

PROGRAMME DE FORMATION

PYTHON DÉBUTANT – PRÉPARATION CERTIFICATION ENI

FICHE SYNTHÉTIQUE DE LA FORMATION

Intitulé	Python Débutant – Préparation Certification ENI
Durée	4 jours (28 heures)
Référence	PY-DEB-4J-ENI
Tarif	2 200 € HT / personne
Modalité	Présentiel ou distanciel
Public	Débutants, reconversion, développeurs, administrateurs système, data analysts
Prérequis	Aucun en programmation. Aisance informatique de base.
Certification	RS6701 – IT Langage Python (Éditions ENI) – Score : 700/1000
Lieu	255 boulevard Pereire, 75017 Paris ou à distance (Zoom)

1. Public visé

Cette formation s'adresse aux débutants complets, adultes en reconversion professionnelle, développeurs souhaitant ajouter Python à leurs compétences, administrateurs système et réseau, ainsi qu'aux professionnels du traitement de la donnée.

2. Prérequis

- Aucune connaissance préalable en programmation n'est requise
- Aisance minimale avec l'outil informatique (naviguer dans des dossiers, utiliser un navigateur)

Un test de positionnement est proposé avant l'entrée en formation afin de vérifier l'adéquation du parcours avec le profil du stagiaire.

3. Objectifs pédagogiques

À l'issue de cette formation, le stagiaire sera capable de :

- **Écrire** des scripts Python structurés en maîtrisant variables, types et conversions (C1)
- **Manipuler** les structures de données composites : listes, tuples, dictionnaires (C2)
- **Mettre en œuvre** les structures de contrôle conditionnelles et itératives (C3)
- **Utiliser** les entrées/sorties standards : print(), input(), f-string, format() (C4)
- **Exploiter** les fonctions de base : len(), range(), random (C5)
- **Définir** et invoquer des fonctions avec paramètres et retour (C6)
- **Organiser** le code en modules et packages réutilisables (C7, C8)
- **Programmer** en orienté objet : classes, héritage, encapsulation, polymorphisme (C9–C16)
- **Implémenter** la gestion des exceptions : try/except, raise, exceptions personnalisées (C17–C20)
- **Interagir** avec le système de fichiers et l'OS : os, pathlib, subprocess (C21–C23)

4. Certification préparée

CERTIFICATION VISÉE

Code RS	RS6701
Intitulé	Langage Python
Certificateur	Éditions ENI
Score minimal	700 points sur 1000
Modalités d'examen	QCM/QCU théorique + évaluation pratique (codage Python)
Compétences	C1 à C23 réparties en 5 blocs

5. Programme détaillé

JOUR 1 — Fondamentaux du langage Python (7h)

Compétences ENI visées : C1, C4, C5

Matin (3h30) – Découverte et prise en main

- Présentation de Python : historique, philosophie, cas d'usage en entreprise
- Installation de l'environnement : Python, VS Code, extensions utiles
- Premier script : `print()`, commentaires, exécution d'un programme (C4-Cr1)
- Variables et types de données : `int`, `float`, `str`, `bool` (C1-Cr1)
- Conversions de types : `str()`, `int()`, `float()` (C1-Cr2)
- Opérateurs : arithmétiques, de comparaison et logiques

Méthodes : Démonstrations live, exercices guidés sur la console Python

Après-midi (3h30) – Entrées/sorties et fonctions de base

- Entrées utilisateur : `input()` et boucle de saisie avec `while` (C4-Cr4, C4-Cr5)
- Formatage de chaînes : `format()`, f-strings (C4-Cr2, C4-Cr3)
- Fonction `len()` pour dénombrer les éléments (C5-Cr1)
- Fonction `range()` pour générer des séquences (C5-Cr3)
- Bibliothèque `random` : générer des nombres aléatoires (C5-Cr4)
- Bonnes pratiques : nommage des variables, conventions PEP 8

Méthodes : Exercices progressifs, démonstrations interactives

Travaux pratiques :

- TP1 : Calculatrice interactive avec saisie utilisateur et formatage des résultats
- TP2 : Programme de conversion d'unités (km/miles, €/€, °C/°F) avec f-strings
- TP3 : Quiz interactif avec score final et génération aléatoire de questions

