

Mis à jour le 08/01/2026

S'inscrire

Formation JetBrains PyCharm

3 jours (21 heures)

Présentation

JetBrains PyCharm est un IDE complet pour développer en Python plus vite et avec moins d'erreurs. Cette formation vous aide à exploiter ses fonctions clés pour coder, déboguer, tester et refactorer efficacement sur des projets réels.

A l'issue de notre formation, vous pourrez rendre votre environnement de développement productif et fiable : navigation dans le code, inspections, auto-complétion, gestion des dépendances, exécution et débogage. Vous apprendrez à structurer un projet, configurer des interpréteurs et automatiser les tâches courantes.

L'approche est 100% pratique via ateliers guidés et démos : création d'un projet, configuration d'un venv, exécution de scripts et tests, débogage pas à pas, refactoring sécurisé, intégration Git. Les livrables incluent un projet Python configuré, des run/debug configurations, une checklist de réglages et des bonnes pratiques d'usage au quotidien.

Objectifs

- Configurer PyCharm et un interpréteur Python (venv/conda).
- Créer et organiser un projet avec exécution et paramètres.
- Déboguer efficacement (breakpoints, watch, step, exceptions).
- Écrire et exécuter des tests avec un workflow rapide.
- Refactorer et naviguer dans le code avec les inspections.

Public visé

- Développeurs Python débutants à intermédiaires
- Développeurs web (Django/Flask) souhaitant industrialiser leur workflow
- Data engineers / data analysts utilisant Python au quotidien

Pré-requis

- Notions de base en Python (scripts, fonctions, modules)
- Compréhension des environnements et dépendances (pip, venv)
- Connaissances élémentaires en ligne de commande
- Notions de Git (commit, branch) appréciées

Pré-requis techniques

- Ordinateur avec 8 Go RAM minimum (16 Go recommandé)
- Windows, macOS ou Linux
- Python 3.x installé et accessible en ligne de commande
- PyCharm installé (Community ou Professional selon besoins)
- Accès à un terminal (PowerShell, zsh, bash) et à Git

Programme de formation JetBrains PyCharm

[Jour 1 - Matin]

Prise en main de PyCharm et configuration d'un environnement Python

- Tour d'horizon : interface, Project view, éditeur, tool windows, raccourcis essentiels
- Création d'un projet : structure, fichiers, templates, interpréteur Python
- Gestion des environnements : venv/virtualenv, sélection d'interpréteur, packages
- Paramétrage productif : inspections, formatage, thèmes, keymap, plugins utiles
- Atelier pratique : Créer un projet, configurer un venv et installer des dépendances.

[Jour 1 - Après-midi]

Édition avancée et navigation efficace dans le code

- Navigation : Search Everywhere, Go to Definition, usages, structure, bookmarks
- Refactoring : rename, extract method/variable, move, safe delete
- Qualité de code : inspections, quick-fixes, intentions, reformat, import optimisés
- Productivité : Live Templates, postfix completion, multi-cursors, macros
- Atelier pratique : Refactorer un module existant en appliquant inspections et templates.

[Jour 2 - Matin]

Débogage et exécution : maîtriser Run/Debug et les configurations

- Run/Debug configurations : scripts, paramètres, variables d'environnement, working directory

- Débogage : breakpoints, step in/over/out, evaluate expression, watches
- Débogage avancé : breakpoints conditionnels, logpoints, exception breakpoints
- Analyse : call stack, frames, variables, threads (selon contexte)
- Atelier pratique : Diagnostiquer et corriger un bug avec breakpoints conditionnels et evaluate.

[Jour 2 - Après-midi]

Tests, qualité et outillage : pytest, coverage et linters

- Configuration des tests : pytest/unittest, discovery, marquages, exécution ciblée
- Debug des tests : exécution pas à pas, analyse des échecs, rerun failed
- Couverture : génération, lecture des rapports, zones non couvertes
- Qualité : intégration flake8/pylint/ruff, mypy (selon projet), gestion des règles
- Atelier pratique : Mettre en place une suite pytest avec coverage et corriger les alertes de lint.

[Jour 3 - Matin]

Git dans PyCharm : workflow quotidien et bonnes pratiques

- Initialiser et cloner : VCS, .gitignore, configuration des remotes
- Workflow : commit, amend, stash, rebase/cherry-pick (selon politique), résolution de conflits
- Comparaisons : diff, historique, annotate/blame, recherche dans l'historique
- Branches et PR : gestion des branches, tags, intégration avec plateformes (selon contexte)
- Atelier pratique : Réaliser une feature branch, résoudre un conflit et préparer un commit propre.

[Jour 3 - Après-midi]

Packaging, exécution outillée et optimisation du workflow PyCharm

- Gestion des dépendances : requirements.txt, pip-tools/poetry (selon projet), synchronisation d'environnement
- Exécution outillée : terminal intégré, tâches, scripts, paramètres réutilisables
- Profiling et performance : profiler (selon édition), analyse de points chauds, pistes d'optimisation
- Personnalisation durable : settings sync, export/import, inspections par projet, conventions d'équipe
- Atelier pratique : Packager et exécuter un projet avec configurations réutilisables et analyser un ralentissement.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.