

## Formation Python Avancé : Web Services + ORM

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

■ <b>Modalités d'accès :</b>	Possibilité de faire un devis en ligne ( <a href="http://www.dawan.fr">www.dawan.fr</a> , <a href="http://moncompteformation.gouv.fr">moncompteformation.gouv.fr</a> , <a href="http://maformation.fr">maformation.fr</a> , etc.) ou en appelant au standard.
■ <b>Délais d'accès :</b>	Variable selon le type de financement.
■ <b>Accessibilité :</b>	Si vous êtes en situation de handicap, nous sommes en mesure de vous accueillir, n'hésitez pas à nous contacter à <a href="mailto:referenthandicap@dawan.fr">referenthandicap@dawan.fr</a> , 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**