

Formation Python Avancé : Web Services + ORM

■ Durée :	5 jours (35 heures)
■ Tarifs inter-entreprise :	2 995,00 CHF HT (standard) 2 396,00 CHF HT (remisé)
■ Public :	Développeurs Python
■ Pré-requis :	Avoir suivi le stage "Python : Initiation + Approfondissement" ou notions équivalentes
■ Objectifs :	Construire une API en Python et implémenter une couche d'accès aux données avec un ORM
■ Modalités pédagogiques, techniques et d'encadrement :	<ul style="list-style-type: none">• Formation synchrone en présentiel et distanciel.• Méthodologie basée sur l'Active Learning : 75 % de pratique minimum.• Un PC par participant en présentiel, possibilité de mettre à disposition en bureau à distance un PC et l'environnement adéquat.• Un formateur expert.
■ Modalités d'évaluation :	<ul style="list-style-type: none">• Définition des besoins et attentes des apprenants en amont de la formation.• Auto-positionnement à l'entrée et la sortie de la formation.• Suivi continu par les formateurs durant les ateliers pratiques.• Évaluation à chaud de l'adéquation au besoin professionnel des apprenants le dernier jour de formation.
■ Sanction :	Attestation de fin de formation mentionnant le résultat des acquis
■ Référence :	PYT100613-F
■ Note de satisfaction des participants:	Pas de données disponibles
■ Contacts :	commercial@dawan.fr - 09 72 37 73 73

■ Modalités d'accès :	Possibilité de faire un devis en ligne (www.dawan.fr , moncompteformation.gouv.fr , maformation.fr , etc.) ou en appelant au standard.
■ Délais d'accès :	Variable selon le type de financement.
■ Accessibilité :	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à referenthandicap@dawan.fr , nous étudierons ensemble vos besoins

Connaître le besoin et l'écosystème

- Présentation des Web Services (WS) : fonctionnement, intérêt, interopérabilité
- Architecture orientée service (SOA) : composantes, technologies
- Technologies : Architecture REST
- Design et documentation : Swagger
- Outils de test de services web

Interroger des services web REST

- Outils d'interrogation en Python
- Gestion des paramètres d'accès aux API
- Types d'authentification aux API RESTful (BASIC HTTP, Bearer Token etc.)

Atelier : Interroger des services web ouverts et avec authentification

Implémenter un service web REST

- Rappels sur l'architecture REST
- Composantes et méthodes d'appel
- Sérialisation des données
- Conseils pour la création de points d'entrée REST

Implémenter un service web avec Python

- Bibliothèques d'API REST : FastAPI, Flask ou Sanic
- Déclarer des routes et des méthodes

- Accepter des paramètres
- Validation des paramètres et gestion des erreurs
- Authentification (BASIC, Bearer Token)

Atelier : Créer un service web avec FastAPI

Implémenter un service web avec Django

- Découverte de l'écosystème des frameworks REST pour Django (DRF, Ninja)
- Déclarer des routes et des méthodes
- Accepter des paramètres
- Validation des paramètres et gestion des erreurs
- Authentification (BASIC, Bearer Token)
- Gestion des droits dans Django
- Mise en cache (Memcached ou Redis)

Atelier : Créer un service web avec Django-Ninja

Déploiement d'un service web

- Architecture de base d'un déploiement Django
- Architecture de base d'un déploiement FastAPI

Manipuler des bases de données en programmation objet (ORM)

- Principe des ORM (Pattern DAO)
- Bibliothèques d'ORM Python : SQLAlchemy, Peewee, PonyORM, Django

Découverte d'un ORM

- Choix d'un ORM
- Découvrir les concepts de base des ORMs
- Mapping : modèles, colonnes, métadonnées de tables
- Gérer son schéma de données
- Concept de migrations de schéma
- Gestion des relations entre tables : n à 1, 1 à 1, n à n

- Optimisations (syndrome des n+1 requêtes, etc.)

Atelier : Découverte de l'ORM Peewee et Django

Ecrire des requêtes avec un ORM

- Sélections de base, filtres
- Jointures en SQL et jointures en objet
- Fonctions d'agrégation, scalaires et de fenêtrage
- Désérialisation lazy/eager

Atelier : Développer une couche modèle efficacement

Découvrir des fonctionnalités avancées

- Mettre en cache (ex. Redis ou autres)
- Outils de mise en cache avec Django
- Sigaux (intercepteurs, event-listeners)

Atelier : Découvrir les outils de cache pour ORM