

PROGRAMME DE FORMATION

DÉVELOPPEMENT WEB AVEC PYTHON – DJANGO

FICHE SYNTHÉTIQUE DE LA FORMATION

Intitulé	Développement Web avec Python – Django
Durée	3 jours (21 heures)
Référence	PY-WEB-3J-DJG
Tarif	1 500 € HT / personne
Modalité	Présentiel ou distanciel
Public	Développeurs, intégrateurs web, administrateurs système, reconversion professionnelle
Prérequis	Notions de base en programmation (variables, boucles, fonctions). Aisance informatique de base.
Lieu	255 boulevard Pereire, 75017 Paris ou à distance (Zoom)

1. Public visé

Cette formation s'adresse aux développeurs souhaitant maîtriser le développement web avec Python et le framework Django, aux intégrateurs web désirant élargir leurs compétences back-end, aux administrateurs système souhaitant automatiser des interfaces web, ainsi qu'aux professionnels en reconversion vers le développement web.

2. Prérequis

- Notions de base en programmation : variables, conditions, boucles, fonctions
- Aisance minimale avec l'outil informatique (naviguer dans des dossiers, utiliser un navigateur, un terminal)
- Une première expérience avec Python est un plus mais n'est pas obligatoire

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 :

- Maîtriser les fondamentaux de Python et la programmation orientée objet (classes, héritage, encapsulation)
- Concevoir et manipuler une base de données relationnelle avec l'ORM Django
- Installer, configurer et structurer un projet Django
- Définir des modèles de données et gérer les migrations
- Configurer le routage d'URL et créer des vues (fonctions et classes)
- Créer des templates dynamiques avec le moteur de templates Django
- Gérer les formulaires et la validation des données
- Développer une application web complète de bout en bout

4. PROGRAMME DÉTAILLÉ

JOUR 1 — Bases de Python et Programmation Orientée Objet (7h)

Matin (3h30) – Fondamentaux du langage Python

- Rappels et consolidation : variables, types de données (int, float, str, bool), conversions
- Structures de contrôle : if/elif/else, boucles for et while, break/continue
- Structures de données : listes, tuples, dictionnaires, ensembles
- Fonctions : définition, paramètres, valeurs de retour, *args, **kwargs
- Manipulation de chaînes de caractères et f-strings
- Gestion des fichiers : lecture/écriture avec open() et le gestionnaire de contexte with

Méthodes : Démonstrations live, exercices guidés, résolution de problèmes progressifs

Après-midi (3h30) – Programmation Orientée Objet

- Concepts fondamentaux : classes, objets, attributs, méthodes
- Le constructeur `__init__()` et le paramètre `self`
- Encapsulation : attributs privés, propriétés, décorateur `@property`
- Héritage : sous-classes, `super()`, redéfinition de méthodes
- Polymorphisme : `isinstance()`, méthodes communes sur objets distincts
- Méthodes spéciales : `__str__()`, `__repr__()`, `__len__()`
- Gestion des exceptions : `try/except`, `raise`, exceptions personnalisées

Méthodes : Exercices progressifs, modélisation collaborative

Travaux pratiques :

- TP1 : Création d'un module utilitaire Python (fonctions, manipulation de fichiers)
- TP2 : Modélisation d'un système de gestion de produits en POO (classes Produit, Catégorie, Inventaire)
- TP3 : Mise en place d'un mini-projet avec héritage et exceptions personnalisées

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

JOUR 2 — Bases de données et Framework Django (7h)

Matin (3h30) – Bases de données relationnelles et SQL

- Introduction aux bases de données relationnelles : concepts, tables, relations
- Le langage SQL : SELECT, INSERT, UPDATE, DELETE
- Jointures et filtres : WHERE, JOIN, GROUP BY, ORDER BY
- SQLite : présentation et utilisation avec Python (module sqlite3)
- Création d'une base de données, insertion et requêtes depuis Python
- Bonnes pratiques : requêtes paramétrées, gestion des connexions

Méthodes : Démonstrations sur bases de données réelles, exercices guidés en console

Après-midi (3h30) – Découverte et mise en place de Django

- Présentation de Django : philosophie, architecture MTV (Model-Template-View)
- Installation de Django et création d'un projet : django-admin, startproject
- Structure d'un projet Django : settings.py, urls.py, wsgi.py
- Création d'une application : startapp, organisation des fichiers
- Introduction à l'ORM Django : définition des modèles (models.py)
- Types de champs : CharField, IntegerField, DateField, ForeignKey, ManyToManyField
- Migrations : makemigrations, migrate — synchronisation schéma/code
- Interface d'administration Django : création de superuser, personnalisation de l'admin

Méthodes : Démonstrations live, configuration pas à pas, exercices en autonomie

Travaux pratiques :

- TP1 : Création et interrogation d'une base SQLite avec Python (carnet de contacts)
- TP2 : Mise en place d'un projet Django avec modèles de données (blog : Article, Auteur, Catégorie)
- TP3 : Configuration de l'interface d'administration et peuplement de la base de données

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

JOUR 3 — Routes, vues, templates et application web finale (7h)

Matin (3h30) – Routage, vues et templates Django

- Système de routage : urls.py, path(), name, espaces de noms (namespaces)
- Vues fonctionnelles : HttpRequest, HttpResponse, render(), redirect()
- Vues basées sur les classes (CBV) : ListView, DetailView, CreateView
- Moteur de templates Django : syntaxe, variables, filtres, balises
- Héritage de templates : base.html, {% block %}, {% extends %}
- Fichiers statiques : CSS, JavaScript, images — configuration et utilisation

Méthodes : Démonstrations interactives, exercices progressifs

Après-midi (3h30) – Formulaires et projet final

- Formulaires Django : Form, ModelForm, validation, affichage
- Gestion des requêtes POST : traitement, sauvegarde, messages de confirmation
- Sécurité de base : CSRF, protection des formulaires
- Mini-projet intégrateur : développement d'une application web complète

Méthodes : Mini-projet intégrateur en autonomie guidée

Travaux pratiques :

- TP1 : Création de vues et templates pour un site vitrine (pages d'accueil, à propos, contact)
- TP2 : Formulaire de contact fonctionnel avec validation et sauvegarde en base
- TP3 (Mini-projet final) : Application web de gestion de tâches — modèles, vues, templates, formulaires, CRUD complet

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

5. 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

MOYENS TECHNIQUES

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

6. MODALITÉS D'ÉVALUATION

Avant l'entrée en formation, chaque participant complète un test de positionnement évaluant son niveau en programmation et ses connaissances de base en informatique.

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 3)

- 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

SATISFACTION

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

7. 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

8. 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.