Évaluation formative : Correction collective des TP, tour de table de fin de journée

JOUR 2 — Structures de données et de contrôle (7h)

Compétences ENI visées : C2, C3, C5

Matin (3h30) – Structures de contrôle

- Conditions : `if`, `elif`, `else` — syntaxe et indentation (C3-Cr1)
- Opérateurs logiques : `and`, `or`, `not` — conditions combinées (C3-Cr1)
- Boucle `for` : parcours de séquences, `range()` (C3-Cr2)

- Boucle while : cas d'usage, condition d'arrêt (C3-Cr2)
- Contrôles de boucles : break, continue (C3-Cr2)
- Gestion des erreurs courantes : valeurs manquantes, types inattendus (C3-Cr4)

Méthodes : Démonstrations sur jeux de données réels, exercices guidés

Après-midi (3h30) – Collections et algorithmes

- Listes : création, indexation, slicing, méthodes principales, append() (C2-Cr1)
- Extraction de sous-listes avec for, if et l'opérateur in (C2-Cr1)
- Tuples : différences avec les listes, immutabilité (C5-Cr2)
- Dictionnaires : clés/valeurs, items(), keys(), values() (C2-Cr2)
- Itération sur structures hiérarchiques avec for (C2-Cr2)
- Élaboration d'algorithmes efficaces sur des ensembles de données (C3-Cr3)

Méthodes : Exercices guidés, résolution de problèmes collaboratifs

Travaux pratiques :

- TP1 : Jeu du nombre mystère (boucle while, break, random)
- TP2 : Gestion d'une liste de courses avec filtrage et recherche
- TP3 : Mini-répertoire téléphonique avec dictionnaires imbriqués

Évaluation formative : QCM mi-parcours (20 questions), correction des TP en groupe

JOUR 3 — Fonctions, modules et système de fichiers (7h)

Compétences ENI visées : C6, C7, C8, C21, C22, C23

Matin (3h30) – Fonctions et modularité

- Définir une fonction : def, paramètres, valeur de retour (C6-Cr1 à Cr4)
- Portée des variables : locale vs globale
- Paramètres par défaut, arguments nommés, *args, **kwargs
- Créer un module : arborescence de package, nommage (C7-Cr1 à Cr3)
- Importer un module : import, from, as (C8-Cr1, C8-Cr2)
- Résolution de conflits de noms avec as (C8-Cr2)

Méthodes : Démonstrations, exercices en autonomie

Après-midi (3h30) – Interaction avec le système

- Modules os et sys : getcwd(), getlogin(), variables d'environnement (C21-Cr1, C21-Cr2)
- Module pathlib : Path, absolute(), exists(), is_dir(), is_file() (C22-Cr1, C22-Cr2)
- Lecture/écriture de fichiers avec pathlib : open(), readlines(), writelines() (C22-Cr3)
- Manipulation de fichiers texte et CSV
- Module subprocess : run() et collecte de la sortie standard (C23-Cr1)
- Bonnes pratiques : gestion du contexte avec with

Méthodes : Exercices guidés sur des fichiers et scripts réels

Travaux pratiques :

- TP1 : Bibliothèque de fonctions utilitaires organisée en package
- TP2 : Script d'exploration du système de fichiers avec pathlib
- TP3 : Mini-projet – Analyseur de logs système (lecture, filtrage, écriture)

Évaluation formative : Correction collective des TP, tour de table de fin de journée

JOUR 4 — Programmation orientée objet et exceptions (7h)

Compétences ENI visées : C9 à C20

Matin (3h30) – POO : fondamentaux et concepts avancés

- Déclarer une classe : class, __init__(), attributs (C9-Cr1 à Cr4)
- Déclarer des méthodes : def, self, accès aux attributs (C10-Cr1 à Cr3)
- Instancier une classe : création d'objets, accès aux membres (C11-Cr1, C11-Cr2)
- Encapsulation : double underscore, propriétés, @property (C12-Cr1 à Cr3)
- Membres de classe : @classmethod, cls, attributs de classe (C13-Cr1 à Cr4)
- Héritage : sous-classes, super(), méthodes héritées (C14-Cr1 à Cr5)
- Redéfinition de méthodes et appel à la super-classe (C15-Cr1, C15-Cr2)
- Polymorphisme : isinstance(), même méthode sur objets distincts (C16-Cr1, C16-Cr2)

