FONDAMENTAUX

Python pour débutants : premiers pas en programmation

Apprenez à programmer avec Python : variables, conditions et boucles, listes et dictionnaires, fonctions et modules, fichiers et débogage, jusqu'à un script d'automatisation.

DURÉE	TARIF	NIVEAU	LANGUE	GROUPE	FORMAT
3j (21h)	**975 € / participant** Inter-entreprise. Intra : sur devis, au plus 975 € / participant	**Débutant**	**Français**	**3-8**	**Formation** À distance

1 PUBLIC VISÉ

- Débutants et personnes en reconversion qui veulent apprendre à programmer
- Professionnels qui veulent automatiser des tâches répétitives avec des scripts

2 PRÉREQUIS

- Créer un dossier, y enregistrer un fichier et le retrouver
- Aucune connaissance en programmation n'est requise

3 OBJECTIFS PÉDAGOGIQUES

- Écrire et exécuter depuis le terminal un script Python qui utilise des variables, les types natifs (entiers, décimaux, chaînes, booléens) et les opérateurs
- Écrire des conditions (if, elif, else) et des boucles (for, while)
- Manipuler des listes, des tuples et des dictionnaires : création, parcours, méthodes courantes
- Écrire des fonctions avec paramètres et valeur de retour, les regrouper dans son propre module et importer les modules standards utiles (os, math, random, datetime)
- Lire et écrire des fichiers texte et CSV, en gérant les erreurs avec try / except
- Lire un traceback et déboguer un script pas à pas dans VS Code

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) : Fondamentaux du langage

Module 1 : Installer Python et écrire ses premiers scripts (3h30)

Environnement et premiers pas (1h30)

- Installer Python et VS Code
- Premier script : print, exécution dans le terminal
- L'interpréteur interactif
- Exercice d'entraînement : installer Python et VS Code, puis écrire et exécuter un premier script depuis le terminal

Variables, types et opérateurs (2h)

- Types natifs : int, float, str, bool
- Affectation, conversion, f-strings
- Opérateurs arithmétiques, de comparaison et logiques
- TP évalué : écrire et exécuter depuis le terminal un script qui calcule un prix TTC à partir d'un prix et d'un taux donnés : variables, conversion de types, opérateurs et affichage avec une f-string

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

Les conditions (1h30)

- if / elif / else
- Exercice d'entraînement : exercices progressifs sur les conditions

Les boucles (2h)

- Boucles for et while
- break, continue, range
- TP évalué : écrire un script qui parcourt une suite de nombres avec une boucle, classe chaque nombre avec if, elif et else, et s'arrête sur une valeur donnée avec break

Jour 2 (7h) : Structures de données et fonctions

Module 3 : Listes, tuples et dictionnaires (3h30)

Listes et tuples (1h30)

- Listes : création, indexation, méthodes (append, sort, pop…)
- Tuples et leur immutabilité
- Exercice d'entraînement : gérer une liste de courses : ajout, tri, suppression

Dictionnaires (2h)

- Dictionnaires : clés et valeurs, parcours, méthodes
- Compréhensions de liste : introduction
- TP évalué : construire un carnet de contacts sous forme de liste de dictionnaires, le parcourir et le trier, puis en extraire une sélection avec une compréhension de liste

Module 4 : Écrire des fonctions et les regrouper en modules (3h30)

Fonctions et portée des variables (1h30)

- Définir et appeler une fonction
- Paramètres, valeurs par défaut, return
- Portée locale et portée globale
- Bonnes pratiques de nommage et de lisibilité
- Exercice d'entraînement : transformer un script répétitif en fonctions

Modules et imports (2h)

- Importer des modules standards : os, math, random, datetime
- Créer son propre module
- pip et les packages tiers : introduction
- TP évalué : regrouper les fonctions du carnet de contacts dans son propre module (paramètres, valeurs de retour), puis l'importer dans un script principal qui utilise datetime

Jour 3 (7h) : Fichiers, débogage et projet

Module 5 : Fichiers, erreurs et débogage (3h)

Lire et écrire des fichiers (1h30)

- Lire et écrire un fichier texte
- Gérer les erreurs avec try / except
- Manipuler des fichiers CSV avec le module csv
- Exercice d'entraînement : lire un fichier CSV et afficher ses lignes

Lire un traceback et déboguer (1h30)

- Lire et comprendre un traceback
- Déboguer pas à pas dans VS Code
- Principes de code propre : nommage, commentaires, ne pas se répéter (DRY)
- TP évalué : corriger un script fourni qui s'arrête en lisant un fichier CSV : lire le traceback, déboguer pas à pas dans VS Code, ajouter la gestion des erreurs avec try / except et écrire le résultat dans un fichier

Module 6 : Projet de synthèse, évalué (4h)

Réalisation du script d'automatisation (2h30)

- Réalisation individuelle d'un script d'automatisation : lire un fichier CSV, filtrer ses lignes et exporter le résultat
- Script organisé en fonctions dans un module, avec conditions, boucles, listes et dictionnaires, et erreurs gérées avec try / except

Revue et retour (1h30)

- Revue de code en groupe
- Retour individuel du formateur

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

5 COMPÉTENCES VISÉES

- Écrire un script Python lisible avec des variables, des conditions et des boucles
- Organiser les données et le code d'un script : listes, dictionnaires, fonctions et modules
- Traiter des fichiers dans un script Python et le déboguer

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée à distance, en classe virtuelle. Le formateur alterne entre méthode démonstrative, méthode interrogative (questionnement pour ancrer les notions) 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
- Environnement de développement : Python 3.x + VS Code + terminal
- Exercices pratiques progressifs avec corrigés pour chaque module
- Exemples de code et structure de départ du projet de synthèse dans le support de cours
- 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

- Script exécuté sans erreur depuis le terminal ; types adaptés aux données, conversions explicites, résultat affiché avec une f-string
- Conditions qui couvrent tous les cas prévus ; boucles qui s'arrêtent comme prévu
- Structure de données adaptée (liste, tuple ou dictionnaire), parcourue et modifiée avec les méthodes courantes
- Fonctions nommées clairement, avec paramètres et valeur de retour, regroupées dans un module importé par le script principal
- Fichier texte ou CSV lu et écrit ; erreurs de lecture gérées avec try / except, sans arrêt du script
- Erreur localisée à partir du traceback et corrigée à l'aide du débogueur de VS Code

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