouvent la meilleure note, comptent les notes au-dessus de la moyenne et trient le tableau

Module 4 : Du pseudocode à Python : projet de synthèse, évalué (3h30)

Traduire un algorithme en Python (1h15)

- Écrire et exécuter un premier script Python dans VS Code
- Correspondance entre pseudocode et Python : variables, conditions, boucles, fonctions, listes
- Lire un message d'erreur de Python
- Exercice d'entraînement : traduire en Python l'algorithme du TP du module 1

Projet de synthèse (2h15)

- Projet de synthèse, réalisé individuellement : à partir d'un énoncé (gestion des notes d'une classe ou d'un stock), écrire l'algorithme en pseudocode avec des fonctions et un tableau, puis le traduire en script Python qui donne les résultats attendus sur le jeu d'essai fourni
- Retour individuel du formateur

Le contenu peut être ajusté selon le niveau, les prérequis et les besoins des participants.

5 COMPÉTENCES VISÉES

- Analyser un problème et le décomposer en étapes algorithmiques
- Écrire des structures de contrôle : conditions et boucles
- Structurer un algorithme en fonctions et traiter des données en tableaux
- Traduire un algorithme dans un premier langage de programmation

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée à distance, en classe virtuelle. Le formateur alterne entre méthode démonstrative, méthode interrogative (questionnement pour faire émerger la logique) et méthode active (exercices d'entraînement, TP évalués et projet de synthèse).

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Fiches d'exercices progressives avec corrigés
- Outil de visualisation de l'exécution d'un algorithme (Flowgorithm ou équivalent)
- Python 3.12 ou plus récent et VS Code, pour le module 4
- Tableau blanc collaboratif en ligne (Excalidraw ou équivalent) pour les schémas
- Classe virtuelle : visioconférence (Microsoft Teams ou équivalent), partage d'écran, chat en direct
- Supports remis par e-mail et sur la plateforme pédagogique LaPolaris (ressources, émargement)

Chaque participant suit la formation depuis son ordinateur, avec un accès à Internet, un micro et des haut-parleurs ou un casque. En formation intra-entreprise, l'entreprise s'assure que chaque participant dispose de ce matériel et des logiciels nécessaires, conformément aux prérequis indiqués dans le programme et aux moyens prévus dans la convention.

8 MODALITÉS D'ÉVALUATION

- Questionnaire de positionnement individuel renseigné par chaque participant avant le démarrage de la formation, analysé par le formateur et donnant lieu, le cas échéant, à un ajustement du déroulé
- Exercices d'entraînement réalisés par chaque participant sur son poste, corrigés en séance
- Évaluation des acquis par les TP évalués, réalisés individuellement sur poste : un par module, puis le projet de synthèse décrit dans le dernier module du programme ; chaque objectif est évalué dans au moins un TP
- Grille de suivi renseignée par le formateur pour chaque participant et pour chacun des objectifs de l'action, au regard des critères d'évaluation ci-dessous

- Questionnaire d’appréciation renseigné par les participants en clôture de formation, et enquête à froid dans un délai de trois à six mois

9 CRITÈRES D'ÉVALUATION

- Problème découpé en entrées, traitement et sorties ; pseudocode avec des variables nommées clairement, typées, et des opérateurs corrects
- Conditions qui couvrent tous les cas prévus, cas limites compris ; boucles qui s’arrêtent ; boucles imbriquées qui donnent le résultat attendu
- Fonctions avec paramètres et valeur de retour, appelées par l’algorithme principal, sans bloc de code répété
- Tableau parcouru du premier au dernier élément ; recherche, comptage et tri qui donnent le résultat attendu sur le jeu d’essai fourni
- Script Python qui s’exécute sans erreur et donne les mêmes résultats que le pseudocode sur le jeu d’essai

10 INDICATEURS DE RÉSULTATS

Les indicateurs de résultats de cette formation (taux de satisfaction et taux de réussite) seront publiés sur sa fiche en ligne à l’issue des premières sessions réalisées.

11 MODALITÉS DE VALIDATION

Une attestation de fin de formation est remise à chaque participant à l’issue du parcours. Elle mentionne **l’intitulé** et la **nature de l’action**, sa **durée** en heures, sa **période de réalisation**, **les objectifs de l’action** et le **niveau d’atteinte** de chacun d’eux, exprimé en trois valeurs : **atteint**, **partiellement atteint**, **non atteint**. **Une attestation d’assiduité**, établie à partir des émargements, est délivrée dans les mêmes conditions.

Formation non certifiante : elle ne délivre pas de certification enregistrée au RNCP ou au Répertoire spécifique.

12 SUIVI ET ACCOMPAGNEMENT

- Émargement électronique signé par chaque participant et par le formateur à chaque demi-journée
- Traçabilité des activités pédagogiques réalisées
- Attestation d’assiduité délivrée en fin de formation
- Suivi individuel via les QCM de fin de journée, les exercices corrigés et les TP évalués

13 CONDITIONS D'ACCÈS

Formation accessible sans prérequis administratif particulier. Cette formation n’est pas éligible aux financements publics ou mutualisés, notamment au compte personnel de formation et à une prise en charge par un opérateur de compétences.

14 DÉLAIS D'ACCÈS

Inscription possible jusqu’à 15 jours ouvrés avant le début de la session pour permettre la préparation logistique et, si besoin, l’adaptation du contenu au contexte client. Pour toute demande spécifique, nous contacter directement.

ACCESSIBILITÉ · HANDICAP

Nos formations peuvent être accessibles aux personnes en situation de handicap. La question d’un besoin d’aménagement (matériel, temporel ou pédagogique) est posée lors de l’analyse du besoin et du questionnaire de positionnement ; vous pouvez aussi nous le signaler à tout moment, y compris pendant la formation, afin d’étudier les adaptations possibles. Référent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr