



PYTHON & DATA : API, INTELLIGENCE ARTIFICIELLE

# Automatisation et scripting avec Python

Automatiser des tâches répétitives avec Python : manipulation de fichiers, emails, scraping web, interaction avec des APIs. Gagnez du temps grâce au scripting.

DURÉE	TARIF HT	NIVEAU	LANGUE	GROUPE	FORMAT
3j (21h)	890.00 €	Intermédiaire	Français	3-12	Formation

## 1 PUBLIC VISÉ

- Développeurs, analystes ou professionnels maîtrisant les bases de Python souhaitant automatiser des tâches répétitives et gagner en productivité.
- Personnes ayant suivi les formations [Python : les fondamentaux du langage](#) et [Python : programmation orientée objet et traitement de données](#) ou disposant d'une expérience Python équivalente.
- Data analysts, administrateurs système ou chefs de projet techniques voulant automatiser des rapports, des alertes ou des échanges de données.
- Toute personne souhaitant écrire des scripts Python utiles et déployables en conditions réelles.

## 2 PRÉREQUIS

- Bonne maîtrise de Python : fonctions, listes, dictionnaires, fichiers, exceptions et notions de POO.
- Les formations [Python : les fondamentaux du langage](#) et [Python : programmation orientée objet et traitement de données](#) sont recommandées.
- Savoir utiliser le terminal et VS Code.

## 3 OBJECTIFS PÉDAGOGIQUES

- Automatiser la manipulation avancée de fichiers et de dossiers avec pathlib et shutil
- Envoyer des emails et des notifications depuis un script Python
- Extraire des données web avec requests et BeautifulSoup
- Automatiser un navigateur web avec Selenium pour les sites dynamiques
- Consommer des APIs REST depuis Python : authentification, pagination, gestion des erreurs
- Planifier l'exécution automatique de scripts avec schedule et cron
- Gérer les secrets et les variables d'environnement de manière sécurisée
- Concevoir et déployer des mini-projets d'automatisation complets et réutilisables

**4 PROGRAMME DÉTAILLÉ****Jour 1 (7h) - Fichiers, emails et notifications****Module 1 - Manipulation avancée de fichiers et dossiers (2h)**

pathlib et shutil (1h)

- pathlib.Path : parcourir, créer, déplacer et supprimer des fichiers et dossiers
- glob et rglob : lister des fichiers selon un motif dans une arborescence
- shutil : copier, déplacer, archiver et compresser des dossiers
- zipfile et tarfile : créer et extraire des archives
- Cas pratique : script d'archivage automatique des fichiers de plus de 30 jours

Surveillance et comparaison de fichiers (1h)

- filecmp : comparer des fichiers et des dossiers
- hashlib : calculer le hash d'un fichier pour détecter les modifications
- watchdog : surveiller un dossier et réagir aux changements en temps réel
- Cas pratique : script de synchronisation qui détecte et copie les fichiers modifiés

**Module 2 - Emails et notifications (2h)**

Envoyer des emails avec Python (1h)

- smtplib : se connecter à un serveur SMTP et envoyer un email simple
- email.mime : composer un email multipart avec texte, HTML et pièces jointes
- Utiliser Gmail, Outlook ou un SMTP d'entreprise depuis Python
- Sécuriser les identifiants SMTP avec les variables d'environnement
- Cas pratique : envoi automatique d'un rapport PDF en pièce jointe

Notifications alternatives (1h)

- Webhooks Slack : envoyer des messages dans un canal depuis un script
- Webhooks Discord : alertes et notifications dans un serveur
- Telegram Bot API : envoyer des messages via un bot Telegram
- Cas pratique : alerte automatique dans Slack quand un fichier d'erreur dépasse un seuil

**Module 3 - Gestion des secrets et planification (3h)**

Gestion des secrets et des variables d'environnement (1h)

- python-dotenv : charger les variables d'environnement depuis un fichier .env
- Ne jamais mettre de secrets dans le code source : les règles de base
- Valider les variables d'environnement au démarrage du script
- keyring : stocker des secrets dans le trousseau système
- Cas pratique : refactoriser un script avec des identifiants codés en dur vers .env