Méthodes : Démonstrations, exercices progressifs, modélisation collaborative

Après-midi (3h30) – Gestion des exceptions et projet final

- Déclencher une exception : raise, tests sur paramètres (C17-Cr1 à Cr3)
- Gestionnaire d'exceptions : try, except, identification des types (C18-Cr1, C18-Cr2)
- Gestionnaires multiples : ordre des blocs except, hiérarchie (C19-Cr1 à Cr4)
- Exceptions personnalisées : hériter de Exception (C20-Cr1 à Cr3)
- Bloc finally et bonnes pratiques de robustesse

Méthodes : Mini-projet intégrateur en autonomie guidée

Travaux pratiques :

- TP1 : Système de gestion de bibliothèque (héritage : Document, Livre, DVD)
- TP2 : Gestionnaire de contacts POO avec encapsulation et exceptions personnalisées
- TP3 (Mini-projet final) : Application de gestion d'employés — POO + fichiers + exceptions

Évaluation formative : Soutenance du mini-projet, QCM final de 30 questions

6. Méthodes et moyens pédagogiques

MÉTHODES MOBILISÉES

- **Apports théoriques** : exposés interactifs avec diaporama, exemples concrets et démonstrations live
- **Mise en pratique** : TP individuels, exercices progressifs, mini-projets intégrateurs en fin de journée
- **Pédagogie active** : résolution de problèmes collaboratifs
- **Alternance théorie/pratique** : ratio 40 % théorie / 60 % pratique
- **Lien certification** : chaque TP est lié aux critères d'évaluation du référentiel ENI (C1–C23)

MOYENS TECHNIQUES (Indicateur 17)

- Salle de formation équipée : vidéoprojecteur, paperboard, connexion Wi-Fi haut débit
- Poste informatique par stagiaire avec VS Code et extensions Python
- En distanciel : Zoom avec partage d'écran, tableau blanc collaboratif
- Supports de cours remis aux stagiaires

7. Modalités d'évaluation

Avant l'entrée en formation, chaque participant complète un test de positionnement évaluant son aisance informatique générale.

POSITIONNEMENT À L'ENTRÉE

- Test de positionnement en ligne avant la formation

ÉVALUATIONS EN COURS DE FORMATION

- **Évaluation formative** : QCM mi-parcours de 20 questions (Jour 2)
- **Évaluation sommative** : QCM final de 30 questions (Jour 4)
- **Critères de réussite** : obtenir 60 % au QCM final
- **Attestation de fin de formation** : remise à l'issue de la formation, mentionnant les objectifs, la durée et les résultats
- **Certification** : passage de la certification RS6701 – Langage Python (Éditions ENI) en fin de formation

SATISFACTION

- **Questionnaire de satisfaction** en fin de formation (formateur, contenu, logistique, atteinte des objectifs)

8. Accessibilité aux personnes en situation de handicap

Pythonia s'engage à rendre ses formations accessibles aux personnes en situation de handicap. Notre référent handicap est disponible pour étudier les aménagements nécessaires :

- Adaptation des supports pédagogiques (taille de police, contraste, format audio)
- Aménagement des conditions de passage (temps majoré, pauses adaptées)
- Orientation vers nos partenaires spécialisés : Agefiph, Cap Emploi, MDPH

Contact référent handicap : contact@pythonia.fr – 06 66 06 38 59

9. Engagement et suivi

- **Suivi de l'assiduité** : feuille d'émargement signée par demi-journée
- **Prévention des abandons** : suivi individualisé, adaptation du rythme, relance en cas d'absence
- **Réclamations** : toute réclamation peut être adressée par email à contact@pythonia.fr. Elle sera traitée sous 48h ouvrées.

PythonIA — Centre de Formation Python & Intelligence Artificielle

255 boulevard Pereire, 75017 Paris | contact@pythonia.fr | 06 66 06 38 59

SIRET : 52349020900039 – NDA : 11757391975