+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 01/08/2026

Register

Jetbrains PyCharm Training

3 days (21 hours)

Overview

JetBrains PyCharm is a comprehensive IDE for developing Python applications faster and with fewer errors. This training course will help you leverage its key features to code, debug, test, and refactor efficiently on real-world projects.

By the end of our training, you will be able to make your development environment productive and reliable: code navigation, inspections, auto-completion, dependency management, execution, and debugging. You will learn how to structure a project, configure interpreters, and automate common tasks.

The approach is 100% practical, with guided workshops and demos: creating a project, configuring a venv, executing scripts and tests, step-by-step debugging, secure refactoring, Git integration. Deliverables include a configured Python project, run/debug configurations, a checklist of settings, and best practices for everyday use.

Objectives

- Configure PyCharm and a Python interpreter (venv/conda).
- Create and organize a project with execution and settings.
- Debug efficiently (breakpoints, watch, step, exceptions).
- Write and run tests with a fast workflow.
- Refactor and navigate code with inspections.

Target audience

- Beginner to intermediate Python developers
- Web developers (Django/Flask) who want to industrialize their workflow
- Data engineers/data analysts who use Python on a daily basis

Prerequisites

- Basic knowledge of Python (scripts, functions, modules)
- Understanding of environments and dependencies (pip, venv)
- Basic knowledge of command line
- Knowledge of Git (commit, branch) appreciated

Technical prerequisites

- Computer with at least 8 GB RAM (16 GB recommended)
- Windows, macOS, or Linux
- Python 3.x installed and accessible via command line
- PyCharm installed (Community or Professional depending on requirements)
- Access to a terminal (PowerShell, zsh, bash) and Git

Jetbrains PyCharm training program

[Day 1 - Morning]

Getting started with PyCharm and setting up a Python environment

- Overview: interface, Project view, editor, tool windows, essential shortcuts
- Creating a project: structure, files, templates, Python interpreter
- Environment management: venv/virtualenv, interpreter selection, packages
- Productive settings: inspections, formatting, themes, keymap, useful plugins
- Hands-on workshop: Create a project, configure a venv, and install dependencies.

[Day 1 - Afternoon]

Advanced editing and efficient code navigation

- Navigation: Search Everywhere, Go to Definition, uses, structure, bookmarks
- Refactoring: rename, extract method/variable, move, safe delete
- Code quality: inspections, quick fixes, intentions, reformat, optimized imports
- Productivity: Live Templates, postfix completion, multi-cursors, macros
- Hands-on workshop: Refactor an existing module by applying inspections and templates.

[Day 2 - Morning]

Debugging and execution: mastering Run/Debug and configurations

- Run/Debug configurations: scripts, parameters, environment variables, working directory

- Debugging: breakpoints, step in/over/out, evaluate expression, watches
- Advanced debugging: conditional breakpoints, logpoints, exception breakpoints
- Analysis: call stack, frames, variables, threads (depending on context)
- Hands-on workshop: Diagnosing and fixing a bug with conditional breakpoints and evaluate.

[Day 2 - Afternoon]

Testing, quality, and tools: pytest, coverage, and linters

- Test configuration: pytest/unittest, discovery, markings, targeted execution
- Test debugging: step-by-step execution, failure analysis, rerun failed
- Coverage: generation, reading reports, uncovered areas
- Quality: flake8/pylint/ruff integration, mypy (depending on project), rule management
- Hands-on workshop: Setting up a pytest suite with coverage and fixing lint alerts.

[Day 3 - Morning]

Git in PyCharm: daily workflow and best practices

- Initialize and clone: VCS, .gitignore, remote configuration
- Workflow: commit, amend, stash, rebase/cherry-pick (depending on policy), conflict resolution
- Comparisons: diff, history, annotate/blame, searching history
- Branches and PR: branch management, tags, integration with platforms (depending on context)
- Hands-on workshop: Creating a feature branch, resolving a conflict, and preparing a clean commit.

[Day 3 - Afternoon]

Packaging, tool-based execution, and PyCharm workflow optimization

- Dependency management: requirements.txt, pip-tools/poetry (depending on project), environment synchronization
- Tool-based execution: integrated terminal, tasks, scripts, reusable parameters
- Profiling and performance: profiler (depending on edition), hotspot analysis, optimization tips
- Long-term customization: settings sync, export/import, project-specific inspections, team conventions
- Hands-on workshop: Package and execute a project with reusable configurations and analyze a slowdown.

Target companies

This training is intended for both individuals and companies, small or large, wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Positioning at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and discussion sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.