Planification des scripts (2h)

- Le module schedule : planifier des tâches récurrentes en Python pur
- Cron sur Linux : syntaxe, crontab -e, logs et gestion des erreurs
- Planificateur de tâches Windows : créer une tâche planifiée depuis le terminal
- Rendre un script robuste : gestion des exceptions, retries et logs
- Cas pratique : script de rapport hebdomadaire planifié avec cron et notification Slack

**Jour 2 (7h) - Web scraping et APIs REST**

**Module 4 - Web scraping avec requests et BeautifulSoup (3h)**

Requêtes HTTP avec requests (1h)

- GET et POST : envoyer des requêtes et lire les réponses
- Headers, cookies et sessions persistantes avec requests.Session
- Gérer les timeouts, les redirections et les erreurs HTTP
- Télécharger des fichiers en streaming depuis une URL
- Cas pratique : télécharger automatiquement des fichiers depuis une liste d'URLs

Extraire des données avec BeautifulSoup (2h)

- Parser du HTML avec BeautifulSoup : find, find\_all, select (CSS selectors)
- Naviguer dans l'arbre DOM : parents, siblings, children
- Extraire des attributs, du texte et des liens
- Respecter le robots.txt et les délais entre les requêtes
- Gérer les encodages et les caractères spéciaux
- Cas pratique : scraper les annonces d'un site de petites annonces et exporter en CSV

**Module 5 - Selenium pour les sites dynamiques (2h)**

Automatiser un navigateur avec Selenium (1h)

- Installer Selenium et configurer le WebDriver Chrome ou Firefox
- Naviguer, cliquer, remplir des formulaires et soumettre
- Attendre le chargement des éléments : WebDriverWait et expected\_conditions
- Mode headless : exécuter Selenium sans interface graphique

Cas d'usage avancés de Selenium (1h)

- Gérer les popups, les iframes et les onglets multiples
- Prendre des captures d'écran automatiquement
- Exécuter du JavaScript depuis Selenium
- Cas pratique : scraper un site avec pagination JavaScript et exporter les données

**Module 6 - Consommer des APIs REST (2h)**

Interagir avec des APIs depuis Python (1h)

- Anatomie d'une requête API REST : endpoint, headers, authentification, body
- Authentification : API key, Bearer token, OAuth2
- Paginer les résultats d'une API : limit, offset, cursor
- Gérer les erreurs et les rate limits : retry avec backoff exponentiel

Cas d'usage réels (1h)

- API GitHub : lister les dépôts, créer des issues automatiquement
- API Notion ou Airtable : synchroniser des données depuis Python
- API de géolocalisation ou météo : enrichir un jeu de données
- Cas pratique : script qui récupère des données depuis une API publique et génère un rapport

**Jour 3 (7h) - Projets d'automatisation avancés****Module 7 - Traitement et génération de documents (2h)**

Manipuler des fichiers Excel et Word (1h)

- openpyxl : lire, modifier et créer des fichiers Excel avec mise en forme



pandas : lire un Excel, filtrer, calculer et réécrire

- python-docx : générer des documents Word depuis un template
- Cas pratique : générer automatiquement un rapport Word à partir de données Excel

Génération de PDF (1h)

- reportlab : créer des PDF avec texte, tableaux et images
- weasyprint : convertir du HTML/CSS en PDF
- PyPDF2 : fusionner, découper et annoter des PDF existants
- Cas pratique : générer des factures PDF automatiquement depuis un fichier CSV

### Module 8 - Logging et robustesse des scripts (1h)

- Configurer le logging : fichier de log rotatif avec RotatingFileHandler
- Structurer les logs : JSON logging pour une ingestion facile
- Gérer les exceptions de manière exhaustive sans bloquer le script
- Envoyer une alerte en cas d'erreur critique : email ou Slack automatique
- Cas pratique : ajouter un système de logging complet à un script existant

### Module 9 - Projet de synthèse (4h)

Réalisation de deux mini-projets d'automatisation complets

- Mini-projet 1 — Rapport automatique (2h) : script qui récupère des données depuis une API REST, les transforme, génère un rapport Excel et PDF, et l'envoie par email avec notification Slack, planifié avec cron ou schedule
- Mini-projet 2 — Alerte prix (2h) : script qui scrape les prix d'un produit sur un site e-commerce avec BeautifulSoup ou Selenium, compare avec un seuil défini dans un fichier .env, envoie une alerte Telegram ou email si le prix passe en dessous du seuil, et journalise toutes les vérifications dans un fichier de log
- Revue de code collective : robustesse, gestion des erreurs, lisibilité et sécurité des scripts
- Retour formateur individualisé sur les projets rendus

## 5 COMPÉTENCES VISÉES

- Écrire des scripts Python d'automatisation robustes avec gestion des erreurs et des logs
- Extraire et transformer des données depuis des pages web et des APIs REST
- Envoyer des notifications automatiques par email ou webhook
- Planifier des tâches récurrentes sur un serveur Linux ou Windows
- Sécuriser les scripts en production : secrets, variables d'environnement, logs
- Structurer un projet d'automatisation réutilisable et maintenable

## 6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (live coding de scripts fonctionnels sur des cas réels), méthode interrogative (analyse des cas d'usage et des problématiques d'automatisation des participants) et méthode active (exercices pratiques et mini-projets opérationnels). L'accent est mis sur la production de scripts immédiatement réutilisables en conditions professionnelles.



## 7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

---

- Support de cours numérique mis à disposition des apprenants
- Dépôt GitHub de démonstration avec scripts d'exemple et corrections par module
- Environnement de développement : VS Code + Python 3.12+ + Chrome WebDriver
- Pour le distanciel : visioconférence (Zoom ou équivalent), partage d'écran, chat en direct
- Accès à la plateforme pédagogique LaPolaris (supports, ressources, émargement)

## 8 MODALITÉS D'ÉVALUATION

---

- En cours de formation : exercices pratiques et scripts corrigés à chaque module
- En fin de formation : réalisation de deux mini-projets d'automatisation complets et opérationnels
- Questionnaire d'auto-évaluation des acquis en fin de parcours

## 9 CRITÈRES D'ÉVALUATION

---

- Scripts fonctionnels et robustes avec gestion complète des exceptions et des cas limites
- Extraction correcte des données web avec BeautifulSoup ou Selenium selon le type de site
- Consommation fiable d'une API REST avec authentification et gestion de la pagination
- Sécurisation des secrets via variables d'environnement sans aucun identifiant dans le code source
- Logs structurés permettant de tracer l'exécution et les erreurs du script en production

## 10 MODALITÉS DE VALIDATION

---

Attestation de fin de formation délivrée à l'issue du parcours, conditionnée à une assiduité d'au moins 80 % et à la réalisation des mini-projets de synthèse. L'attestation précise les objectifs atteints et les compétences acquises.

## 11 SUIVI ET ACCOMPAGNEMENT

---

- Feuilles d'émargement signées par demi-journée (présentiel) ou émargement numérique (distanciel)
- Traçabilité des activités pédagogiques réalisées
- Attestation d'assiduité délivrée en fin de formation
- Suivi individuel via les exercices corrigés et les mini-projets de synthèse

## 12 CONDITIONS D'ACCÈS

---

Formation accessible sur inscription directe, sans prérequis administratif particulier. Le financement peut être pris en charge par l'employeur dans le cadre d'un plan de développement des compétences, ou en autofinancement avec possibilité de paiement en plusieurs fois.

## 13 DÉLAIS D'ACCÈS

---

Inscription possible jusqu'à **5 jours ouvrés** avant le début de la session. Pour toute demande urgente, nous contacter directement.



## ACCESSIBILITÉ · HANDICAP

Nos formations sont accessibles aux personnes en situation de handicap. Pour toute situation nécessitant un aménagement (matériel, temporel ou pédagogique), nous vous invitons à nous contacter avant l'inscription afin d'étudier les adaptations possibles.

Référent handicap : [contact@lapolaris.fr](mailto:contact@lapolaris.fr)

## LaPolaris

TÉL. +33762584798

EMAIL [contact@lapolaris.fr](mailto:contact@lapolaris.fr)

WEB [lapolaris.fr](http://lapolaris.fr)